

Handbuch | DE

# TX1200

TwinCAT 2 | PLC-Bibliothek: TcLON





# Inhaltsverzeichnis

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Vorwort.....</b>                                       | <b>15</b> |
| 1.1      | Hinweise zur Dokumentation .....                          | 15        |
| 1.2      | Zu Ihrer Sicherheit.....                                  | 16        |
| 1.3      | Hinweise zur Informationssicherheit .....                 | 17        |
| <b>2</b> | <b>Einleitung.....</b>                                    | <b>18</b> |
| <b>3</b> | <b>Zielgruppen.....</b>                                   | <b>19</b> |
| <b>4</b> | <b>LON.....</b>   | <b>20</b> |
| 4.1      | Unterstützte SNVTs .....                                  | 20        |
| 4.2      | Arbeitsweise der KL6401 .....                             | 25        |
| <b>5</b> | <b>Konfiguration mit KS2000 .....</b>                     | <b>27</b> |
| <b>6</b> | <b>Integration in TwinCAT .....</b>                       | <b>30</b> |
| 6.1      | KL6401 - Verknüpfung mit dem TwinCAT System Manager ..... | 30        |
| 6.2      | Integration in TwinCAT (CX9020) .....                     | 35        |
| 6.3      | Integration in TwinCAT (BC9191) .....                     | 38        |
| <b>7</b> | <b>Programmierung .....</b>                               | <b>43</b> |
| 7.1      | Allgemeine Informationen .....                            | 58        |
| 7.2      | Bausteine .....   | 59        |
| 7.2.1    | FB_LON_KL6401.....  | 68        |
| 7.2.2    | FB_READ_001_SNVT_amp.....                                 | 70        |
| 7.2.3    | FB_READ_002_SNVT_amp_mil .....                            | 70        |
| 7.2.4    | FB_READ_003_SNVT_angle .....                              | 71        |
| 7.2.5    | FB_READ_004_SNVT_angle_vel.....                           | 72        |
| 7.2.6    | FB_READ_005_SNVT_btu_kilo.....                            | 73        |
| 7.2.7    | FB_READ_006_SNVT_btu_mega .....                           | 73        |
| 7.2.8    | FB_READ_007_SNVT_char_ascii.....                          | 74        |
| 7.2.9    | FB_READ_008_SNVT_count .....                              | 75        |
| 7.2.10   | FB_READ_009_SNVT_count_inc.....                           | 76        |
| 7.2.11   | FB_READ_011_SNVT_date_day .....                           | 76        |
| 7.2.12   | FB_READ_013_SNVT_elec_kwh.....                            | 77        |
| 7.2.13   | FB_READ_014_SNVT_elec_whr.....                            | 78        |
| 7.2.14   | FB_READ_015_SNVT_flow.....                                | 79        |
| 7.2.15   | FB_READ_016_SNVT_flow_mil.....                            | 80        |
| 7.2.16   | FB_READ_017_SNVT_length .....                             | 80        |
| 7.2.17   | FB_READ_018_SNVT_length_kilo.....                         | 81        |
| 7.2.18   | FB_READ_019_SNVT_length_micr .....                        | 82        |
| 7.2.19   | FB_READ_020_SNVT_length_mil.....                          | 83        |
| 7.2.20   | FB_READ_021_SNVT_lev_cont.....                            | 83        |
| 7.2.21   | FB_READ_023_SNVT_mass .....                               | 84        |
| 7.2.22   | FB_READ_024_SNVT_mass_kilo.....                           | 85        |
| 7.2.23   | FB_READ_025_SNVT_mass_mega .....                          | 86        |
| 7.2.24   | FB_READ_026_SNVT_mass_mil.....                            | 86        |
| 7.2.25   | FB_READ_027_SNVT_power .....                              | 87        |
| 7.2.26   | FB_READ_028_SNVT_power_kilo.....                          | 88        |

|        |                                   |     |
|--------|-----------------------------------|-----|
| 7.2.27 | FB_READ_029_SNVT_ppm.....         | 89  |
| 7.2.28 | FB_READ_030_SNVT_press.....       | 89  |
| 7.2.29 | FB_READ_031_SNVT_res.....         | 90  |
| 7.2.30 | FB_READ_032_SNVT_res_kilo.....    | 91  |
| 7.2.31 | FB_READ_033_SNVT_sound_db.....    | 92  |
| 7.2.32 | FB_READ_034_SNVT_speed.....       | 92  |
| 7.2.33 | FB_READ_035_SNVT_speed_mil.....   | 93  |
| 7.2.34 | FB_READ_036_SNVT_str_asc.....     | 94  |
| 7.2.35 | FB_READ_037_SNVT_str_int.....     | 95  |
| 7.2.36 | FB_READ_038_SNVT_telcom.....      | 95  |
| 7.2.37 | FB_READ_039_SNVT_temp.....        | 96  |
| 7.2.38 | FB_READ_041_SNVT_vol.....         | 97  |
| 7.2.39 | FB_READ_042_SNVT_vol_kilo.....    | 98  |
| 7.2.40 | FB_READ_043_SNVT_vol_mil.....     | 99  |
| 7.2.41 | FB_READ_044_SNVT_volt.....        | 99  |
| 7.2.42 | FB_READ_045_SNVT_volt_dbmv.....   | 100 |
| 7.2.43 | FB_READ_046_SNVT_volt_kilo.....   | 101 |
| 7.2.44 | FB_READ_047_SNVT_volt_mil.....    | 102 |
| 7.2.45 | FB_READ_048_SNVT_amp_f.....       | 102 |
| 7.2.46 | FB_READ_049_SNVT_angle_f.....     | 103 |
| 7.2.47 | FB_READ_050_SNVT_angle_vel_f..... | 104 |
| 7.2.48 | FB_READ_051_SNVT_count_f.....     | 105 |
| 7.2.49 | FB_READ_052_SNVT_count_inc_f..... | 105 |
| 7.2.50 | FB_READ_053_SNVT_flow_f.....      | 106 |
| 7.2.51 | FB_READ_054_SNVT_length_f.....    | 107 |
| 7.2.52 | FB_READ_055_SNVT_lev_cont_f.....  | 108 |
| 7.2.53 | FB_READ_056_SNVT_mass_f.....      | 108 |
| 7.2.54 | FB_READ_057_SNVT_power_f.....     | 109 |
| 7.2.55 | FB_READ_058_SNVT_ppm_f.....       | 110 |
| 7.2.56 | FB_READ_059_SNVT_press_f.....     | 111 |
| 7.2.57 | FB_READ_060_SNVT_res_f.....       | 111 |
| 7.2.58 | FB_READ_061_SNVT_sound_db_f.....  | 112 |
| 7.2.59 | FB_READ_062_SNVT_speed_f.....     | 113 |
| 7.2.60 | FB_READ_063_SNVT_temp_f.....      | 114 |
| 7.2.61 | FB_READ_064_SNVT_time_f.....      | 114 |
| 7.2.62 | FB_READ_065_SNVT_vol_f.....       | 115 |
| 7.2.63 | FB_READ_066_SNVT_volt_f.....      | 116 |
| 7.2.64 | FB_READ_067_SNVT_btu_f.....       | 117 |
| 7.2.65 | FB_READ_068_SNVT_elec_whr_f.....  | 117 |
| 7.2.66 | FB_READ_069_SNVT_config_src.....  | 118 |
| 7.2.67 | FB_READ_070_SNVT_color.....       | 119 |
| 7.2.68 | FB_READ_071_SNVT_grammage.....    | 120 |
| 7.2.69 | FB_READ_072_SNVT_grammage_f.....  | 120 |
| 7.2.70 | FB_READ_073_SNVT_file_req.....    | 121 |
| 7.2.71 | FB_READ_074_SNVT_file_status..... | 122 |
| 7.2.72 | FB_READ_075_SNVT_freq_f.....      | 123 |

|         |                                   |     |
|---------|-----------------------------------|-----|
| 7.2.73  | FB_READ_076_SNVT_freq_hz.....     | 123 |
| 7.2.74  | FB_READ_077_SNVT_freq_kilohz..... | 124 |
| 7.2.75  | FB_READ_078_SNVT_freq_milhz.....  | 125 |
| 7.2.76  | FB_READ_079_SNVT_lux.....         | 126 |
| 7.2.77  | FB_READ_081_SNVT_lev_percent..... | 126 |
| 7.2.78  | FB_READ_082_SNVT_multiplier.....  | 127 |
| 7.2.79  | FB_READ_083_SNVT_state.....       | 128 |
| 7.2.80  | FB_READ_084_SNVT_time_stamp.....  | 129 |
| 7.2.81  | FB_READ_085_SNVT_zerospan.....    | 130 |
| 7.2.82  | FB_READ_086_SNVT_magcard.....     | 130 |
| 7.2.83  | FB_READ_087_SNVT_elapsed_tm.....  | 131 |
| 7.2.84  | FB_READ_088_SNVT_alarm.....       | 132 |
| 7.2.85  | FB_READ_089_SNVT_currency.....    | 133 |
| 7.2.86  | FB_READ_090_SNVT_file_pos.....    | 134 |
| 7.2.87  | FB_READ_091_SNVT_muldiv.....      | 134 |
| 7.2.88  | FB_READ_092_SNVT_obj_request..... | 135 |
| 7.2.89  | FB_READ_093_SNVT_obj_status.....  | 136 |
| 7.2.90  | FB_READ_094_SNVT_preset.....      | 137 |
| 7.2.91  | FB_READ_095_SNVT_switch.....      | 138 |
| 7.2.92  | FB_READ_096_SNVT_trans_table..... | 138 |
| 7.2.93  | FB_READ_097_SNVT_override.....    | 139 |
| 7.2.94  | FB_READ_098_SNVT_pwr_fact.....    | 140 |
| 7.2.95  | FB_READ_099_SNVT_pwr_fact_f.....  | 141 |
| 7.2.96  | FB_READ_100_SNVT_density.....     | 142 |
| 7.2.97  | FB_READ_101_SNVT_density_f.....   | 142 |
| 7.2.98  | FB_READ_102_SNVT_rpm.....         | 143 |
| 7.2.99  | FB_READ_103_SNVT_hvac_emerg.....  | 144 |
| 7.2.100 | FB_READ_104_SNVT_angle_deg.....   | 145 |
| 7.2.101 | FB_READ_105_SNVT_temp_p.....      | 145 |
| 7.2.102 | FB_READ_106_SNVT_temp_setpt.....  | 146 |
| 7.2.103 | FB_READ_107_SNVT_time_sec.....    | 147 |
| 7.2.104 | FB_READ_108_SNVT_hvac_mode.....   | 148 |
| 7.2.105 | FB_READ_109_SNVT_occupancy.....   | 148 |
| 7.2.106 | FB_READ_110_SNVT_area.....        | 149 |
| 7.2.107 | FB_READ_111_SNVT_hvac_overid..... | 150 |
| 7.2.108 | FB_READ_112_SNVT_hvac_status..... | 151 |
| 7.2.109 | FB_READ_113_SNVT_press_p.....     | 152 |
| 7.2.110 | FB_READ_114_SNVT_address.....     | 152 |
| 7.2.111 | FB_READ_115_SNVT_scene.....       | 153 |
| 7.2.112 | FB_READ_116_SNVT_scene_cfg.....   | 154 |
| 7.2.113 | FB_READ_117_SNVT_setting.....     | 155 |
| 7.2.114 | FB_READ_118_SNVT_evap_state.....  | 155 |
| 7.2.115 | FB_READ_119_SNVT_therm_mode.....  | 156 |
| 7.2.116 | FB_READ_120_SNVT_defr_mode.....   | 157 |
| 7.2.117 | FB_READ_121_SNVT_defr_term.....   | 158 |
| 7.2.118 | FB_READ_122_SNVT_defr_state.....  | 159 |

|         |                                    |     |
|---------|------------------------------------|-----|
| 7.2.119 | FB_READ_123_SNVT_time_min .....    | 159 |
| 7.2.120 | FB_READ_124_SNVT_time_hour .....   | 160 |
| 7.2.121 | FB_READ_125_SNVT_ph .....          | 161 |
| 7.2.122 | FB_READ_126_SNVT_ph_f .....        | 162 |
| 7.2.123 | FB_READ_127_SNVT_chlr_status ..... | 162 |
| 7.2.124 | FB_READ_128_SNVT_tod_event .....   | 163 |
| 7.2.125 | FB_READ_129_SNVT_smo_obscur .....  | 164 |
| 7.2.126 | FB_READ_130_SNVT_fire_test .....   | 165 |
| 7.2.127 | FB_READ_131_SNVT_temp_ror .....    | 166 |
| 7.2.128 | FB_READ_132_SNVT_fire_init .....   | 166 |
| 7.2.129 | FB_READ_133_SNVT_fire_indcte ..... | 167 |
| 7.2.130 | FB_READ_134_SNVT_time_zone .....   | 168 |
| 7.2.131 | FB_READ_135_SNVT_earth_pos .....   | 169 |
| 7.2.132 | FB_READ_136_SNVT_reg_val .....     | 170 |
| 7.2.133 | FB_READ_137_SNVT_reg_val_ts .....  | 170 |
| 7.2.134 | FB_READ_138_SNVT_volt_ac .....     | 171 |
| 7.2.135 | FB_READ_139_SNVT_amp_ac .....      | 172 |
| 7.2.136 | FB_READ_143_SNVT_turbidity .....   | 173 |
| 7.2.137 | FB_READ_144_SNVT_turbidity_f ..... | 174 |
| 7.2.138 | FB_READ_145_SNVT_hvac_type .....   | 174 |
| 7.2.139 | FB_READ_146_SNVT_elec_kwh_l .....  | 175 |
| 7.2.140 | FB_READ_147_SNVT_temp_diff_p ..... | 176 |
| 7.2.141 | FB_READ_148_SNVT_ctrl_req .....    | 177 |
| 7.2.142 | FB_READ_149_SNVT_ctrl_resp .....   | 178 |
| 7.2.143 | FB_READ_150_SNVT_ptz .....         | 178 |
| 7.2.144 | FB_READ_151_SNVT_privacyzone ..... | 179 |
| 7.2.145 | FB_READ_152_SNVT_pos_ctrl .....    | 180 |
| 7.2.146 | FB_READ_153_SNVT_enthalpy .....    | 181 |
| 7.2.147 | FB_READ_154_SNVT_gfci_status ..... | 182 |
| 7.2.148 | FB_READ_155_SNVT_motor_state ..... | 182 |
| 7.2.149 | FB_READ_156_SNVT_pumpset_mn .....  | 183 |
| 7.2.150 | FB_READ_157_SNVT_ex_control .....  | 184 |
| 7.2.151 | FB_READ_158_SNVT_pumpset_sn .....  | 185 |
| 7.2.152 | FB_READ_159_SNVT_pump_sensor ..... | 186 |
| 7.2.153 | FB_READ_160_SNVT_abs_humid .....   | 186 |
| 7.2.154 | FB_READ_161_SNVT_flow_p .....      | 187 |
| 7.2.155 | FB_READ_162_SNVT_dev_c_mode .....  | 188 |
| 7.2.156 | FB_READ_163_SNVT_valve_mode .....  | 189 |
| 7.2.157 | FB_READ_164_SNVT_alarm_2 .....     | 189 |
| 7.2.158 | FB_READ_165_SNVT_state_64 .....    | 190 |
| 7.2.159 | FB_READ_166_SNVT_nv_type .....     | 191 |
| 7.2.160 | FB_READ_168_SNVT_ent_opmode .....  | 192 |
| 7.2.161 | FB_READ_169_SNVT_ent_state .....   | 193 |
| 7.2.162 | FB_READ_170_SNVT_ent_status .....  | 193 |
| 7.2.163 | FB_READ_171_SNVT_flow_dir .....    | 194 |
| 7.2.164 | FB_READ_172_SNVT_hvac_satsts ..... | 195 |

|         |                                       |     |
|---------|---------------------------------------|-----|
| 7.2.165 | FB_READ_173_SNVT_dev_status.....      | 196 |
| 7.2.166 | FB_READ_174_SNVT_dev_fault .....      | 197 |
| 7.2.167 | FB_READ_175_SNVT_dev_maint .....      | 197 |
| 7.2.168 | FB_READ_176_SNVT_date_event .....     | 198 |
| 7.2.169 | FB_READ_177_SNVT_sched_val.....       | 199 |
| 7.2.170 | FB_READ_180_SNVT_sblnd_state.....     | 200 |
| 7.2.171 | FB_READ_181_SNVT_rac_ctrl .....       | 201 |
| 7.2.172 | FB_READ_182_SNVT_rac_req.....         | 201 |
| 7.2.173 | FB_READ_183_SNVT_count_32 .....       | 202 |
| 7.2.174 | FB_READ_184_SNVT_clothes_w_c .....    | 203 |
| 7.2.175 | FB_READ_185_SNVT_clothes_w_m.....     | 204 |
| 7.2.176 | FB_READ_186_SNVT_clothes_w_s .....    | 205 |
| 7.2.177 | FB_READ_187_SNVT_clothes_w_a.....     | 205 |
| 7.2.178 | FB_READ_188_SNVT_multiplier_s.....    | 206 |
| 7.2.179 | FB_READ_189_SNVT_switch_2 .....       | 207 |
| 7.2.180 | FB_READ_190_SNVT_color_2 .....        | 208 |
| 7.2.181 | FB_READ_191_SNVT_log_status.....      | 209 |
| 7.2.182 | FB_READ_192_SNVT_time_stamp_p .....   | 209 |
| 7.2.183 | FB_READ_193_SNVT_log_fx_request ..... | 210 |
| 7.2.184 | FB_READ_194_SNVT_log_fx_status.....   | 211 |
| 7.2.185 | FB_READ_195_SNVT_log_request .....    | 212 |
| 7.2.186 | FB_READ_196_SNVT_enthalpy_d .....     | 213 |
| 7.2.187 | FB_READ_197_SNVT_amp_ac_mil.....      | 213 |
| 7.2.188 | FB_READ_198_SNVT_time_hour_p.....     | 214 |
| 7.2.189 | FB_READ_199_SNVT_lamp_status.....     | 215 |
| 7.2.190 | FB_READ_200_SNVT_environment .....    | 216 |
| 7.2.191 | FB_READ_201_SNVT_geo_loc .....        | 216 |
| 7.2.192 | FB_SEND_001_SNVT_amp.....             | 217 |
| 7.2.193 | FB_SEND_002_SNVT_amp_mil .....        | 219 |
| 7.2.194 | FB_SEND_003_SNVT_angle .....          | 220 |
| 7.2.195 | FB_SEND_004_SNVT_angle_vel.....       | 221 |
| 7.2.196 | FB_SEND_005_SNVT_btu_kilo.....        | 223 |
| 7.2.197 | FB_SEND_006_SNVT_btu_mega.....        | 224 |
| 7.2.198 | FB_SEND_007_SNVT_char_ascii.....      | 225 |
| 7.2.199 | FB_SEND_008_SNVT_count .....          | 227 |
| 7.2.200 | FB_SEND_009_SNVT_count_inc.....       | 228 |
| 7.2.201 | FB_SEND_011_SNVT_date_day .....       | 229 |
| 7.2.202 | FB_SEND_013_SNVT_elec_kwh.....        | 231 |
| 7.2.203 | FB_SEND_014_SNVT_elec_whr.....        | 232 |
| 7.2.204 | FB_SEND_015_SNVT_flow.....            | 233 |
| 7.2.205 | FB_SEND_016_SNVT_flow_mil.....        | 235 |
| 7.2.206 | FB_SEND_017_SNVT_length .....         | 236 |
| 7.2.207 | FB_SEND_018_SNVT_length_kilo.....     | 237 |
| 7.2.208 | FB_SEND_019_SNVT_length_micr .....    | 239 |
| 7.2.209 | FB_SEND_020_SNVT_length_mil.....      | 240 |
| 7.2.210 | FB_SEND_021_SNVT_lev_cont.....        | 241 |

|         |                              |     |
|---------|------------------------------|-----|
| 7.2.211 | FB_SEND_023_SNVT_mass        | 243 |
| 7.2.212 | FB_SEND_024_SNVT_mass_kilo   | 244 |
| 7.2.213 | FB_SEND_025_SNVT_mass_mega   | 245 |
| 7.2.214 | FB_SEND_026_SNVT_mass_mil    | 247 |
| 7.2.215 | FB_SEND_027_SNVT_power       | 248 |
| 7.2.216 | FB_SEND_028_SNVT_power_kilo  | 249 |
| 7.2.217 | FB_SEND_029_SNVT_ppm         | 251 |
| 7.2.218 | FB_SEND_030_SNVT_press       | 252 |
| 7.2.219 | FB_SEND_031_SNVT_res         | 253 |
| 7.2.220 | FB_SEND_032_SNVT_res_kilo    | 255 |
| 7.2.221 | FB_SEND_033_SNVT_sound_db    | 256 |
| 7.2.222 | FB_SEND_034_SNVT_speed       | 257 |
| 7.2.223 | FB_SEND_035_SNVT_speed_mil   | 259 |
| 7.2.224 | FB_SEND_036_SNVT_str_asc     | 260 |
| 7.2.225 | FB_SEND_037_SNVT_str_int     | 261 |
| 7.2.226 | FB_SEND_038_SNVT_telcom      | 262 |
| 7.2.227 | FB_SEND_039_SNVT_temp        | 264 |
| 7.2.228 | FB_SEND_041_SNVT_vol         | 265 |
| 7.2.229 | FB_SEND_042_SNVT_vol_kilo    | 266 |
| 7.2.230 | FB_SEND_043_SNVT_vol_mil     | 268 |
| 7.2.231 | FB_SEND_044_SNVT_volt        | 269 |
| 7.2.232 | FB_SEND_045_SNVT_volt_dbmv   | 270 |
| 7.2.233 | FB_SEND_046_SNVT_volt_kilo   | 272 |
| 7.2.234 | FB_SEND_047_SNVT_volt_mil    | 273 |
| 7.2.235 | FB_SEND_048_SNVT_amp_f       | 274 |
| 7.2.236 | FB_SEND_049_SNVT_angle_f     | 276 |
| 7.2.237 | FB_SEND_050_SNVT_angle_vel_f | 277 |
| 7.2.238 | FB_SEND_051_SNVT_count_f     | 278 |
| 7.2.239 | FB_SEND_052_SNVT_count_inc_f | 280 |
| 7.2.240 | FB_SEND_053_SNVT_flow_f      | 281 |
| 7.2.241 | FB_SEND_054_SNVT_length_f    | 282 |
| 7.2.242 | FB_SEND_055_SNVT_lev_cont_f  | 284 |
| 7.2.243 | FB_SEND_056_SNVT_mass_f      | 285 |
| 7.2.244 | FB_SEND_057_SNVT_power_f     | 286 |
| 7.2.245 | FB_SEND_058_SNVT_ppm_f       | 288 |
| 7.2.246 | FB_SEND_059_SNVT_press_f     | 289 |
| 7.2.247 | FB_SEND_060_SNVT_res_f       | 290 |
| 7.2.248 | FB_SEND_061_SNVT_sound_db_f  | 292 |
| 7.2.249 | FB_SEND_062_SNVT_speed_f     | 293 |
| 7.2.250 | FB_SEND_063_SNVT_temp_f      | 294 |
| 7.2.251 | FB_SEND_064_SNVT_time_f      | 296 |
| 7.2.252 | FB_SEND_065_SNVT_vol_f       | 297 |
| 7.2.253 | FB_SEND_066_SNVT_volt_f      | 298 |
| 7.2.254 | FB_SEND_067_SNVT_btu_f       | 300 |
| 7.2.255 | FB_SEND_068_SNVT_elec_whr_f  | 301 |
| 7.2.256 | FB_SEND_069_SNVT_config_src  | 302 |



|         |                                    |     |
|---------|------------------------------------|-----|
| 7.2.257 | FB_SEND_070_SNVT_color .....       | 304 |
| 7.2.258 | FB_SEND_071_SNVT_grammage.....     | 305 |
| 7.2.259 | FB_SEND_072_SNVT_grammage_f.....   | 306 |
| 7.2.260 | FB_SEND_073_SNVT_file_req .....    | 307 |
| 7.2.261 | FB_SEND_074_SNVT_file_status.....  | 309 |
| 7.2.262 | FB_SEND_075_SNVT_freq_f.....       | 310 |
| 7.2.263 | FB_SEND_076_SNVT_freq_hz.....      | 311 |
| 7.2.264 | FB_SEND_077_SNVT_freq_kilohz.....  | 313 |
| 7.2.265 | FB_SEND_078_SNVT_freq_milhz .....  | 314 |
| 7.2.266 | FB_SEND_079_SNVT_lux .....         | 315 |
| 7.2.267 | FB_SEND_081_SNVT_lev_percent ..... | 317 |
| 7.2.268 | FB_SEND_082_SNVT_multiplier.....   | 318 |
| 7.2.269 | FB_SEND_083_SNVT_state .....       | 319 |
| 7.2.270 | FB_SEND_084_SNVT_time_stamp .....  | 321 |
| 7.2.271 | FB_SEND_085_SNVT_zerospan .....    | 322 |
| 7.2.272 | FB_SEND_086_SNVT_magcard .....     | 323 |
| 7.2.273 | FB_SEND_087_SNVT_elapsed_tm .....  | 324 |
| 7.2.274 | FB_SEND_088_SNVT_alarm .....       | 325 |
| 7.2.275 | FB_SEND_089_SNVT_currency .....    | 327 |
| 7.2.276 | FB_SEND_090_SNVT_file_pos.....     | 328 |
| 7.2.277 | FB_SEND_091_SNVT_muldiv.....       | 329 |
| 7.2.278 | FB_SEND_092_SNVT_obj_request ..... | 330 |
| 7.2.279 | FB_SEND_093_SNVT_obj_status.....   | 331 |
| 7.2.280 | FB_SEND_094_SNVT_preset.....       | 333 |
| 7.2.281 | FB_SEND_095_SNVT_switch .....      | 334 |
| 7.2.282 | FB_SEND_096_SNVT_trans_table ..... | 335 |
| 7.2.283 | FB_SEND_097_SNVT_override .....    | 336 |
| 7.2.284 | FB_SEND_098_SNVT_pwr_fact .....    | 337 |
| 7.2.285 | FB_SEND_099_SNVT_pwr_fact_f .....  | 339 |
| 7.2.286 | FB_SEND_100_SNVT_density.....      | 340 |
| 7.2.287 | FB_SEND_101_SNVT_density_f.....    | 341 |
| 7.2.288 | FB_SEND_102_SNVT_rpm.....          | 343 |
| 7.2.289 | FB_SEND_103_SNVT_hvac_emerg .....  | 344 |
| 7.2.290 | FB_SEND_104_SNVT_angle_deg .....   | 345 |
| 7.2.291 | FB_SEND_105_SNVT_temp_p.....       | 347 |
| 7.2.292 | FB_SEND_106_SNVT_temp_setpt.....   | 348 |
| 7.2.293 | FB_SEND_107_SNVT_time_sec.....     | 349 |
| 7.2.294 | FB_SEND_108_SNVT_hvac_mode .....   | 350 |
| 7.2.295 | FB_SEND_109_SNVT_occupancy.....    | 352 |
| 7.2.296 | FB_SEND_110_SNVT_area.....         | 353 |
| 7.2.297 | FB_SEND_111_SNVT_hvac_overid.....  | 354 |
| 7.2.298 | FB_SEND_112_SNVT_hvac_status.....  | 355 |
| 7.2.299 | FB_SEND_113_SNVT_press_p .....     | 357 |
| 7.2.300 | FB_SEND_114_SNVT_address .....     | 358 |
| 7.2.301 | FB_SEND_115_SNVT_scene .....       | 359 |
| 7.2.302 | FB_SEND_116_SNVT_scene_cfg.....    | 361 |

|         |                                    |     |
|---------|------------------------------------|-----|
| 7.2.303 | FB_SEND_117_SNVT_setting .....     | 362 |
| 7.2.304 | FB_SEND_118_SNVT_evap_state .....  | 363 |
| 7.2.305 | FB_SEND_119_SNVT_therm_mode.....   | 364 |
| 7.2.306 | FB_SEND_120_SNVT_defr_mode.....    | 365 |
| 7.2.307 | FB_SEND_121_SNVT_defr_term.....    | 367 |
| 7.2.308 | FB_SEND_122_SNVT_defr_state .....  | 368 |
| 7.2.309 | FB_SEND_123_SNVT_time_min .....    | 369 |
| 7.2.310 | FB_SEND_124_SNVT_time_hour.....    | 370 |
| 7.2.311 | FB_SEND_125_SNVT_ph.....           | 372 |
| 7.2.312 | FB_SEND_126_SNVT_ph_f.....         | 373 |
| 7.2.313 | FB_SEND_127_SNVT_chlr_status.....  | 374 |
| 7.2.314 | FB_SEND_128_SNVT_tod_event .....   | 376 |
| 7.2.315 | FB_SEND_129_SNVT_smo_obscur .....  | 377 |
| 7.2.316 | FB_SEND_130_SNVT_fire_test.....    | 378 |
| 7.2.317 | FB_SEND_131_SNVT_temp_ror.....     | 379 |
| 7.2.318 | FB_SEND_132_SNVT_fire_init .....   | 381 |
| 7.2.319 | FB_SEND_133_SNVT_fire_indcte ..... | 382 |
| 7.2.320 | FB_SEND_134_SNVT_time_zone .....   | 383 |
| 7.2.321 | FB_SEND_135_SNVT_earth_pos.....    | 384 |
| 7.2.322 | FB_SEND_136_SNVT_reg_val .....     | 386 |
| 7.2.323 | FB_SEND_137_SNVT_reg_val_ts .....  | 387 |
| 7.2.324 | FB_SEND_138_SNVT_volt_ac.....      | 388 |
| 7.2.325 | FB_SEND_139_SNVT_amp_ac.....       | 389 |
| 7.2.326 | FB_SEND_143_SNVT_turbidity .....   | 391 |
| 7.2.327 | FB_SEND_144_SNVT_turbidity_f.....  | 392 |
| 7.2.328 | FB_SEND_145_SNVT_hvac_type.....    | 393 |
| 7.2.329 | FB_SEND_146_SNVT_elec_kwh_l.....   | 395 |
| 7.2.330 | FB_SEND_147_SNVT_temp_diff_p ..... | 396 |
| 7.2.331 | FB_SEND_148_SNVT_ctrl_req.....     | 397 |
| 7.2.332 | FB_SEND_149_SNVT_ctrl_resp .....   | 399 |
| 7.2.333 | FB_SEND_150_SNVT_ptz .....         | 400 |
| 7.2.334 | FB_SEND_151_SNVT_privacyzone.....  | 401 |
| 7.2.335 | FB_SEND_152_SNVT_pos_ctrl .....    | 402 |
| 7.2.336 | FB_SEND_153_SNVT_enthalpy .....    | 404 |
| 7.2.337 | FB_SEND_154_SNVT_gfci_status.....  | 405 |
| 7.2.338 | FB_SEND_155_SNVT_motor_state.....  | 406 |
| 7.2.339 | FB_SEND_156_SNVT_pumpset_mn .....  | 407 |
| 7.2.340 | FB_SEND_157_SNVT_ex_control .....  | 409 |
| 7.2.341 | FB_SEND_158_SNVT_pumpset_sn .....  | 410 |
| 7.2.342 | FB_SEND_159_SNVT_pump_sensor ..... | 411 |
| 7.2.343 | FB_SEND_160_SNVT_abs_humid .....   | 412 |
| 7.2.344 | FB_SEND_161_SNVT_flow_p.....       | 414 |
| 7.2.345 | FB_SEND_162_SNVT_dev_c_mode .....  | 415 |
| 7.2.346 | FB_SEND_163_SNVT_valve_mode.....   | 416 |
| 7.2.347 | FB_SEND_164_SNVT_alarm_2.....      | 417 |
| 7.2.348 | FB_SEND_165_SNVT_state_64 .....    | 419 |

|         |                                       |     |
|---------|---------------------------------------|-----|
| 7.2.349 | FB_SEND_166_SNVT_nv_type .....        | 420 |
| 7.2.350 | FB_SEND_168_SNVT_ent_opmode .....     | 421 |
| 7.2.351 | FB_SEND_169_SNVT_ent_state .....      | 422 |
| 7.2.352 | FB_SEND_170_SNVT_ent_status .....     | 423 |
| 7.2.353 | FB_SEND_171_SNVT_flow_dir .....       | 425 |
| 7.2.354 | FB_SEND_172_SNVT_hvac_satsts .....    | 426 |
| 7.2.355 | FB_SEND_173_SNVT_dev_status .....     | 427 |
| 7.2.356 | FB_SEND_174_SNVT_dev_fault .....      | 428 |
| 7.2.357 | FB_SEND_175_SNVT_dev_maint .....      | 430 |
| 7.2.358 | FB_SEND_176_SNVT_date_event .....     | 431 |
| 7.2.359 | FB_SEND_177_SNVT_sched_val .....      | 432 |
| 7.2.360 | FB_SEND_180_SNVT_sblnd_state .....    | 433 |
| 7.2.361 | FB_SEND_181_SNVT_rac_ctrl .....       | 435 |
| 7.2.362 | FB_SEND_182_SNVT_rac_req .....        | 436 |
| 7.2.363 | FB_SEND_183_SNVT_count_32 .....       | 437 |
| 7.2.364 | FB_SEND_184_SNVT_clothes_w_c .....    | 438 |
| 7.2.365 | FB_SEND_185_SNVT_clothes_w_m .....    | 440 |
| 7.2.366 | FB_SEND_186_SNVT_clothes_w_s .....    | 441 |
| 7.2.367 | FB_SEND_187_SNVT_clothes_w_a .....    | 442 |
| 7.2.368 | FB_SEND_188_SNVT_multiplier_s .....   | 443 |
| 7.2.369 | FB_SEND_189_SNVT_switch_2 .....       | 445 |
| 7.2.370 | FB_SEND_190_SNVT_color_2 .....        | 446 |
| 7.2.371 | FB_SEND_191_SNVT_log_status .....     | 447 |
| 7.2.372 | FB_SEND_192_SNVT_time_stamp_p .....   | 448 |
| 7.2.373 | FB_SEND_193_SNVT_log_fx_request ..... | 450 |
| 7.2.374 | FB_SEND_194_SNVT_log_fx_status .....  | 451 |
| 7.2.375 | FB_SEND_195_SNVT_log_request .....    | 452 |
| 7.2.376 | FB_SEND_196_SNVT_enthalpy_d .....     | 453 |
| 7.2.377 | FB_SEND_197_SNVT_amp_ac_mil .....     | 455 |
| 7.2.378 | FB_SEND_198_SNVT_time_hour_p .....    | 456 |
| 7.2.379 | FB_SEND_199_SNVT_lamp_status .....    | 457 |
| 7.2.380 | FB_SEND_200_SNVT_environment .....    | 459 |
| 7.2.381 | FB_SEND_201_SNVT_geo_loc .....        | 460 |
| 7.3     | Datentypen .....                      | 461 |
| 7.3.1   | E_LON_ERROR .....                     | 467 |
| 7.3.2   | E_LON_Parameter_Datatypes .....       | 485 |
| 7.3.3   | E_LON_alarm_type_t .....              | 493 |
| 7.3.4   | E_LON_appl_cwc_t .....                | 495 |
| 7.3.5   | E_LON_appl_cwp_t .....                | 495 |
| 7.3.6   | E_LON_appl_cws_t .....                | 496 |
| 7.3.7   | E_LON_appl_rin_t .....                | 497 |
| 7.3.8   | E_LON_boolean_t .....                 | 497 |
| 7.3.9   | E_LON_calendar_type_t .....           | 497 |
| 7.3.10  | E_LON_cam_act_t .....                 | 497 |
| 7.3.11  | E_LON_cam_func_t .....                | 498 |
| 7.3.12  | E_LON_chiller_t .....                 | 498 |

|        |                                     |     |
|--------|-------------------------------------|-----|
| 7.3.13 | E_LON_color_encoding_t.....         | 498 |
| 7.3.14 | E_LON_config_source_t.....          | 499 |
| 7.3.15 | E_LON_control_resp_t.....           | 499 |
| 7.3.16 | E_LON_currency_t.....               | 499 |
| 7.3.17 | E_LON_days_of_week_t.....           | 502 |
| 7.3.18 | E_LON_defrost_mode_t.....           | 502 |
| 7.3.19 | E_LON_defrost_state_t.....          | 503 |
| 7.3.20 | E_LON_defrost_term_t.....           | 503 |
| 7.3.21 | E_LON_device_c_mode_t.....          | 504 |
| 7.3.22 | E_LON_device_select_t.....          | 505 |
| 7.3.23 | E_LON_discrete_levels_t.....        | 505 |
| 7.3.24 | E_LON_emerg_t.....                  | 505 |
| 7.3.25 | E_LON_ent_cmd_t.....                | 506 |
| 7.3.26 | E_LON_ent_opmode_cmd_t.....         | 506 |
| 7.3.27 | E_LON_evap_t.....                   | 507 |
| 7.3.28 | E_LON_ex_control_t.....             | 508 |
| 7.3.29 | E_LON_file_request_t.....           | 508 |
| 7.3.30 | E_LON_file_status_t.....            | 508 |
| 7.3.31 | E_LON_fire_indicator_t.....         | 509 |
| 7.3.32 | E_LON_fire_initiator_t.....         | 510 |
| 7.3.33 | E_LON_fire_test_t.....              | 510 |
| 7.3.34 | E_LON_flow_direction_t.....         | 511 |
| 7.3.35 | E_LON_gfci_status_t.....            | 511 |
| 7.3.36 | E_LON_hvac_hvt_t.....               | 511 |
| 7.3.37 | E_LON_hvac_overid_t.....            | 512 |
| 7.3.38 | E_LON_hvac_t.....                   | 514 |
| 7.3.39 | E_LON_learn_mode_t.....             | 515 |
| 7.3.40 | E_LON_log_status_t.....             | 516 |
| 7.3.41 | E_LON_motor_state_t.....            | 516 |
| 7.3.42 | E_LON_nv_type_category_t.....       | 516 |
| 7.3.43 | E_LON_object_request_t.....         | 517 |
| 7.3.44 | E_LON_occup_t.....                  | 518 |
| 7.3.45 | E_LON_override_t.....               | 518 |
| 7.3.46 | E_LON_pan_dir_t.....                | 519 |
| 7.3.47 | E_LON_priority_level_t.....         | 519 |
| 7.3.48 | E_LON_privacyzone_t.....            | 520 |
| 7.3.49 | E_LON_rail_audio_sensor_type_t..... | 520 |
| 7.3.50 | E_LON_rail_audio_type_t.....        | 521 |
| 7.3.51 | E_LON_reg_val_unit_t.....           | 522 |
| 7.3.52 | E_LON_sblnd_cmd_source_t.....       | 524 |
| 7.3.53 | E_LON_sblnd_error_t.....            | 526 |
| 7.3.54 | E_LON_scene_config_t.....           | 526 |
| 7.3.55 | E_LON_scene_t.....                  | 527 |
| 7.3.56 | E_LON_sec_state_t.....              | 528 |
| 7.3.57 | E_LON_sec_status_t.....             | 529 |
| 7.3.58 | E_LON_setting_t.....                | 530 |

|         |                                |     |
|---------|--------------------------------|-----|
| 7.3.59  | E_LON_switch_state_t.....      | 531 |
| 7.3.60  | E_LON_telcom_states_t.....     | 533 |
| 7.3.61  | E_LON_therm_mode_t.....        | 534 |
| 7.3.62  | E_LON_tilt_dir_t.....          | 534 |
| 7.3.63  | E_LON_unit_temp_t.....         | 534 |
| 7.3.64  | E_LON_valve_mode_t.....        | 535 |
| 7.3.65  | E_LON_zoom_t.....              | 535 |
| 7.3.66  | AuxiliaryStructure.....        | 535 |
| 7.3.67  | ST_LON_Parameter_IN_36B.....   | 560 |
| 7.3.68  | ST_LON_Parameter_OUT_36B.....  | 561 |
| 7.3.69  | ST_ExplicitMessage.....        | 561 |
| 7.3.70  | ST_LON_Communication.....      | 562 |
| 7.3.71  | ST_LON_ParameterInterface..... | 562 |
| 7.3.72  | ST_LON_WriteData.....          | 563 |
| 7.3.73  | ST_Prm.....                    | 563 |
| 7.3.74  | ST_KL6401.....                 | 563 |
| 7.3.75  | ST_LON_AddressTable.....       | 564 |
| 7.3.76  | ST_LON_ConfigTable.....        | 564 |
| 7.3.77  | ST_LON_DomainTable.....        | 565 |
| 7.3.78  | ST_LON_SNVT_alarm.....         | 565 |
| 7.3.79  | ST_LON_SNVT_alarm_2.....       | 566 |
| 7.3.80  | ST_LON_SNVT_chlr_status.....   | 566 |
| 7.3.81  | ST_LON_SNVT_clothes_w_a.....   | 566 |
| 7.3.82  | ST_LON_SNVT_clothes_w_c.....   | 568 |
| 7.3.83  | ST_LON_SNVT_clothes_w_m.....   | 568 |
| 7.3.84  | ST_LON_SNVT_clothes_w_s.....   | 569 |
| 7.3.85  | ST_LON_SNVT_color.....         | 569 |
| 7.3.86  | ST_LON_SNVT_color_2.....       | 569 |
| 7.3.87  | ST_LON_SNVT_ctrl_req.....      | 569 |
| 7.3.88  | ST_LON_SNVT_ctrl_resp.....     | 570 |
| 7.3.89  | ST_LON_SNVT_currency.....      | 570 |
| 7.3.90  | ST_LON_SNVT_date_event.....    | 570 |
| 7.3.91  | ST_LON_SNVT_dev_fault.....     | 570 |
| 7.3.92  | ST_LON_SNVT_dev_maint.....     | 571 |
| 7.3.93  | ST_LON_SNVT_dev_status.....    | 571 |
| 7.3.94  | ST_LON_SNVT_earth_pos.....     | 571 |
| 7.3.95  | ST_LON_SNVT_elapsed_tm.....    | 572 |
| 7.3.96  | ST_LON_SNVT_ent_status.....    | 572 |
| 7.3.97  | ST_LON_SNVT_environment.....   | 573 |
| 7.3.98  | ST_LON_SNVT_ex_control.....    | 574 |
| 7.3.99  | ST_LON_SNVT_file_pos.....      | 574 |
| 7.3.100 | ST_LON_SNVT_file_req.....      | 574 |
| 7.3.101 | ST_LON_SNVT_file_status.....   | 575 |
| 7.3.102 | ST_LON_SNVT_geo_loc.....       | 575 |
| 7.3.103 | ST_LON_SNVT_hvac_overid.....   | 575 |
| 7.3.104 | ST_LON_SNVT_hvac_satsts.....   | 575 |

|          |                                  |            |
|----------|----------------------------------|------------|
| 7.3.105  | ST_LON_SNVT_hvac_status .....    | 576        |
| 7.3.106  | ST_LON_SNVT_lamp_status .....    | 577        |
| 7.3.107  | ST_LON_SNVT_log_fx_request ..... | 577        |
| 7.3.108  | ST_LON_SNVT_log_fx_status .....  | 577        |
| 7.3.109  | ST_LON_SNVT_log_status .....     | 578        |
| 7.3.110  | ST_LON_SNVT_muldiv .....         | 578        |
| 7.3.111  | ST_LON_SNVT_nv_type .....        | 578        |
| 7.3.112  | ST_LON_SNVT_obj_request .....    | 579        |
| 7.3.113  | ST_LON_SNVT_obj_status .....     | 579        |
| 7.3.114  | ST_LON_SNVT_pos_ctrl .....       | 580        |
| 7.3.115  | ST_LON_SNVT_preset .....         | 581        |
| 7.3.116  | ST_LON_SNVT_privacyzone .....    | 581        |
| 7.3.117  | ST_LON_SNVT_ptz .....            | 582        |
| 7.3.118  | ST_LON_SNVT_pump_sensor .....    | 582        |
| 7.3.119  | ST_LON_SNVT_pumpset_mn .....     | 583        |
| 7.3.120  | ST_LON_SNVT_pumpset_sn .....     | 583        |
| 7.3.121  | ST_LON_SNVT_rac_ctrl .....       | 584        |
| 7.3.122  | ST_LON_SNVT_rac_req .....        | 584        |
| 7.3.123  | ST_LON_SNVT_reg_val .....        | 585        |
| 7.3.124  | ST_LON_SNVT_reg_val_ts .....     | 585        |
| 7.3.125  | ST_LON_SNVT_sblnd_state .....    | 585        |
| 7.3.126  | ST_LON_SNVT_scene .....          | 586        |
| 7.3.127  | ST_LON_SNVT_scene_cfg .....      | 586        |
| 7.3.128  | ST_LON_SNVT_setting .....        | 586        |
| 7.3.129  | ST_LON_SNVT_str_int .....        | 587        |
| 7.3.130  | ST_LON_SNVT_switch .....         | 587        |
| 7.3.131  | ST_LON_SNVT_switch_2 .....       | 587        |
| 7.3.132  | ST_LON_SNVT_temp_setpt .....     | 587        |
| 7.3.133  | ST_LON_SNVT_time_zone .....      | 588        |
| 7.3.134  | ST_LON_SNVT_tod_event .....      | 588        |
| 7.3.135  | ST_LON_SNVT_trans_table .....    | 589        |
| 7.3.136  | ST_LON_SNVT_zerospans .....      | 589        |
| 7.3.137  | str_AddressTable .....           | 589        |
| 7.4      | Globale_Variablen_LON .....      | 590        |
| 7.5      | Fehlercodes .....                | 590        |
| <b>8</b> | <b>Anhang .....</b>              | <b>612</b> |
| 8.1      | Automatisches Senden .....       | 612        |
| 8.2      | dwErrorKL .....                  | 613        |
| 8.3      | Beispiele .....                  | 613        |
| 8.4      | SNVT-Variablen (OFF) .....       | 613        |
| 8.5      | Support und Service .....        | 615        |

# 1 Vorwort

## 1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.

Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.

Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentlichte Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

### Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.

Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.

Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

### Marken

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.

Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

### Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

### Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

Weitergabe sowie Vervielfältigung dieses Dokuments, Verwertung und Mitteilung seines Inhalts sind verboten, soweit nicht ausdrücklich gestattet.

Zuwendungen verpflichten zu Schadenersatz. Alle Rechte für den Fall der Patent-, Gebrauchsmuster- oder Geschmacksmustereintragung vorbehalten.

## 1.2 Zu Ihrer Sicherheit

### Sicherheitsbestimmungen

Lesen Sie die folgenden Erklärungen zu Ihrer Sicherheit. Beachten und befolgen Sie stets produktspezifische Sicherheitshinweise, die Sie gegebenenfalls an den entsprechenden Stellen in diesem Dokument vorfinden.

### Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

### Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

### Signalwörter

Im Folgenden werden die Signalwörter eingeordnet, die in der Dokumentation verwendet werden. Um Personen- und Sachschäden zu vermeiden, lesen und befolgen Sie die Sicherheits- und Warnhinweise.

### Warnungen vor Personenschäden

#### **GEFAHR**

Es besteht eine Gefährdung mit hohem Risikograd, die den Tod oder eine schwere Verletzung zur Folge hat.

#### **WARNUNG**

Es besteht eine Gefährdung mit mittlerem Risikograd, die den Tod oder eine schwere Verletzung zur Folge haben kann.

#### **VORSICHT**

Es besteht eine Gefährdung mit geringem Risikograd, die eine mittelschwere oder leichte Verletzung zur Folge haben kann.

### Warnung vor Umwelt- oder Sachschäden

#### **HINWEIS**

Es besteht eine mögliche Schädigung für Umwelt, Geräte oder Daten.

### Information zum Umgang mit dem Produkt



Diese Information beinhaltet z. B.:  
Handlungsempfehlungen, Hilfestellungen oder weiterführende Informationen zum Produkt.



## 1.3 Hinweise zur Informationssicherheit

Die Produkte der Beckhoff Automation GmbH & Co. KG (Beckhoff) sind, sofern sie online zu erreichen sind, mit Security-Funktionen ausgestattet, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen. Trotz der Security-Funktionen sind die Erstellung, Implementierung und ständige Aktualisierung eines ganzheitlichen Security-Konzepts für den Betrieb notwendig, um die jeweilige Anlage, das System, die Maschine und die Netzwerke gegen Cyber-Bedrohungen zu schützen. Die von Beckhoff verkauften Produkte bilden dabei nur einen Teil des gesamtheitlichen Security-Konzepts. Der Kunde ist dafür verantwortlich, dass unbefugte Zugriffe durch Dritte auf seine Anlagen, Systeme, Maschinen und Netzwerke verhindert werden. Letztere sollten nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn entsprechende Schutzmaßnahmen eingerichtet wurden.

Zusätzlich sollten die Empfehlungen von Beckhoff zu entsprechenden Schutzmaßnahmen beachtet werden. Weiterführende Informationen über Informationssicherheit und Industrial Security finden Sie in unserem <https://www.beckhoff.de/secguide>.

Die Produkte und Lösungen von Beckhoff werden ständig weiterentwickelt. Dies betrifft auch die Security-Funktionen. Aufgrund der stetigen Weiterentwicklung empfiehlt Beckhoff ausdrücklich, die Produkte ständig auf dem aktuellen Stand zu halten und nach Bereitstellung von Updates diese auf die Produkte aufzuspielen. Die Verwendung veralteter oder nicht mehr unterstützter Produktversionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Hinweise zur Informationssicherheit zu Produkten von Beckhoff informiert zu sein, abonnieren Sie den RSS Feed unter <https://www.beckhoff.de/secinfo>.

## 2 Einleitung

Die LON-Bibliothek ist eine umfangreiche TwinCAT SPS Bibliothek zum Datenaustausch mit LON Geräten. Die Kommunikation mit diesen Geräten wird über SNVTs (**S**tandard **N**etwork **V**ariable **T**ypes) realisiert. Diese SNVTs sind in der LONMark definiert (siehe auch [LONMARK](#) und [LONMARK Resource Files](#)). Für jeden SNVT gibt es je einen Funktionsbaustein zum Senden und zum Empfangen.

Alle Funktionsblöcke aus der Bibliothek müssen in derselben Task aufgerufen werden.

Diese Bibliothek ist nur in Verbindung mit einer [KL6401](#) (LON-Masterklemme) einzusetzen.

Die SNVT müssen mit der [KS2000 \[► 27\]](#) in der Klemme konfiguriert werden.

Die Verknüpfung der SNVTs erfolgt mit einem LON-Konfigurationstool (z.B. LON-Maker von Echelon). Dieses Tool wird nicht von Beckhoff geliefert.

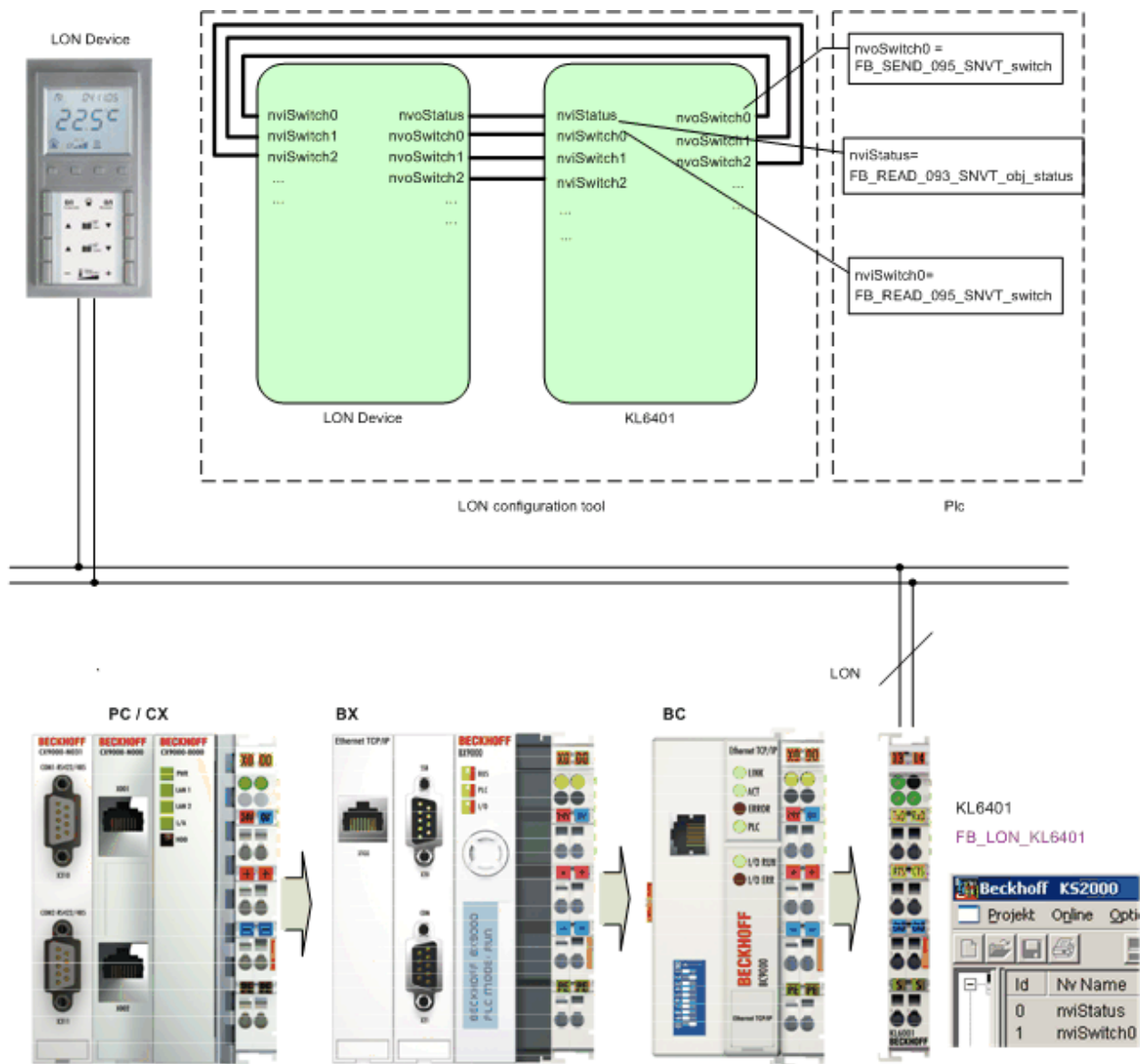
### 3 Zielgruppen

Für den Nutzer dieser Bibliothek werden folgende Grundkenntnisse vorausgesetzt:

- TwinCAT PLC Control
- TwinCAT System Manager
- KS2000
- PC und Netzwerkkennnisse
- Aufbau und Eigenschaften der Beckhoff Embedded-PC und deren Busklemmensystem
- Technologie von LON Geräten / LON Konfigurationstools
- Einschlägige Sicherheitsvorschriften der technischen Gebäudeausrüstung

Diese Softwarebibliothek ist für Gebäudeautomation-Systempartner der Beckhoff Automation GmbH & Co. KG. Die Systempartner sind tätig in dem Bereich Gebäudeautomation und beschäftigen sich mit Errichtung, Inbetriebsetzung, Erweiterung, Wartung und Service von mess-, steuer- und regelungstechnischen Anlagen der technischen Gebäudeausrüstung.

## 4 LON



Jedes LON Gerät stellt Netzwerkvariablen (SNVTs) zur Kommunikation mit anderen Geräten zur Verfügung. Anhand der Gerätedokumentation oder dem Xif File (maschinenlesbare Darstellung aller verwendete SNVTs) werden die gewünschten Variablen ausgewählt und mit der [KS2000](#) [► 27] in die [KL6401](#) eingetragen. Mit der [KS2000](#) kann nun ebenfalls ein Xif File erstellt werden. Beide Xif Files mit einem LON Konfigurationstool (wird nicht von Beckhoff angeboten) einlesen und verbinden. Nähere Informationen dazu bitte dem jeweiligen Tool entnehmen.

Auf SPS Seite muss für jeden in der [KL6401](#) eingetragenen SNVT, der dazu passende Funktionsbaustein programmiert werden. Über diese Funktionsbausteine kann nun auf die SNVTs der Geräte zugegriffen werden.

### 4.1 Unterstützte SNVTs

Allgemeine Informationen zu SNVTs finden Sie im Internet unter [www.lonmark.org](http://www.lonmark.org).

Die [KL6401](#) unterstützt die folgenden SNVTs (LonMark-Datentypen).

| SNVT ID | SNVT NAME | Read        | Send         | Zusatzinfo |
|---------|-----------|-------------|--------------|------------|
| 1       | SNVT_amp  | Read [► 70] | Send [► 217] |            |

| SNVT ID | SNVT NAME        | Read                        | Send                         | Zusatzinfo                         |
|---------|------------------|-----------------------------|------------------------------|------------------------------------|
| 2       | SNVT_amp_mil     | <a href="#">Read [▶ 70]</a> | <a href="#">Send [▶ 219]</a> |                                    |
| 3       | SNVT_angle       | <a href="#">Read [▶ 71]</a> | <a href="#">Send [▶ 220]</a> |                                    |
| 4       | SNVT_angle_vel   | <a href="#">Read [▶ 72]</a> | <a href="#">Send [▶ 221]</a> |                                    |
| 5       | SNVT_btu_kilo    | <a href="#">Read [▶ 73]</a> | <a href="#">Send [▶ 223]</a> |                                    |
| 6       | SNVT_btu_mega    | <a href="#">Read [▶ 73]</a> | <a href="#">Send [▶ 224]</a> |                                    |
| 7       | SNVT_char_ascii  | <a href="#">Read [▶ 74]</a> | <a href="#">Send [▶ 225]</a> |                                    |
| 8       | SNVT_count       | <a href="#">Read [▶ 75]</a> | <a href="#">Send [▶ 227]</a> |                                    |
| 9       | SNVT_count_inc   | <a href="#">Read [▶ 76]</a> | <a href="#">Send [▶ 228]</a> |                                    |
| 10      | SNVT_date_cal    |                             |                              | SNVT veraltet, nicht implementiert |
| 11      | SNVT_date_day    | <a href="#">Read [▶ 76]</a> | <a href="#">Send [▶ 229]</a> |                                    |
| 12      | SNVT_date_time   |                             |                              | SNVT veraltet, nicht implementiert |
| 13      | SNVT_elec_kwh    | <a href="#">Read [▶ 77]</a> | <a href="#">Send [▶ 231]</a> |                                    |
| 14      | SNVT_elec_whr    | <a href="#">Read [▶ 78]</a> | <a href="#">Send [▶ 232]</a> |                                    |
| 15      | SNVT_flow        | <a href="#">Read [▶ 79]</a> | <a href="#">Send [▶ 233]</a> |                                    |
| 16      | SNVT_flow_mil    | <a href="#">Read [▶ 80]</a> | <a href="#">Send [▶ 235]</a> |                                    |
| 17      | SNVT_length      | <a href="#">Read [▶ 80]</a> | <a href="#">Send [▶ 236]</a> |                                    |
| 18      | SNVT_length_kilo | <a href="#">Read [▶ 81]</a> | <a href="#">Send [▶ 237]</a> |                                    |
| 19      | SNVT_length_micr | <a href="#">Read [▶ 82]</a> | <a href="#">Send [▶ 239]</a> |                                    |
| 20      | SNVT_length_mil  | <a href="#">Read [▶ 83]</a> | <a href="#">Send [▶ 240]</a> |                                    |
| 21      | SNVT_lev_cont    | <a href="#">Read [▶ 83]</a> | <a href="#">Send [▶ 241]</a> |                                    |
| 22      | SNVT_lev_disc    |                             |                              | SNVT veraltet, nicht implementiert |
| 23      | SNVT_mass        | <a href="#">Read [▶ 84]</a> | <a href="#">Send [▶ 243]</a> |                                    |
| 24      | SNVT_mass_kilo   | <a href="#">Read [▶ 85]</a> | <a href="#">Send [▶ 244]</a> |                                    |
| 25      | SNVT_mass_mega   | <a href="#">Read [▶ 86]</a> | <a href="#">Send [▶ 245]</a> |                                    |
| 26      | SNVT_mass_mil    | <a href="#">Read [▶ 86]</a> | <a href="#">Send [▶ 247]</a> |                                    |
| 27      | SNVT_power       | <a href="#">Read [▶ 87]</a> | <a href="#">Send [▶ 248]</a> |                                    |
| 28      | SNVT_power_kilo  | <a href="#">Read [▶ 88]</a> | <a href="#">Send [▶ 249]</a> |                                    |
| 29      | SNVT_ppm         | <a href="#">Read [▶ 89]</a> | <a href="#">Send [▶ 251]</a> |                                    |
| 30      | SNVT_press       | <a href="#">Read [▶ 89]</a> | <a href="#">Send [▶ 252]</a> |                                    |
| 31      | SNVT_res         | <a href="#">Read [▶ 90]</a> | <a href="#">Send [▶ 253]</a> |                                    |
| 32      | SNVT_res_kilo    | <a href="#">Read [▶ 91]</a> | <a href="#">Send [▶ 255]</a> |                                    |
| 33      | SNVT_sound_db    | <a href="#">Read [▶ 92]</a> | <a href="#">Send [▶ 256]</a> |                                    |
| 34      | SNVT_speed       | <a href="#">Read [▶ 92]</a> | <a href="#">Send [▶ 257]</a> |                                    |
| 35      | SNVT_speed_mil   | <a href="#">Read [▶ 93]</a> | <a href="#">Send [▶ 259]</a> |                                    |
| 36      | SNVT_str_asc     | <a href="#">Read [▶ 94]</a> | <a href="#">Send [▶ 260]</a> |                                    |
| 37      | SNVT_str_int     | <a href="#">Read [▶ 95]</a> | <a href="#">Send [▶ 261]</a> |                                    |
| 38      | SNVT_telcom      | <a href="#">Read [▶ 95]</a> | <a href="#">Send [▶ 262]</a> |                                    |
| 39      | SNVT_temp        | <a href="#">Read [▶ 96]</a> | <a href="#">Send [▶ 264]</a> |                                    |
| 40      | SNVT_time_passed |                             |                              | SNVT veraltet, nicht implementiert |
| 41      | SNVT_vol         | <a href="#">Read [▶ 97]</a> | <a href="#">Send [▶ 265]</a> |                                    |
| 42      | SNVT_vol_kilo    | <a href="#">Read [▶ 98]</a> | <a href="#">Send [▶ 266]</a> |                                    |

| SNVT ID | SNVT NAME        | Read         | Send         | Zusatzinfo                            |
|---------|------------------|--------------|--------------|---------------------------------------|
| 43      | SNVT_vol_mil     | Read [▶ 99]  | Send [▶ 268] |                                       |
| 44      | SNVT_volt        | Read [▶ 99]  | Send [▶ 269] |                                       |
| 45      | SNVT_volt_dbmv   | Read [▶ 100] | Send [▶ 270] |                                       |
| 46      | SNVT_volt_kilo   | Read [▶ 101] | Send [▶ 272] |                                       |
| 47      | SNVT_volt_mil    | Read [▶ 102] | Send [▶ 273] |                                       |
| 48      | SNVT_amp_f       | Read [▶ 102] | Send [▶ 274] |                                       |
| 49      | SNVT_angle_f     | Read [▶ 103] | Send [▶ 276] |                                       |
| 50      | SNVT_angle_vel_f | Read [▶ 104] | Send [▶ 277] |                                       |
| 51      | SNVT_count_f     | Read [▶ 105] | Send [▶ 278] |                                       |
| 52      | SNVT_count_inc_f | Read [▶ 105] | Send [▶ 280] |                                       |
| 53      | SNVT_flow_f      | Read [▶ 106] | Send [▶ 281] |                                       |
| 54      | SNVT_length_f    | Read [▶ 107] | Send [▶ 282] |                                       |
| 55      | SNVT_lev_cont_f  | Read [▶ 108] | Send [▶ 284] |                                       |
| 56      | SNVT_mass_f      | Read [▶ 108] | Send [▶ 285] |                                       |
| 57      | SNVT_power_f     | Read [▶ 109] | Send [▶ 286] |                                       |
| 58      | SNVT_ppm_f       | Read [▶ 110] | Send [▶ 288] |                                       |
| 59      | SNVT_press_f     | Read [▶ 111] | Send [▶ 289] |                                       |
| 60      | SNVT_res_f       | Read [▶ 111] | Send [▶ 290] |                                       |
| 61      | SNVT_sound_db_f  | Read [▶ 112] | Send [▶ 292] |                                       |
| 62      | SNVT_speed_f     | Read [▶ 113] | Send [▶ 293] |                                       |
| 63      | SNVT_temp_f      | Read [▶ 114] | Send [▶ 294] |                                       |
| 64      | SNVT_time_f      | Read [▶ 114] | Send [▶ 296] |                                       |
| 65      | SNVT_vol_f       | Read [▶ 115] | Send [▶ 297] |                                       |
| 66      | SNVT_volt_f      | Read [▶ 116] | Send [▶ 298] |                                       |
| 67      | SNVT_btu_f       | Read [▶ 117] | Send [▶ 300] |                                       |
| 68      | SNVT_elec_whr_f  | Read [▶ 117] | Send [▶ 301] |                                       |
| 69      | SNVT_config_src  | Read [▶ 118] | Send [▶ 302] |                                       |
| 70      | SNVT_color       | Read [▶ 119] | Send [▶ 304] |                                       |
| 71      | SNVT_grammage    | Read [▶ 120] | Send [▶ 305] |                                       |
| 72      | SNVT_grammage_f  | Read [▶ 120] | Send [▶ 306] |                                       |
| 73      | SNVT_file_req    | Read [▶ 121] | Send [▶ 307] | In Arbeit, bitte noch nicht benutzen. |
| 74      | SNVT_file_status | Read [▶ 122] | Send [▶ 309] |                                       |
| 75      | SNVT_freq_f      | Read [▶ 123] | Send [▶ 310] |                                       |
| 76      | SNVT_freq_hz     | Read [▶ 123] | Send [▶ 311] |                                       |
| 77      | SNVT_freq_kilohz | Read [▶ 124] | Send [▶ 313] |                                       |
| 78      | SNVT_freq_milhz  | Read [▶ 125] | Send [▶ 314] |                                       |
| 79      | SNVT_lux         | Read [▶ 126] | Send [▶ 315] |                                       |
| 80      | SNVT_ISO_7811    |              |              | SNVT veraltet, nicht implementiert    |
| 81      | SNVT_lev_percent | Read [▶ 126] | Send [▶ 317] |                                       |
| 82      | SNVT_multiplier  | Read [▶ 127] | Send [▶ 318] |                                       |
| 83      | SNVT_state       | Read [▶ 128] | Send [▶ 319] |                                       |
| 84      | SNVT_time_stamp  | Read [▶ 129] | Send [▶ 321] |                                       |

| SNVT ID | SNVT NAME        | Read                         | Send                         | Zusatzinfo |
|---------|------------------|------------------------------|------------------------------|------------|
| 85      | SNVT_zerospan    | <a href="#">Read [▶ 130]</a> | <a href="#">Send [▶ 322]</a> |            |
| 86      | SNVT_magcard     | <a href="#">Read [▶ 130]</a> | <a href="#">Send [▶ 323]</a> |            |
| 87      | SNVT_elapsed_tm  | <a href="#">Read [▶ 131]</a> | <a href="#">Send [▶ 324]</a> |            |
| 88      | SNVT_alarm       | <a href="#">Read [▶ 132]</a> | <a href="#">Send [▶ 325]</a> |            |
| 89      | SNVT_currency    | <a href="#">Read [▶ 133]</a> | <a href="#">Send [▶ 327]</a> |            |
| 90      | SNVT_file_pos    | <a href="#">Read [▶ 134]</a> | <a href="#">Send [▶ 328]</a> |            |
| 91      | SNVT_muldiv      | <a href="#">Read [▶ 134]</a> | <a href="#">Send [▶ 329]</a> |            |
| 92      | SNVT_obj_request | <a href="#">Read [▶ 135]</a> | <a href="#">Send [▶ 330]</a> |            |
| 93      | SNVT_obj_status  | <a href="#">Read [▶ 136]</a> | <a href="#">Send [▶ 331]</a> |            |
| 94      | SNVT_preset      | <a href="#">Read [▶ 137]</a> | <a href="#">Send [▶ 333]</a> |            |
| 95      | SNVT_switch      | <a href="#">Read [▶ 138]</a> | <a href="#">Send [▶ 334]</a> |            |
| 96      | SNVT_trans_table | <a href="#">Read [▶ 138]</a> | <a href="#">Send [▶ 335]</a> |            |
| 97      | SNVT_override    | <a href="#">Read [▶ 139]</a> | <a href="#">Send [▶ 336]</a> |            |
| 98      | SNVT_pwr_fact    | <a href="#">Read [▶ 140]</a> | <a href="#">Send [▶ 337]</a> |            |
| 99      | SNVT_pwr_fact_f  | <a href="#">Read [▶ 141]</a> | <a href="#">Send [▶ 339]</a> |            |
| 100     | SNVT_density     | <a href="#">Read [▶ 142]</a> | <a href="#">Send [▶ 340]</a> |            |
| 101     | SNVT_density_f   | <a href="#">Read [▶ 142]</a> | <a href="#">Send [▶ 341]</a> |            |
| 102     | SNVT_rpm         | <a href="#">Read [▶ 143]</a> | <a href="#">Send [▶ 343]</a> |            |
| 103     | SNVT_hvac_emerg  | <a href="#">Read [▶ 144]</a> | <a href="#">Send [▶ 344]</a> |            |
| 104     | SNVT_angle_deg   | <a href="#">Read [▶ 145]</a> | <a href="#">Send [▶ 345]</a> |            |
| 105     | SNVT_temp_p      | <a href="#">Read [▶ 145]</a> | <a href="#">Send [▶ 347]</a> |            |
| 106     | SNVT_temp_setpt  | <a href="#">Read [▶ 146]</a> | <a href="#">Send [▶ 348]</a> |            |
| 107     | SNVT_time_sec    | <a href="#">Read [▶ 147]</a> | <a href="#">Send [▶ 349]</a> |            |
| 108     | SNVT_hvac_mode   | <a href="#">Read [▶ 148]</a> | <a href="#">Send [▶ 350]</a> |            |
| 109     | SNVT_occupancy   | <a href="#">Read [▶ 148]</a> | <a href="#">Send [▶ 352]</a> |            |
| 110     | SNVT_area        | <a href="#">Read [▶ 149]</a> | <a href="#">Send [▶ 353]</a> |            |
| 111     | SNVT_hvac_overid | <a href="#">Read [▶ 150]</a> | <a href="#">Send [▶ 354]</a> |            |
| 112     | SNVT_hvac_status | <a href="#">Read [▶ 151]</a> | <a href="#">Send [▶ 355]</a> |            |
| 113     | SNVT_press_p     | <a href="#">Read [▶ 152]</a> | <a href="#">Send [▶ 357]</a> |            |
| 114     | SNVT_address     | <a href="#">Read [▶ 152]</a> | <a href="#">Send [▶ 358]</a> |            |
| 115     | SNVT_scene       | <a href="#">Read [▶ 153]</a> | <a href="#">Send [▶ 359]</a> |            |
| 116     | SNVT_scene_cfg   | <a href="#">Read [▶ 154]</a> | <a href="#">Send [▶ 361]</a> |            |
| 117     | SNVT_setting     | <a href="#">Read [▶ 155]</a> | <a href="#">Send [▶ 362]</a> |            |
| 118     | SNVT_evap_state  | <a href="#">Read [▶ 155]</a> | <a href="#">Send [▶ 363]</a> |            |
| 119     | SNVT_therm_mode  | <a href="#">Read [▶ 156]</a> | <a href="#">Send [▶ 364]</a> |            |
| 120     | SNVT_defr_mode   | <a href="#">Read [▶ 157]</a> | <a href="#">Send [▶ 365]</a> |            |
| 121     | SNVT_defr_term   | <a href="#">Read [▶ 158]</a> | <a href="#">Send [▶ 367]</a> |            |
| 122     | SNVT_defr_state  | <a href="#">Read [▶ 159]</a> | <a href="#">Send [▶ 368]</a> |            |
| 123     | SNVT_time_min    | <a href="#">Read [▶ 159]</a> | <a href="#">Send [▶ 369]</a> |            |
| 124     | SNVT_time_hour   | <a href="#">Read [▶ 160]</a> | <a href="#">Send [▶ 370]</a> |            |
| 125     | SNVT_ph          | <a href="#">Read [▶ 161]</a> | <a href="#">Send [▶ 372]</a> |            |
| 126     | SNVT_ph_f        | <a href="#">Read [▶ 162]</a> | <a href="#">Send [▶ 373]</a> |            |
| 127     | SNVT_chlr_status | <a href="#">Read [▶ 162]</a> | <a href="#">Send [▶ 374]</a> |            |

| SNVT ID | SNVT NAME        | Read                         | Send                         | Zusatzinfo |
|---------|------------------|------------------------------|------------------------------|------------|
| 128     | SNVT_tod_event   | <a href="#">Read [▶ 163]</a> | <a href="#">Send [▶ 376]</a> |            |
| 129     | SNVT_smo_obscur  | <a href="#">Read [▶ 164]</a> | <a href="#">Send [▶ 377]</a> |            |
| 130     | SNVT_fire_test   | <a href="#">Read [▶ 165]</a> | <a href="#">Send [▶ 378]</a> |            |
| 131     | SNVT_temp_ror    | <a href="#">Read [▶ 166]</a> | <a href="#">Send [▶ 379]</a> |            |
| 132     | SNVT_fire_init   | <a href="#">Read [▶ 166]</a> | <a href="#">Send [▶ 381]</a> |            |
| 133     | SNVT_fire_indcte | <a href="#">Read [▶ 167]</a> | <a href="#">Send [▶ 382]</a> |            |
| 134     | SNVT_time_zone   | <a href="#">Read [▶ 168]</a> | <a href="#">Send [▶ 383]</a> |            |
| 135     | SNVT_earth_pos   | <a href="#">Read [▶ 169]</a> | <a href="#">Send [▶ 384]</a> |            |
| 136     | SNVT_reg_val     | <a href="#">Read [▶ 170]</a> | <a href="#">Send [▶ 386]</a> |            |
| 137     | SNVT_reg_val_ts  | <a href="#">Read [▶ 170]</a> | <a href="#">Send [▶ 387]</a> |            |
| 138     | SNVT_volt_ac     | <a href="#">Read [▶ 171]</a> | <a href="#">Send [▶ 388]</a> |            |
| 139     | SNVT_amp_ac      | <a href="#">Read [▶ 172]</a> | <a href="#">Send [▶ 389]</a> |            |
|         |                  |                              |                              |            |
| 143     | SNVT_turbidity   | <a href="#">Read [▶ 173]</a> | <a href="#">Send [▶ 391]</a> |            |
| 144     | SNVT_turbidity_f | <a href="#">Read [▶ 174]</a> | <a href="#">Send [▶ 392]</a> |            |
| 145     | SNVT_hvac_type   | <a href="#">Read [▶ 174]</a> | <a href="#">Send [▶ 393]</a> |            |
| 146     | SNVT_elec_kwh_l  | <a href="#">Read [▶ 175]</a> | <a href="#">Send [▶ 395]</a> |            |
| 147     | SNVT_temp_diff_p | <a href="#">Read [▶ 176]</a> | <a href="#">Send [▶ 396]</a> |            |
| 148     | SNVT_ctrl_req    | <a href="#">Read [▶ 177]</a> | <a href="#">Send [▶ 397]</a> |            |
| 149     | SNVT_ctrl_resp   | <a href="#">Read [▶ 178]</a> | <a href="#">Send [▶ 399]</a> |            |
| 150     | SNVT_ptz         | <a href="#">Read [▶ 178]</a> | <a href="#">Send [▶ 400]</a> |            |
| 151     | SNVT_privacyzone | <a href="#">Read [▶ 179]</a> | <a href="#">Send [▶ 401]</a> |            |
| 152     | SNVT_pos_ctrl    | <a href="#">Read [▶ 180]</a> | <a href="#">Send [▶ 402]</a> |            |
| 153     | SNVT_enthalpy    | <a href="#">Read [▶ 181]</a> | <a href="#">Send [▶ 404]</a> |            |
| 154     | SNVT_gfci_status | <a href="#">Read [▶ 182]</a> | <a href="#">Send [▶ 405]</a> |            |
| 155     | SNVT_motor_state | <a href="#">Read [▶ 182]</a> | <a href="#">Send [▶ 406]</a> |            |
| 156     | SNVT_pumpset_mn  | <a href="#">Read [▶ 183]</a> | <a href="#">Send [▶ 407]</a> |            |
| 157     | SNVT_ex_control  | <a href="#">Read [▶ 184]</a> | <a href="#">Send [▶ 409]</a> |            |
| 158     | SNVT_pumpset_sn  | <a href="#">Read [▶ 185]</a> | <a href="#">Send [▶ 410]</a> |            |
| 159     | SNVT_pump_sensor | <a href="#">Read [▶ 186]</a> | <a href="#">Send [▶ 411]</a> |            |
| 160     | SNVT_abs_humid   | <a href="#">Read [▶ 186]</a> | <a href="#">Send [▶ 412]</a> |            |
| 161     | SNVT_flow_p      | <a href="#">Read [▶ 187]</a> | <a href="#">Send [▶ 414]</a> |            |
| 162     | SNVT_dev_c_mode  | <a href="#">Read [▶ 188]</a> | <a href="#">Send [▶ 415]</a> |            |
| 163     | SNVT_valve_mode  | <a href="#">Read [▶ 189]</a> | <a href="#">Send [▶ 416]</a> |            |
| 164     | SNVT_alarm_2     | <a href="#">Read [▶ 189]</a> | <a href="#">Send [▶ 417]</a> |            |
| 165     | SNVT_state_64    | <a href="#">Read [▶ 190]</a> | <a href="#">Send [▶ 419]</a> |            |
| 166     | SNVT_nv_type     | <a href="#">Read [▶ 191]</a> | <a href="#">Send [▶ 420]</a> |            |
|         |                  |                              |                              |            |
| 168     | SNVT_ent_opmode  | <a href="#">Read [▶ 192]</a> | <a href="#">Send [▶ 421]</a> |            |
| 169     | SNVT_ent_state   | <a href="#">Read [▶ 193]</a> | <a href="#">Send [▶ 422]</a> |            |
| 170     | SNVT_ent_status  | <a href="#">Read [▶ 193]</a> | <a href="#">Send [▶ 423]</a> |            |
| 171     | SNVT_flow_dir    | <a href="#">Read [▶ 194]</a> | <a href="#">Send [▶ 425]</a> |            |
| 172     | SNVT_hvac_satsts | <a href="#">Read [▶ 195]</a> | <a href="#">Send [▶ 426]</a> |            |



| SNVT ID | SNVT NAME           | Read                         | Send                         | Zusatzinfo                         |
|---------|---------------------|------------------------------|------------------------------|------------------------------------|
| 173     | SNVT_dev_status     | <a href="#">Read [▶ 196]</a> | <a href="#">Send [▶ 427]</a> |                                    |
| 174     | SNVT_dev_fault      | <a href="#">Read [▶ 197]</a> | <a href="#">Send [▶ 428]</a> |                                    |
| 175     | SNVT_dev_maint      | <a href="#">Read [▶ 197]</a> | <a href="#">Send [▶ 430]</a> |                                    |
| 176     | SNVT_date_event     | <a href="#">Read [▶ 198]</a> | <a href="#">Send [▶ 431]</a> |                                    |
| 177     | SNVT_sched_val      | <a href="#">Read [▶ 199]</a> | <a href="#">Send [▶ 432]</a> |                                    |
| 178     | SNVT_sec_state      |                              |                              | SNVT veraltet, nicht implementiert |
| 179     | SNVT_sec_status     |                              |                              | SNVT veraltet, nicht implementiert |
| 180     | SNVT_sbldnd_state   | <a href="#">Read [▶ 200]</a> | <a href="#">Send [▶ 433]</a> |                                    |
| 181     | SNVT_rac_ctrl       | <a href="#">Read [▶ 201]</a> | <a href="#">Send [▶ 435]</a> |                                    |
| 182     | SNVT_rac_req        | <a href="#">Read [▶ 201]</a> | <a href="#">Send [▶ 436]</a> |                                    |
| 183     | SNVT_count_32       | <a href="#">Read [▶ 202]</a> | <a href="#">Send [▶ 437]</a> |                                    |
| 184     | SNVT_clothes_w_c    | <a href="#">Read [▶ 203]</a> | <a href="#">Send [▶ 438]</a> |                                    |
| 185     | SNVT_clothes_w_m    | <a href="#">Read [▶ 204]</a> | <a href="#">Send [▶ 440]</a> |                                    |
| 186     | SNVT_clothes_w_s    | <a href="#">Read [▶ 205]</a> | <a href="#">Send [▶ 441]</a> |                                    |
| 187     | SNVT_clothes_w_a    | <a href="#">Read [▶ 205]</a> | <a href="#">Send [▶ 442]</a> |                                    |
| 188     | SNVT_multiplier_s   | <a href="#">Read [▶ 206]</a> | <a href="#">Send [▶ 443]</a> |                                    |
| 189     | SNVT_switch_2       | <a href="#">Read [▶ 207]</a> | <a href="#">Send [▶ 445]</a> |                                    |
| 190     | SNVT_color_2        | <a href="#">Read [▶ 208]</a> | <a href="#">Send [▶ 446]</a> |                                    |
| 191     | SNVT_log_status     | <a href="#">Read [▶ 209]</a> | <a href="#">Send [▶ 447]</a> |                                    |
| 192     | SNVT_time_stamp_p   | <a href="#">Read [▶ 209]</a> | <a href="#">Send [▶ 448]</a> |                                    |
| 193     | SNVT_log_fx_request | <a href="#">Read [▶ 210]</a> | <a href="#">Send [▶ 450]</a> |                                    |
| 194     | SNVT_log_fx_status  | <a href="#">Read [▶ 211]</a> | <a href="#">Send [▶ 451]</a> |                                    |
| 195     | SNVT_log_request    | <a href="#">Read [▶ 212]</a> | <a href="#">Send [▶ 452]</a> |                                    |
| 196     | SNVT_enthalpy_d     | <a href="#">Read [▶ 213]</a> | <a href="#">Send [▶ 453]</a> |                                    |
| 197     | SNVT_amp_ac_mil     | <a href="#">Read [▶ 213]</a> | <a href="#">Send [▶ 455]</a> |                                    |
| 198     | SNVT_time_hour_p    | <a href="#">Read [▶ 214]</a> | <a href="#">Send [▶ 456]</a> |                                    |
| 199     | SNVT_lamp_status    | <a href="#">Read [▶ 215]</a> | <a href="#">Send [▶ 457]</a> |                                    |
| 200     | SNVT_environment    | <a href="#">Read [▶ 216]</a> | <a href="#">Send [▶ 459]</a> |                                    |
| 201     | SNVT_geo_loc        | <a href="#">Read [▶ 216]</a> | <a href="#">Send [▶ 460]</a> |                                    |

## 4.2 Arbeitsweise der KL6401

### Senden

Die [KL6401](#) verschickt SNVT-Variablen einzeln. Das bedeutet, eine zur KL6401 übertragene SNVT-Variable wird von der Klemme einzeln ins LON-Netzwerk gesendet. Erst wenn diese erfolgreich verschickt wurde, können Sie die nächste SNVT-Variable zur KL6401 übertragen.

### Empfangen

Die KL6401 besitzt 2 Pufferbereiche, den Telegramm-Puffer und den Index-Puffer.

Die Eingangsvariablen eines empfangenen LON-Telegramms werden im Telegramm-Puffer abgelegt. Außerdem wird die Index-Nummer des eingehenden Telegramms in den Index-Puffer eingetragen.

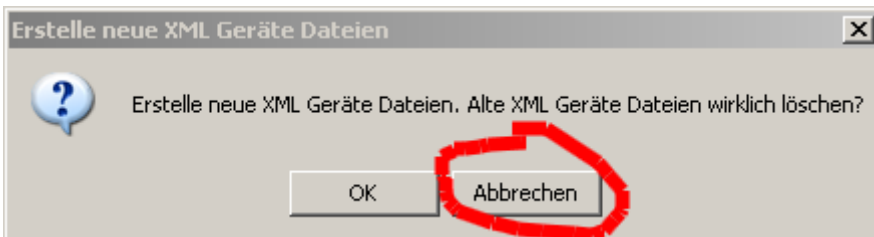
Der TwinCAT LON-Funktionsbaustein wertet den Index-Puffer aus, übergibt empfangene Eingangsvariablen an das Applikationsprogramm und trägt diese aus dem Index-Puffer aus. Wenn der Funktionsbaustein die Daten nicht schnell genug aus dem Telegramm-Puffer liest, kann dieser bis zu 62 Einträge (also die maximale Anzahl der SNVT-Variablen) speichern.

Wird von der KL6401 ein bereits eingegangenes Telegramm (das der Funktionsbaustein noch nicht an das Applikationsprogramm übertragen hat, das also noch im Index-Puffer eingetragen ist) erneut empfangen, dann wird kein neuer Eintrag im Index-Puffer vorgenommen, die Eingangsvariablen im Telegramm-Puffer werden aber aktualisiert.

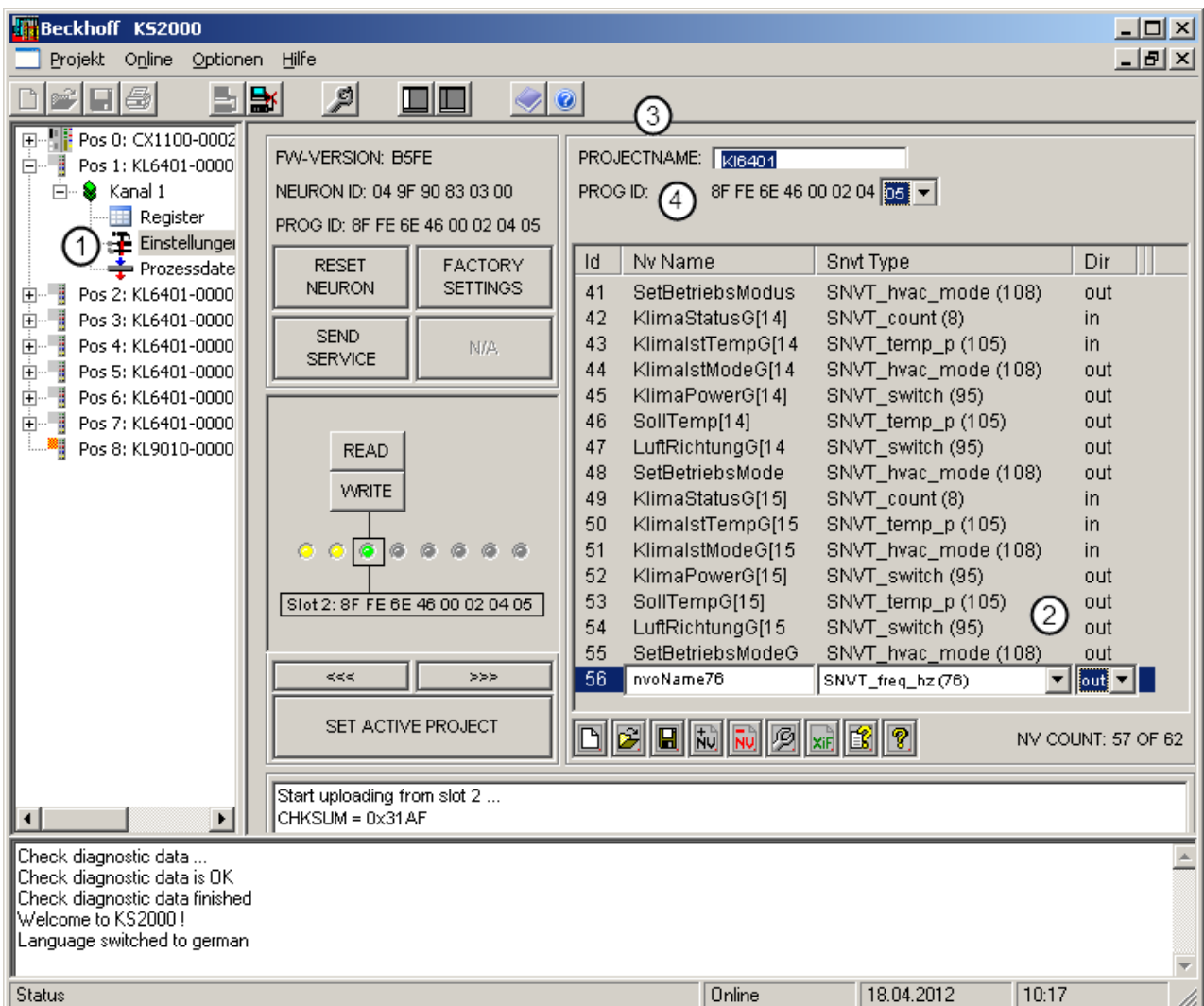
# 5 Konfiguration mit KS2000




## Schritte zur Inbetriebnahme

1. Installieren Sie Konfigurations-Software KS2000, Version 4.3.0.39 oder höher.
2. Stecken Sie an ihrem Buskoppler die KL6401 und schließen Sie den K-Bus mit einer KL9010 ab.
3. Loggen Sie sich ein. Sollte sich das Dialogfenster "Erstelle neue XML Geräte Dateien" öffnen, brechen Sie an dieser Stelle ab.



In den "Einstellungen" 1 kann nun die Klemme konfiguriert werden.



In der Tabelle  werden die SNVTs eingetragen. Mit  kann eine Variable hinzugefügt und mit  eine markierte Variable entfernt werden.


Die Spalte "Id" kann nicht editiert werden. Hier steht der NV-Index drin. Dieser Index wird auch als Input Variable "wNVIndex" der SPS-Bausteine [\[► 59\]](#) benötigt.

In der Spalte "Nv Name" kann ein beliebiger Text eingetragen werden. Diese Bezeichnungen werden im LON-Konfigurationstool (wird nicht von Beckhoff geliefert) zur Identifizierung der jeweiligen Variablen benötigt.

In der Spalte "Snvt Type" wird der gewünschte SNVT ausgewählt.

In der Spalte "Dir" (Direction) wird angegeben, ob die Variable gesendet (out) oder empfangen (in) werden soll.

 Stellen Sie einen beliebigen Projektnamen ein (Klick auf "PROJECTNAME").


Anschließend editieren Sie die Programm-ID (Prog ID) . Klicken Sie hierzu auf Programm ID und wählen Sie eine Programm-ID aus. Diese Programm-ID dürfen Sie in Ihrem LON-Projekt nur einmal vergeben. Sie können maximal 256 (0-255) Programm-IDs verwenden. Das heißt, dass Sie in einem Projekt 256 verschiedene Konfigurationen verwenden können.



LON-Klemmen mit gleicher Konfiguration besitzen auch die gleiche Programm-ID.

---

Mit der Schaltfläche  kann nun das Xif File erstellt werden. Diese Datei wird im LON-Konfigurationstool (wird nicht von Beckhoff geliefert) benötigt.

Mit der Schaltfläche  kann die Konfiguration in ein BLC-File abgespeichert werden. Mit dieser Datei kann bei einem Klemmentausch, oder wenn die gleiche Konfiguration in einer anderen Klemme verwendet werden soll, die Konfiguration geladen werden.

Die Daten können jetzt in die Klemme geschrieben werden. Dazu mit den Schaltflächen [>>>] oder [<<<] den gewünschten Slot auswählen und mit der Schaltfläche "WRITE" die Daten in die Klemme schreiben. Der Slot darf dabei nicht aktiviert (grün) sein.



Ausgeliefert (Herstellereinstellung) wird die Klemme mit aktivem Slot 0. Dieser Slot enthält eine feste Konfiguration und kann nicht verändert werden. Sollen andere Werte übertragen werden, muss ein anderer Slot konfiguriert und aktiviert werden. Es ist immer nur ein Slot aktiv.

---

Nach erfolgreichem Download muss das Projekt aktiviert werden. Dazu die Schaltfläche "SET ACTIVE PROJECT" betätigen. Die KL6401 muss jetzt spannungslos geschaltet werden. Nach erneutem Einschalten der KL6401 (Spannung wieder einschalten) ist die gewünschte Konfiguration aktiviert.

### Beschreibung der Editierfunktionen





Löscht die Tabelle



Öffnet ein bestehendes Konfigurationsfile (\*.BLC) der SNVTs für die KL6401



Speichert die aktuelle Konfiguration als BLC-File ab



Editiert eine SNVT-Variable



Fügt eine SNVT-Variable an



Löscht eine SNVT-Variable



Erstellt ein XIF-File für ein LON-Konfigurationstool (wird nicht von Beckhoff geliefert)



Öffnet die Hilfe



Info

## 6 Integration in TwinCAT

### 6.1 KL6401 - Verknüpfung mit dem TwinCAT System Manager

#### Wie verknüpfe ich die KL6401 mit dem System Manager?

Die Variable *wParameter* der Eingangsstruktur mit der rechten Maustaste anklicken und "Verknüpfung ändern" auswählen. (Siehe Bild 1)

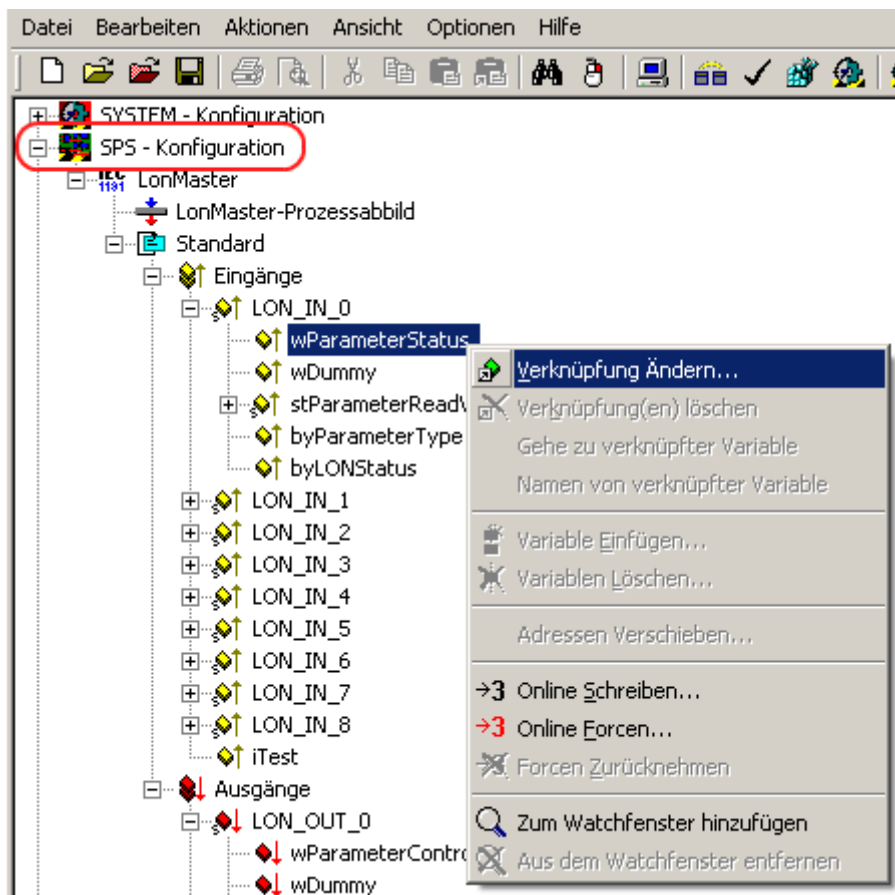


Bild 1

In der "E/A Konfiguration" die gewünschte Klemme auswählen, "Parameter Status" anklicken und "OK" betätigen. (Siehe Bild 2)

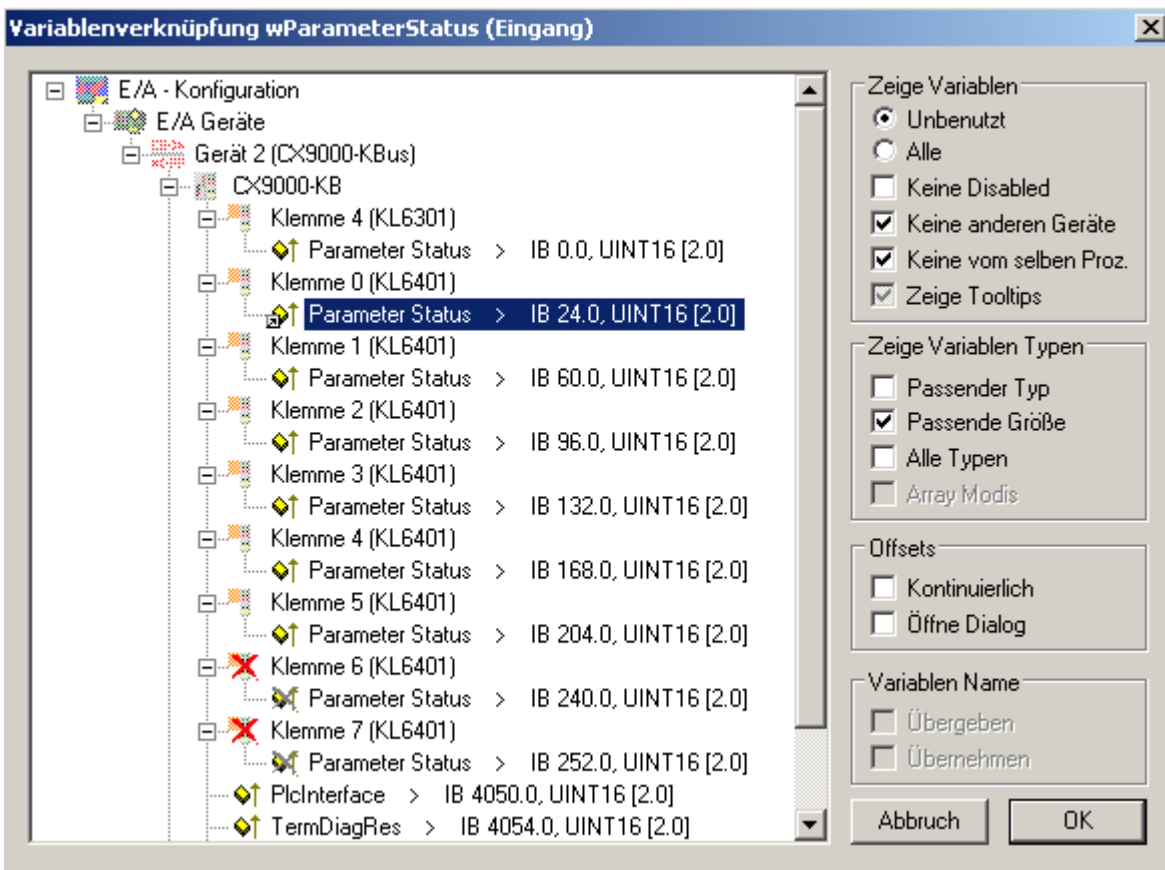


Abb. 1: TcLON\_VerknuepfungenSMBild2

Bild 2

Im nächsten Schritt die Struktur *stParameterReadValue* der Eingangsstruktur mit der rechten Maustaste anklicken und "Verknüpfung ändern" auswählen. (Siehe Bild 3) *wDummy* wird nicht verknüpft.

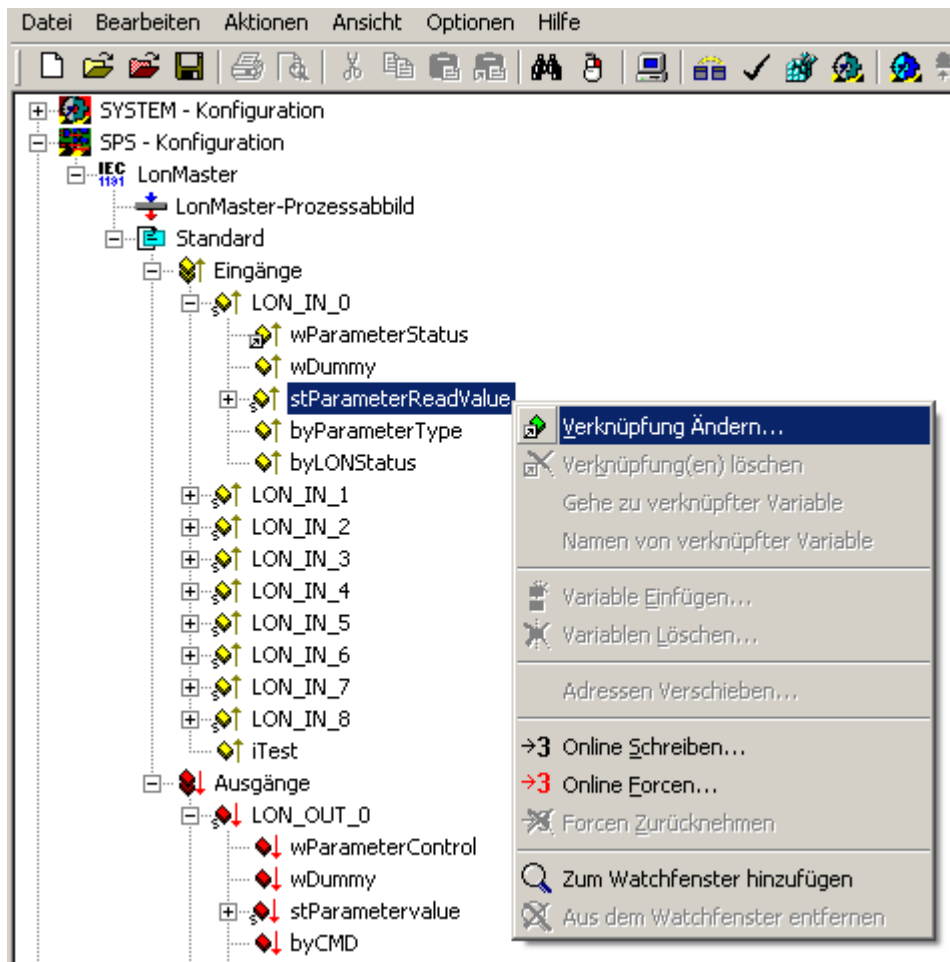


Abb. 2: TcLON\_VerknuepfungenSMBild3

*Bild 3*

Wählen Sie "Alle Typen" und "Kontinuierlich" an, und markieren Sie "Parameter Input Data 1" .bis "Parameter Input Data 8" mit der linken Maustaste + linker >SCHIFT< Taste (siehe Bild 4). Anschließend "OK" drücken.



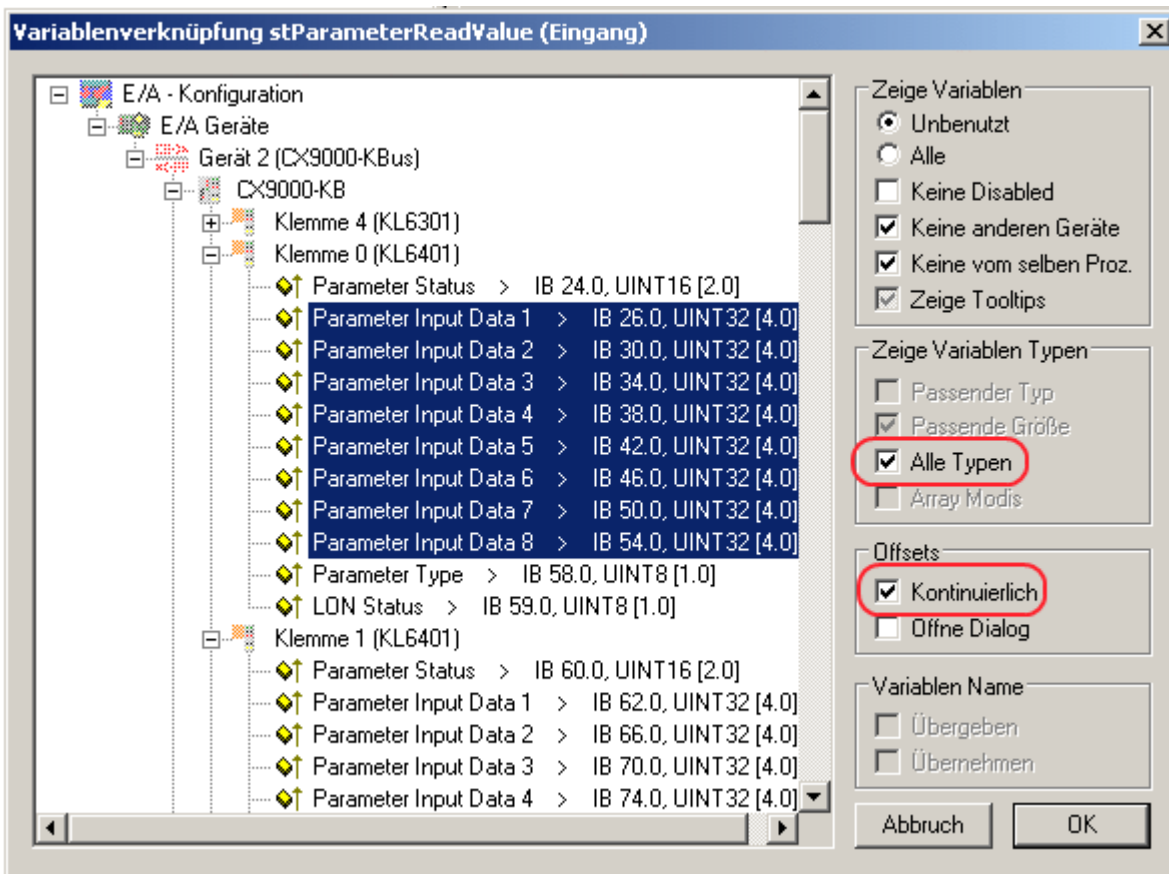


Abb. 3: TcLON\_VerknuepfungenSMBild4

Bild 4

Verknüpfen Sie nun auch die Variablen *byParameterType* und *byLONStatus*. (Siehe Bild 5 und 6)

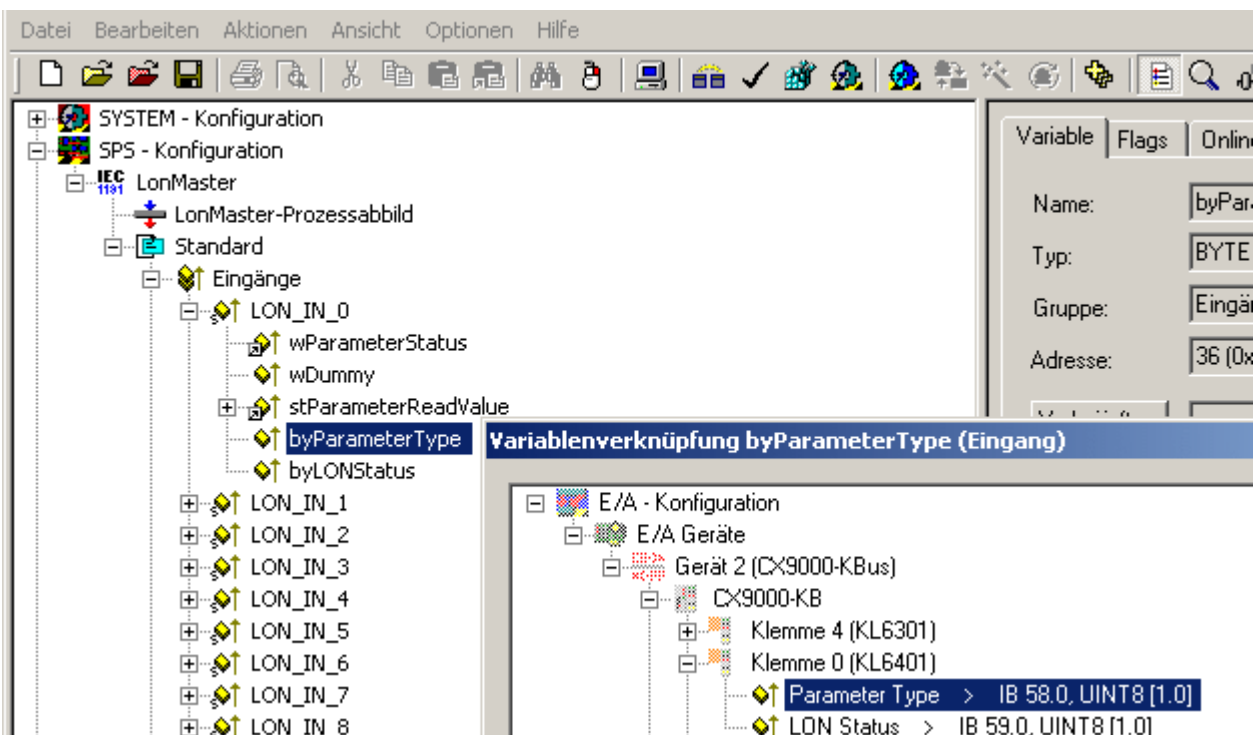


Abb. 4: TcLON\_VerknuepfungenSMBild5

Bild 5

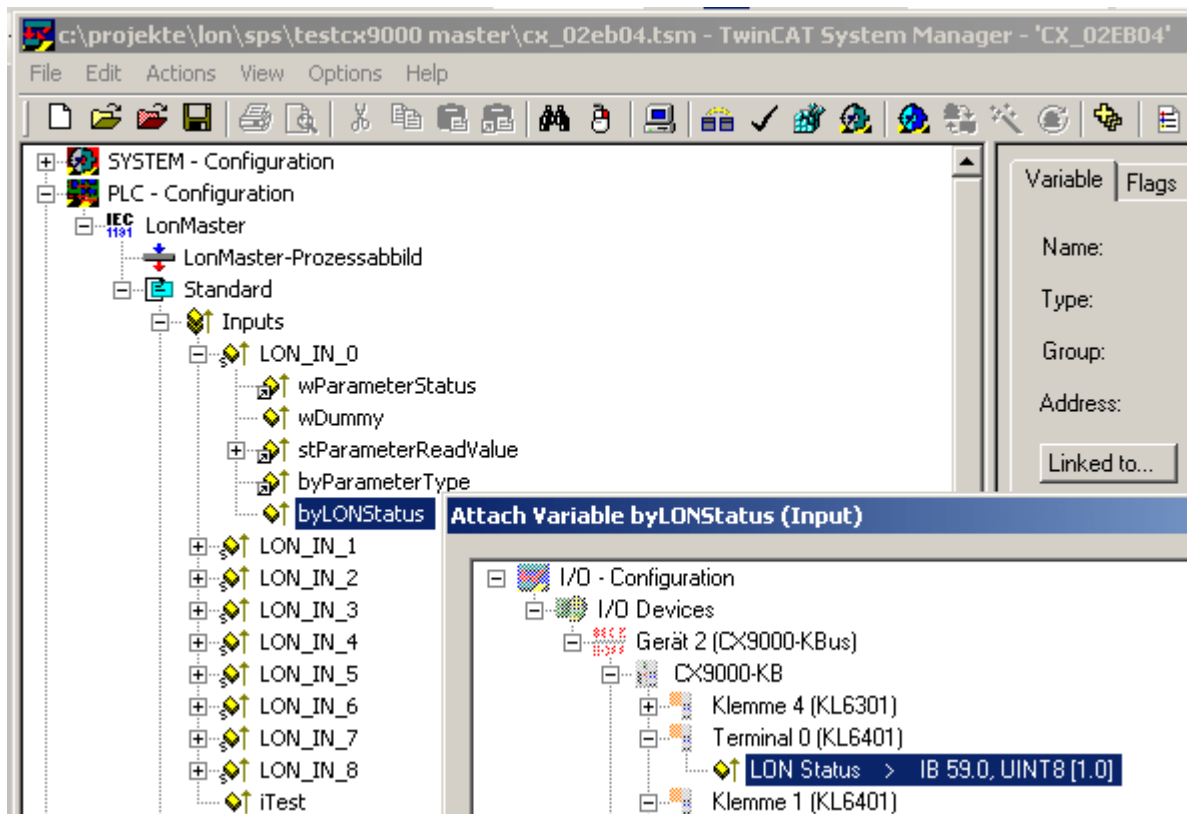


Abb. 5: TcLON\_VerknuepfungenSMBild6

Bild 6

Sie können die Verknüpfung jetzt kontrollieren. Gehen Sie dazu in der "E/A Konfiguration / E/AGeräte" auf die KL6401 und öffnen Sie diese. Alle Daten der Klemme müssen jetzt mit einem kleinen Pfeil markiert sein (siehe Bild 7). Ist dies der Fall fahren Sie genauso mit den Ausgängen fort.

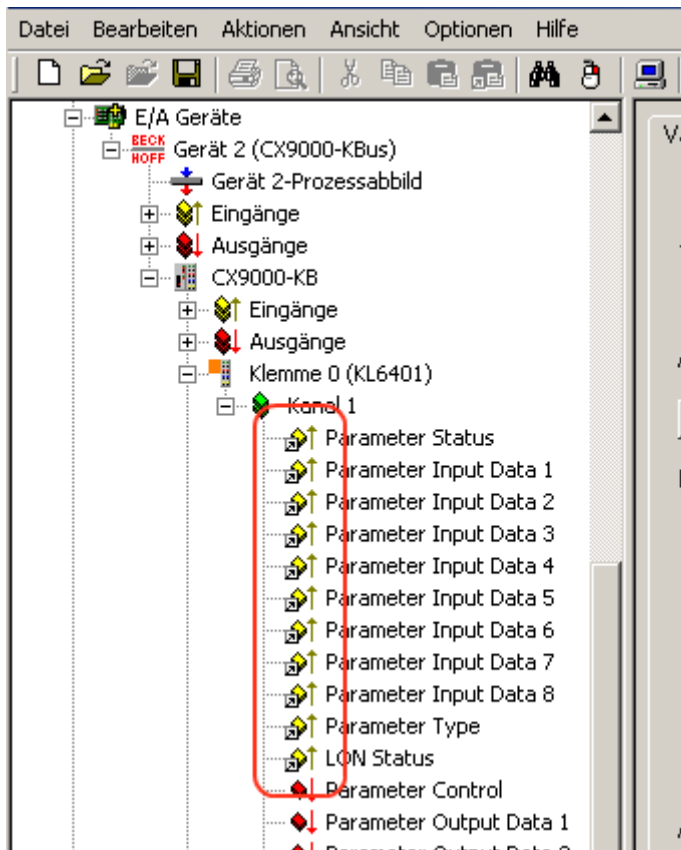


Bild 7

## 6.2 Integration in TwinCAT (CX9020)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für LON in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll der Zustand eines Schaltausgangs per Taster verändert werden.

<https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997701515.zip> <https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997701515.zip>

### Hardware

#### Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Embedded-PC [CX9020](#)
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Setz und Reset Funktion)
- 1x LON-Klemme [KL6401](#)
- 1x Endklemme KL9010

Richten Sie die Hardware sowie die LON-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass ein Setz-Taster auf den ersten und ein Reset-Taster auf den zweiten Eingang der KL1002 gelegt wurde.

## Software

### Erstellung des SPS-Programms

Erstellen Sie ein neues SPS-Projekt für PC-basierte Systeme (ARM) und fügen die Bibliothek *TcLON.lib* hinzu.

Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
  bSet          AT %I*      : BOOL;
  bReset        AT %I*      : BOOL;
  stParameter_IN AT %I*     : ST_LON_Parameter_IN_36B;
  stParameter_OUT AT %Q*    : ST_LON_Parameter_OUT_36B;
  stLON_Com     : ST_LON_Communication;
END_VAR
```

**bSwitch** : Eingangsvariable für den Setz-Taster.

**bReset** : Eingangsvariable für den Reset-Taster.

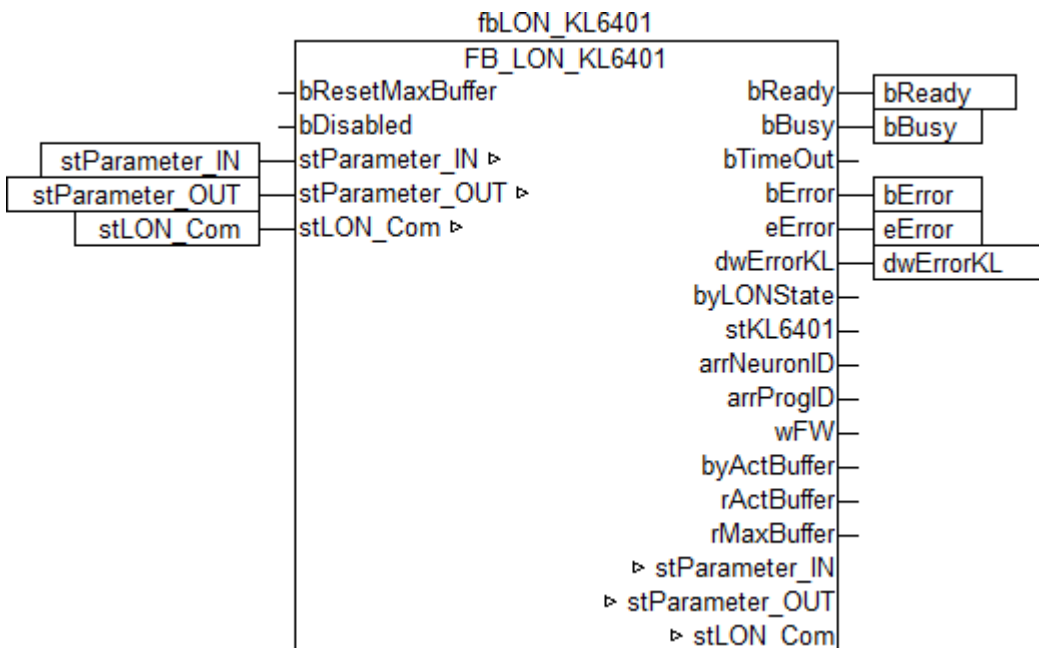
**stParameter\_IN** : Eingangsvariable [▶ 560] für die LON-Klemme.

**stParameter\_OUT** : Ausgangsvariable [▶ 561] für die LON-Klemme.

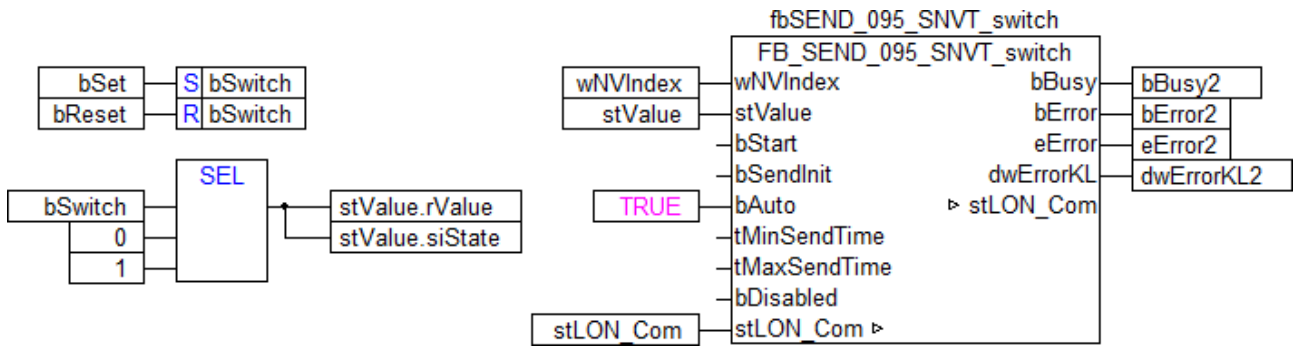
**stLON\_Com** : Wird für die Kommunikation [▶ 562] mit LON benötigt.

Alle Bausteine bei LON müssen in derselben Task aufgerufen werden.

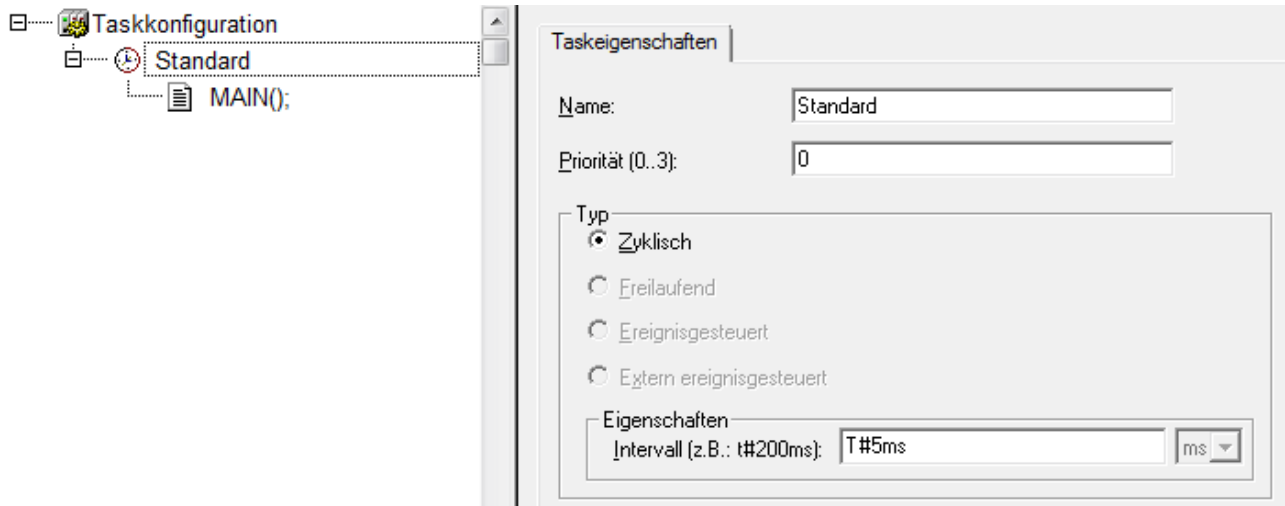
Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine FB\_LON\_KL6401() [▶ 68] und FB\_SEND\_095\_SNVT\_switch() [▶ 334] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *stParameter\_IN*, *stParameter\_OUT* und *stLON\_Com* zu verknüpfen.



Die lokale Variable *bSwitch* wird mit den globalen Variablen *bSet* und *bReset* verknüpft und anschließend mit dem Eingang der Selektion. Verknüpfen Sie die lokale Variable *stValue* erst mit dem Ausgang der Selektion und dann mit dem Eingang *stValue* des Sendebausteins.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins `FB_LON_KL6401()` [► 68].

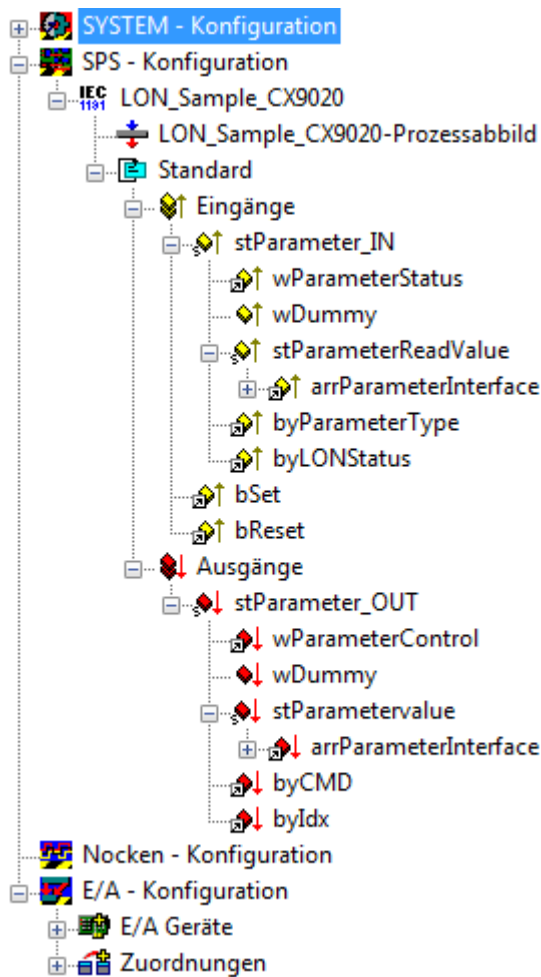


Laden Sie das Projekt als Bootprojekt auf den CX und speichern Sie es ab.

### Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den CX und lassen Sie nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.

Ihr CX ist jetzt einsatzbereit.

Durch Betätigen der Taster kann der Schaltausgang gesetzt bzw. zurückgesetzt werden.

## 6.3 Integration in TwinCAT (BC9191)

Dieses Beispiel beschreibt, wie ein einfaches SPS-Programm für LON in TwinCAT geschrieben werden kann und wie es mit der Hardware verknüpft wird. Es soll der Zustand eines Schaltausgangs per Taster verändert werden.

<https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997702923.zip> <https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997702923.zip>

### Hardware

#### Einrichtung der Komponenten

Es wird folgende Hardware benötigt:

- 1x Busklemmen Controller [BC9191](#)
- 1x Potenzialeinspeiseklemme 24V DC
- 1x Digitale 2-Kanal-Eingangsklemme KL1002 (für die Setz und Reset Funktion)
- 1x LON-Klemme [KL6401](#)
- 1x Endklemme KL9010

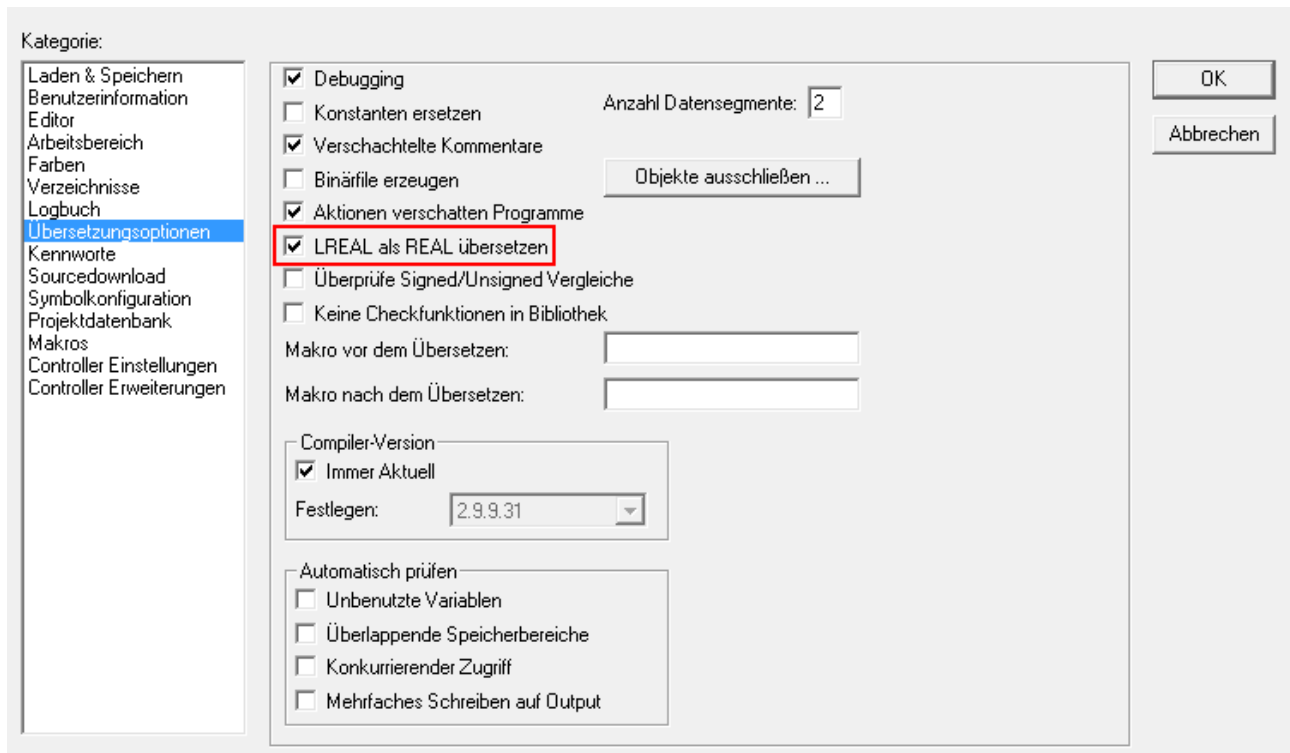
Richten Sie die Hardware sowie die LON-Komponenten wie in den entsprechenden Dokumentationen beschrieben ein.

Dieses Beispiel geht davon aus, dass ein Setz-Taster auf den ersten und ein Reset-Taster auf den zweiten Eingang der KL1002 gelegt wurde.

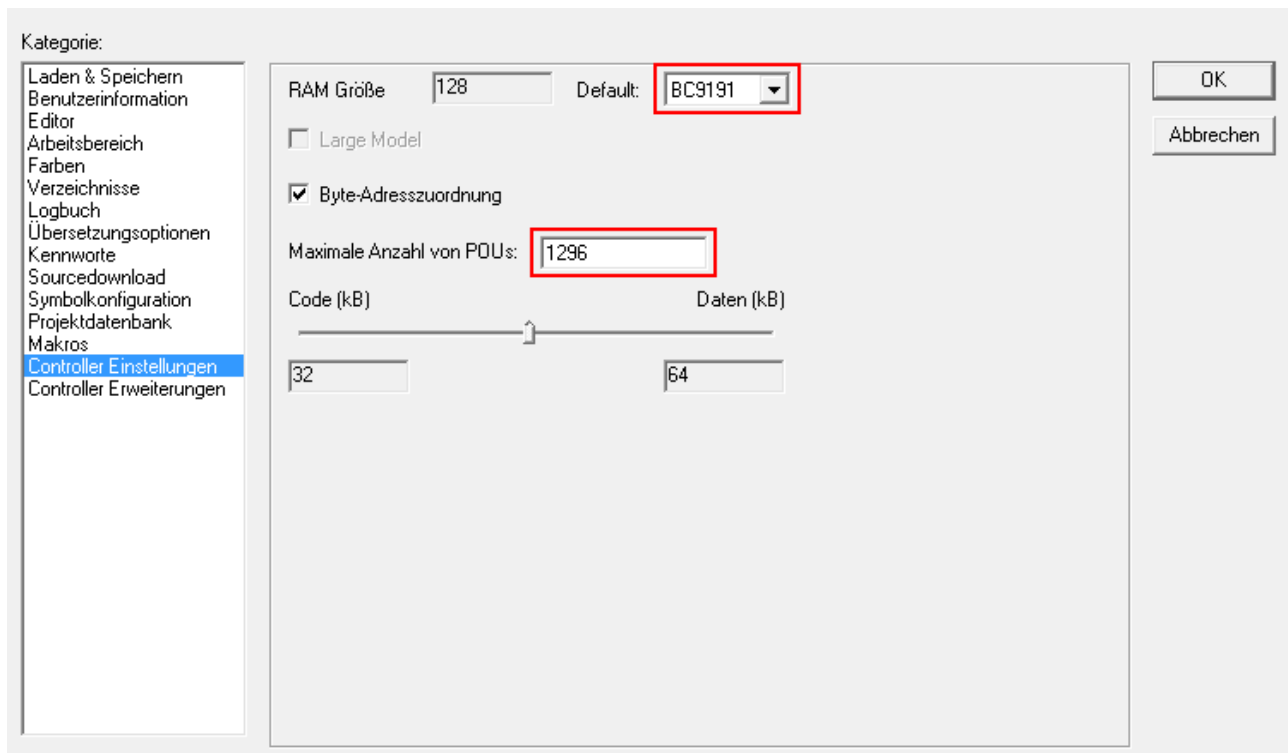
**Software**

**Erstellung des SPS-Programms**

Erstellen Sie ein neues SPS-Projekt für BC-basierte Systeme (BCxx50 über AMS) und fügen die Bibliotheken *TcLON.lbx* und *TcSystemBCxx50.lbx* hinzu. Gehen Sie danach im Menü auf *Projekt* → *Optionen...* → *Übersetzungsoptionen* und wählen *LREAL als REAL übersetzen* an.



Ändern Sie anschließend noch unter *Controller Einstellungen* den Controller auf BC9191 und erhöhen Sie die maximale Anzahl von POU's auf 1296.



Erzeugen Sie als Nächstes die folgenden globalen Variablen:

```
VAR_GLOBAL
  bSet      AT %I*      : BOOL;
  bReset    AT %I*      : BOOL;
  stParameter_IN  AT %I* : ST_LON_Parameter_IN_36B;
  stParameter_OUT AT %Q* : ST_LON_Parameter_OUT_36B;
  stLON_Com : ST_LON_Communication;
END_VAR
```

**bSwitch** : Eingangsvariable für den Setz-Taster.

**bReset** : Eingangsvariable für den Reset-Taster.

**stParameter\_IN** : Eingangsvariable [► 560] für die LON-Klemme.

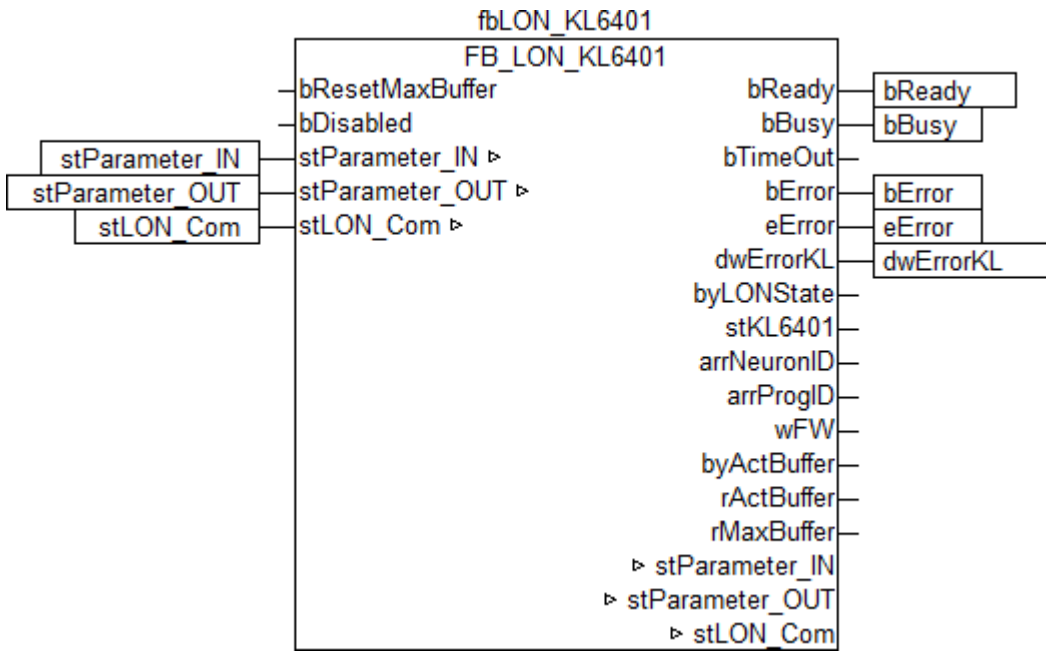
**stParameter\_OUT** : Ausgangsvariable [► 561] für die LON-Klemme.

**stLON\_Com** : Wird für die Kommunikation [► 562] mit LON benötigt.

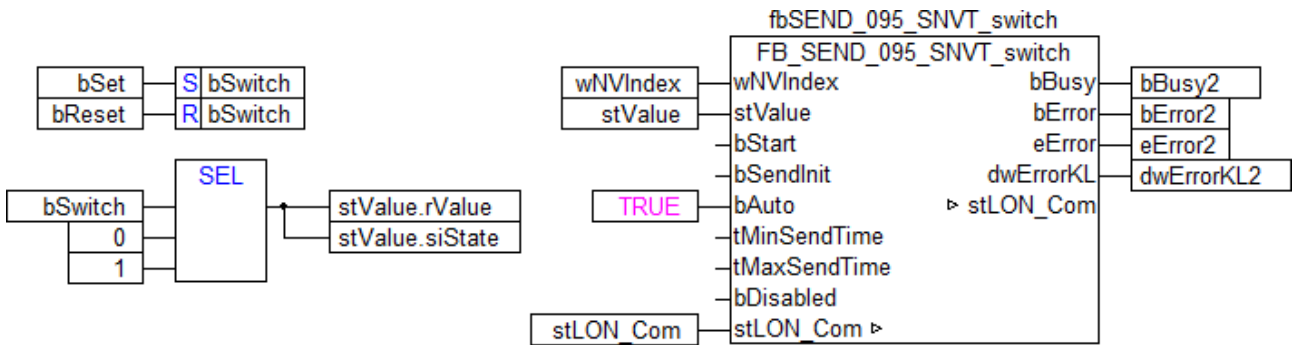
Da BC-Geräte nur über eine Task verfügen, kann die Kommunikation mit LON nicht separat ausgeführt werden.

Legen Sie daher ein MAIN-Programm (CFC) an in dem die Bausteine FB LON\_KL6401() [► 68] und FB SEND\_095\_SNVT\_switch() [► 334] aufgerufen werden. Achten Sie beim Kommunikationsbaustein darauf, mit *stParameter\_IN*, *stParameter\_OUT* und *stLON\_Com* zu verknüpfen.

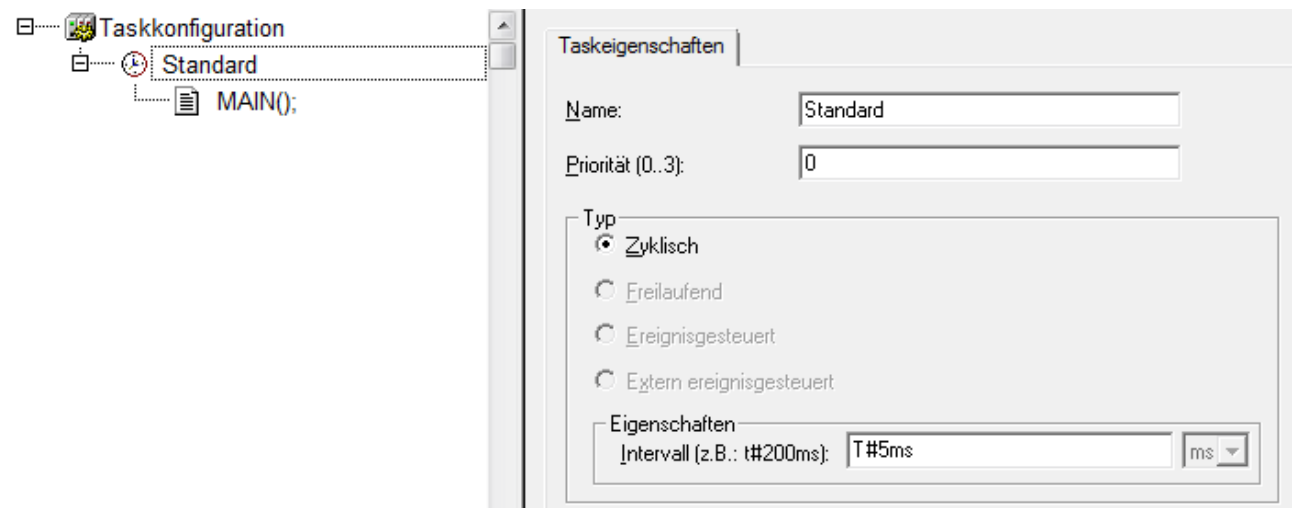




Die lokale Variable *bSwitch* wird mit den globalen Variablen *bSet* und *bReset* verknüpft und anschließend mit dem Eingang der Selektion. Verknüpfen Sie die lokale Variable *stValue* erst mit dem Ausgang der Selektion und dann mit dem Eingang *stValue* des Sendebausteins.



Gehen Sie in die Taskkonfiguration und geben Sie der Task eine niedrigere Intervall-Zeit. Genauere Informationen dazu finden Sie in der Beschreibung des Bausteins `FB_LON_KL6401()` [P. 68].

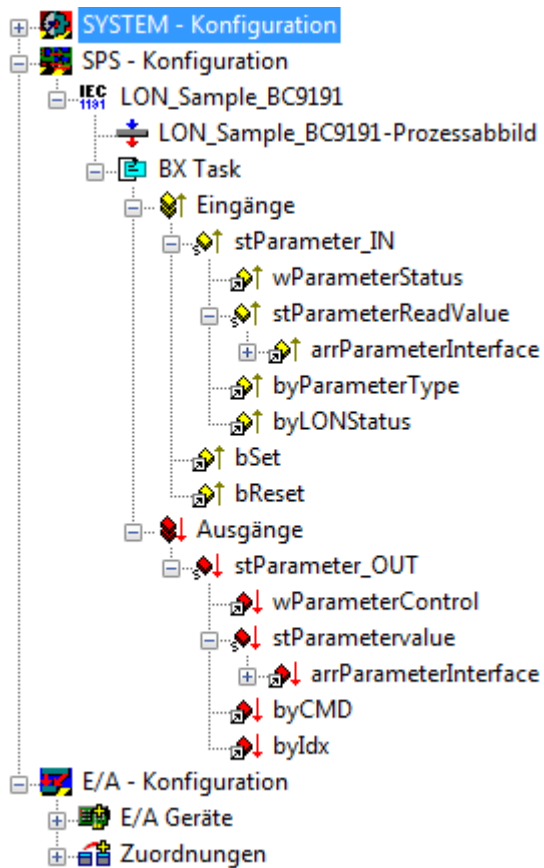


Laden Sie das Projekt als Bootprojekt auf den BC und speichern Sie es ab.

### Konfiguration im System Manager

Legen Sie ein neues TwinCAT System-Manager-Projekt an, wählen Sie als Zielsystem den BC und lassen Sie nach dessen Hardware suchen.

Fügen Sie das oben angelegte SPS-Programm unter SPS-Konfiguration hinzu.



Verknüpfen Sie die globalen Variablen des SPS-Programms nun mit den Ein- und Ausgängen der Busklemmen, erzeugen Sie die Zuordnungen und aktivieren Sie die Konfiguration. Starten Sie dann das Gerät im Run-Modus.

Ihr BC ist jetzt einsatzbereit.

Durch Betätigen der Taster kann der Schaltausgang gesetzt bzw. zurückgesetzt werden.

# 7 Programmierung

| Inhalt   |
|--|
| Allgemeine Informationen [ <a href="#">▶ 58</a> ]                    |
| KL6401 - Verknüpfung mit dem System Manager [ <a href="#">▶ 30</a> ] |

## Bausteine

| Bausteine                              | Beschreibung              |
|--|---------------------------|
| FB_LON_KL6401 [ <a href="#">▶ 68</a> ] | Sende- / Empfangsbaustein |

## Bausteine/Read

| Bausteine   | Beschreibung                                    |
|---|---|
| FB_READ_001_SNVT_amp [ <a href="#">▶ 70</a> ]         | Elektrischer Strom (Ampere)                     |
| FB_READ_002_SNVT_amp_mil [ <a href="#">▶ 70</a> ]     | Elektrischer Strom (Milliampere)                |
| FB_READ_003_SNVT_angle [ <a href="#">▶ 71</a> ]       | Winkelabstand (Bogenmaß)                        |
| FB_READ_004_SNVT_angle_vel [ <a href="#">▶ 72</a> ]   | Winkelgeschwindigkeit (Bogenmaß / Sekunde)      |
| FB_READ_005_SNVT_btu_kilo [ <a href="#">▶ 73</a> ]    | Thermische Energie (Kilo-Btus)                  |
| FB_READ_006_SNVT_btu_mega [ <a href="#">▶ 73</a> ]    | Thermische Energie (Mega-Btus)                  |
| FB_READ_007_SNVT_char_ascii [ <a href="#">▶ 74</a> ]  | ASCII Zeichen (8-bit ASCII character)           |
| FB_READ_008_SNVT_count [ <a href="#">▶ 75</a> ]       | Absolute Anzahl (Stück)                         |
| FB_READ_009_SNVT_count_inc [ <a href="#">▶ 76</a> ]   | Inkrement Zähler (Stück(delta))                 |
| FB_READ_011_SNVT_date_day [ <a href="#">▶ 76</a> ]    | Wochentag                                       |
| FB_READ_013_SNVT_elec_kwh [ <a href="#">▶ 77</a> ]    | Elektrische Energie (kW/h)                      |
| FB_READ_014_SNVT_elec_whr [ <a href="#">▶ 78</a> ]    | Elektrische Energie (W/h)                       |
| FB_READ_015_SNVT_flow [ <a href="#">▶ 79</a> ]        | Volumenstrom (Liter / Sekunde)                  |
| FB_READ_016_SNVT_flow_mil [ <a href="#">▶ 80</a> ]    | Volumenstrom (Milliliter / Sekunde)             |
| FB_READ_017_SNVT_length [ <a href="#">▶ 80</a> ]      | Länge (Meter)                                   |
| FB_READ_018_SNVT_length_kilo [ <a href="#">▶ 81</a> ] | Länge (Kilometer)                               |
| FB_READ_019_SNVT_length_micr [ <a href="#">▶ 82</a> ] | Länge (Mikrometer)                              |
| FB_READ_020_SNVT_length_mil [ <a href="#">▶ 83</a> ]  | Länge (Millimeter)                              |
| FB_READ_021_SNVT_lev_cont [ <a href="#">▶ 83</a> ]    | Stufenlose Wertänderung (% vom obersten Niveau) |
| FB_READ_023_SNVT_mass [ <a href="#">▶ 84</a> ]        | Masse (Gramm)                                   |
| FB_READ_024_SNVT_mass_kilo [ <a href="#">▶ 85</a> ]   | Masse (Kilogramm)                               |
| FB_READ_025_SNVT_mass_mega [ <a href="#">▶ 86</a> ]   | Masse (Metrische Tonne)                         |
| FB_READ_026_SNVT_mass_mil [ <a href="#">▶ 86</a> ]    | Masse (Milligramm)                              |
| FB_READ_027_SNVT_power [ <a href="#">▶ 87</a> ]       | Leistung (Watt)                                 |
| FB_READ_028_SNVT_power_kilo [ <a href="#">▶ 88</a> ]  | Leistung (Kilowatt)                             |
| FB_READ_029_SNVT_ppm [ <a href="#">▶ 89</a> ]         | Konzentration (ppm)                             |
| FB_READ_030_SNVT_press [ <a href="#">▶ 89</a> ]       | Druck (Überdruck) (Kilopascal)                  |
| FB_READ_031_SNVT_res [ <a href="#">▶ 90</a> ]         | Elektrischer Widerstand (Ohm)                   |
| FB_READ_032_SNVT_res_kilo [ <a href="#">▶ 91</a> ]    | Elektrischer Widerstand (Kilohm)                |
| FB_READ_033_SNVT_sound_db [ <a href="#">▶ 92</a> ]    | Schalldruckpegel (dB)                           |
| FB_READ_034_SNVT_speed [ <a href="#">▶ 92</a> ]       | Lineare Geschwindigkeit (Meter / Sekunde)       |
| FB_READ_035_SNVT_speed_mil [ <a href="#">▶ 93</a> ]   | Lineare Geschwindigkeit (Millimeter / Sekunde)  |

| Bausteine  | Beschreibung                                      |
|--|---|
| <a href="#">FB_READ_036_SNVT_str_asc [▶ 94]</a>      | Zeichenkette                                      |
| <a href="#">FB_READ_037_SNVT_str_int [▶ 95]</a>      | Wide Character String                             |
| <a href="#">FB_READ_038_SNVT_telcom [▶ 95]</a>       | Telefonstatus (Telefonstatus Namen)               |
| <a href="#">FB_READ_039_SNVT_temp [▶ 96]</a>         | Temperatur (Grad Celsius)                         |
| <a href="#">FB_READ_041_SNVT_vol [▶ 97]</a>          | Volumen (Liter)                                   |
| <a href="#">FB_READ_042_SNVT_vol_kilo [▶ 98]</a>     | Volumen (Kiloliter)                               |
| <a href="#">FB_READ_043_SNVT_vol_mil [▶ 99]</a>      | Volumen (Milliliter)                              |
| <a href="#">FB_READ_044_SNVT_volt [▶ 99]</a>         | Elektrische Spannung (Volt)                       |
| <a href="#">FB_READ_045_SNVT_volt_dbmv [▶ 100]</a>   | Elektrische Spannung (dB Mikrovolt)               |
| <a href="#">FB_READ_046_SNVT_volt_kilo [▶ 101]</a>   | Elektrische Spannung (Kilovolt)                   |
| <a href="#">FB_READ_047_SNVT_volt_mil [▶ 102]</a>    | Elektrische Spannung (Millivolt)                  |
| <a href="#">FB_READ_048_SNVT_amp_f [▶ 102]</a>       | Elektrischer Strom (Ampere)                       |
| <a href="#">FB_READ_049_SNVT_angle_f [▶ 103]</a>     | Winkelabstand (Bogenmaß)                          |
| <a href="#">FB_READ_050_SNVT_angle_vel_f [▶ 104]</a> | Winkelgeschwindigkeit (Radiant / Sekunde)         |
| <a href="#">FB_READ_051_SNVT_count_f [▶ 105]</a>     | Absolute Anzahl (Stück)                           |
| <a href="#">FB_READ_052_SNVT_count_inc_f [▶ 105]</a> | Inkrement Zähler (Stück(delta))                   |
| <a href="#">FB_READ_053_SNVT_flow_f [▶ 106]</a>      | Volumenstrom (Liter / Sekunde)                    |
| <a href="#">FB_READ_054_SNVT_length_f [▶ 107]</a>    | Länge (Meter)                                     |
| <a href="#">FB_READ_055_SNVT_lev_cont_f [▶ 108]</a>  | Stufenlose Wertänderung (% vom obersten Niveau)   |
| <a href="#">FB_READ_056_SNVT_mass_f [▶ 108]</a>      | Masse (Gramm)                                     |
| <a href="#">FB_READ_057_SNVT_power_f [▶ 109]</a>     | Leistung (Watt)                                   |
| <a href="#">FB_READ_058_SNVT_ppm_f [▶ 110]</a>       | Konzentration (ppm)                               |
| <a href="#">FB_READ_059_SNVT_press_f [▶ 111]</a>     | Druck (Überdruck) (Pascal)                        |
| <a href="#">FB_READ_060_SNVT_res_f [▶ 111]</a>       | Elektrischer Widerstand (Ohm)                     |
| <a href="#">FB_READ_061_SNVT_sound_db_f [▶ 112]</a>  | Schalldruckpegel (dBspl)                          |
| <a href="#">FB_READ_062_SNVT_speed_f [▶ 113]</a>     | Lineare Geschwindigkeit (Meter / Sekunde)         |
| <a href="#">FB_READ_063_SNVT_temp_f [▶ 114]</a>      | Temperatur (Grad Celsius)                         |
| <a href="#">FB_READ_064_SNVT_time_f [▶ 114]</a>      | Abgelaufene Zeit (Sekunden)                       |
| <a href="#">FB_READ_065_SNVT_vol_f [▶ 115]</a>       | Volumen (Liter)                                   |
| <a href="#">FB_READ_066_SNVT_volt_f [▶ 116]</a>      | Elektrische Spannung (Volt)                       |
| <a href="#">FB_READ_067_SNVT_btu_f [▶ 117]</a>       | Thermische Energie (Btus)                         |
| <a href="#">FB_READ_068_SNVT_elec_whr_f [▶ 117]</a>  | Elektrische Energie (Watt / Stunden)              |
| <a href="#">FB_READ_069_SNVT_config_src [▶ 118]</a>  | Konfigurationseigenschaften                       |
| <a href="#">FB_READ_070_SNVT_color [▶ 119]</a>       | Farbe nach CIE Norm                               |
| <a href="#">FB_READ_071_SNVT_grammage [▶ 120]</a>    | amerikanisches Mass für Papiergewichte und Dichte |
| <a href="#">FB_READ_072_SNVT_grammage_f [▶ 120]</a>  | amerikanisches Mass für Papiergewichte und Dichte |
| <a href="#">FB_READ_073_SNVT_file_req [▶ 121]</a>    | Dateianforderung                                  |
| <a href="#">FB_READ_074_SNVT_file_status [▶ 122]</a> | Dateiinformatio                                   |
| <a href="#">FB_READ_075_SNVT_freq_f [▶ 123]</a>      | Frequenz (Hertz)                                  |
| <a href="#">FB_READ_076_SNVT_freq_hz [▶ 123]</a>     | Frequenz (Hertz)                                  |
| <a href="#">FB_READ_077_SNVT_freq_kilohz [▶ 124]</a> | Frequenz (Kilohertz)                              |
| <a href="#">FB_READ_078_SNVT_freq_milhz [▶ 125]</a>  | Frequenz (Millihertz)                             |
| <a href="#">FB_READ_079_SNVT_lux [▶ 126]</a>         | Beleuchtungsstärke (Lux)                          |

| <b>Bausteine</b>                     | <b>Beschreibung</b>                              |
|--------------------------------------|--|
| FB_READ_081_SNVT_lev_percent [▶_126] | Prozentangabe                                    |
| FB_READ_082_SNVT_multiplier [▶_127]  | Multiplikator                                    |
| FB_READ_083_SNVT_state [▶_128]       | Statusinformation                                |
| FB_READ_084_SNVT_time_stamp [▶_129]  | Zeitstempel                                      |
| FB_READ_085_SNVT_zerospan [▶_130]    | Nullpunkt und Proportionalitätsfaktor            |
| FB_READ_086_SNVT_magcard [▶_130]     | ISO 7811 (40 Hexadezimal Zahlen)                 |
| FB_READ_087_SNVT_elapsed_tm [▶_131]  | Abgelaufene Zeit                                 |
| FB_READ_088_SNVT_alarm [▶_132]       | Alarm Status                                     |
| FB_READ_089_SNVT_currency [▶_133]    | Währung  |
| FB_READ_090_SNVT_file_pos [▶_134]    | Dateiposition                                    |
| FB_READ_091_SNVT_muldiv [▶_134]      | Verstärkungsfaktor/Dämpfungsfaktor               |
| FB_READ_092_SNVT_obj_request [▶_135] | Funktionsauswahl                                 |
| FB_READ_093_SNVT_obj_status [▶_136]  | Objektstatus                                     |
| FB_READ_094_SNVT_preset [▶_137]      | Voreinstellung                                   |
| FB_READ_095_SNVT_switch [▶_138]      | Schalter   |
| FB_READ_096_SNVT_trans_table [▶_138] | Übersetzungstabelle                              |
| FB_READ_097_SNVT_override [▶_139]    | Übersteuerungsmodus                              |
| FB_READ_098_SNVT_pwr_fact [▶_140]    | Leistungsfaktor                                  |
| FB_READ_099_SNVT_pwr_fact_f [▶_141]  | Leistungsfaktor                                  |
| FB_READ_100_SNVT_density [▶_142]     | Dichte (kg/m <sup>3</sup> )                      |
| FB_READ_101_SNVT_density_f [▶_142]   | Dichte (kg/m <sup>3</sup> )                      |
| FB_READ_102_SNVT_rpm [▶_143]         | Winkelgeschwindigkeit (Umdrehungen/Minute (RPM)) |
| FB_READ_103_SNVT_hvac_emerg [▶_144]  | HLK Notbetrieb                                   |
| FB_READ_104_SNVT_angle_deg [▶_145]   | Winkelangabe                                     |
| FB_READ_105_SNVT_temp_p [▶_145]      | Temperatur (Grad Celsius)                        |
| FB_READ_106_SNVT_temp_setpt [▶_146]  | Temperatur (6 Temperaturwerte)                   |
| FB_READ_107_SNVT_time_sec [▶_147]    | Abgelaufene Zeit (Sekunde)                       |
| FB_READ_108_SNVT_hvac_mode [▶_148]   | HLK Betriebsmodus                                |
| FB_READ_109_SNVT_occupancy [▶_148]   | Präsenzmeldung (Stati)                           |
| FB_READ_110_SNVT_area [▶_149]        | Fläche (Quadratmeter)                            |
| FB_READ_111_SNVT_hvac_overid [▶_150] | HLK Übersteuerungsmodus                          |
| FB_READ_112_SNVT_hvac_status [▶_151] | HLK Status                                       |
| FB_READ_113_SNVT_press_p [▶_152]     | Druck (Überdruck) (Pascal)                       |
| FB_READ_114_SNVT_address [▶_152]     | Neuron Adresse                                   |
| FB_READ_115_SNVT_scene [▶_153]       | Szenen   |
| FB_READ_116_SNVT_scene_cfg [▶_154]   | Szenen Einstellung                               |
| FB_READ_117_SNVT_setting [▶_155]     | Einstellungen                                    |
| FB_READ_118_SNVT_evap_state [▶_155]  | Verdampferstatus                                 |
| FB_READ_119_SNVT_therm_mode [▶_156]  | Thermostatmodus                                  |
| FB_READ_120_SNVT_defr_mode [▶_157]   | Abtaumodus                                       |
| FB_READ_121_SNVT_defr_term [▶_158]   | Abschluss des Abtauvorgangs                      |
| FB_READ_122_SNVT_defr_state [▶_159]  | Abtaustatus (Enumeration)                        |

| <b>Bausteine</b>                     | <b>Beschreibung</b>                            |
|--------------------------------------|--|
| FB_READ_123_SNVT_time_min [▶ 159]    | Abgelaufen Zeit (Minuten)                      |
| FB_READ_124_SNVT_time_hour [▶ 160]   | Abgelaufene Zeit (Stunde)                      |
| FB_READ_125_SNVT_ph [▶ 161]          | Säuregehalt (pH)                               |
| FB_READ_126_SNVT_ph_f [▶ 162]        | Säuregehalt (pH)                               |
| FB_READ_127_SNVT_chlr_status [▶ 162] | Kälteerzeugerstatus                            |
| FB_READ_128_SNVT_tod_event [▶ 163]   | Präsenzzeit                                    |
| FB_READ_129_SNVT_smo_obscur [▶ 164]  | Verdunkelung durch Rauch                       |
| FB_READ_130_SNVT_fire_test [▶ 165]   | Anforderung Test Brandmeldeanlage              |
| FB_READ_131_SNVT_temp_ror [▶ 166]    | Wert der Temperaturänderung/-erhöhung          |
| FB_READ_132_SNVT_fire_init [▶ 166]   | Brandmeldertyp                                 |
| FB_READ_133_SNVT_fire_indcte [▶ 167] | Brandmeldeanzeige                              |
| FB_READ_134_SNVT_time_zone [▶ 168]   | Zeitzonebeschreibung                           |
| FB_READ_135_SNVT_earth_pos [▶ 169]   | Position auf der Erde                          |
| FB_READ_136_SNVT_reg_val [▶ 170]     | Registerwert                                   |
| FB_READ_137_SNVT_reg_val_ts [▶ 170]  | Registerwert                                   |
| FB_READ_138_SNVT_volt_ac [▶ 171]     | Wechselspannung (Volt AC)                      |
| FB_READ_139_SNVT_amp_ac [▶ 172]      | Wechselstrom                                   |
| FB_READ_143_SNVT_turbidity [▶ 173]   | Trübung  |
| FB_READ_144_SNVT_turbidity_f [▶ 174] | Trübung  |
| FB_READ_145_SNVT_hvac_type [▶ 174]   | HLK Anlagentyp                                 |
| FB_READ_146_SNVT_elec_kwh_l [▶ 175]  | Elektrische Energie (Kilowatt / Stunde)        |
| FB_READ_147_SNVT_temp_diff_p [▶ 176] | Temperaturdifferenz (Grad Celsius)             |
| FB_READ_148_SNVT_ctrl_req [▶ 177]    | Steuerungsanfrage                              |
| FB_READ_149_SNVT_ctrl_resp [▶ 178]   | Steuerungsantwort                              |
| FB_READ_150_SNVT_ptz [▶ 178]         | Kamera PTZ (SNZ)                               |
| FB_READ_151_SNVT_privacyzone [▶ 179] | Privatzone                                     |
| FB_READ_152_SNVT_pos_ctrl [▶ 180]    | Positionseinstellung für Kameras in Netzwerken |
| FB_READ_153_SNVT_enthalpy [▶ 181]    | Enthalpie (kiloJoules/kg)                      |
| FB_READ_154_SNVT_gfci_status [▶ 182] | Fehlerstromschutzschalter Status               |
| FB_READ_155_SNVT_motor_state [▶ 182] | Motorstatus (Motorstatus Benennung)            |
| FB_READ_156_SNVT_pumpset_mn [▶ 183]  | Pumpengruppe                                   |
| FB_READ_157_SNVT_ex_control [▶ 184]  | Exklusive Kontrolle                            |
| FB_READ_158_SNVT_pumpset_sn [▶ 185]  | Pumpengruppensensor                            |
| FB_READ_159_SNVT_pump_sensor [▶ 186] | Pumpensensor                                   |
| FB_READ_160_SNVT_abs_humid [▶ 186]   | Absolute Feuchtigkeit                          |
| FB_READ_161_SNVT_flow_p [▶ 187]      | Volumenstrom (Kubikmeter / Stunde)             |
| FB_READ_162_SNVT_dev_c_mode [▶ 188]  | Gerätebetriebsart                              |
| FB_READ_163_SNVT_valve_mode [▶ 189]  | Ventilzustand                                  |
| FB_READ_164_SNVT_alarm_2 [▶ 189]     | Alarmstatus 2                                  |
| FB_READ_165_SNVT_state_64 [▶ 190]    | Statusinformation (64 individuelle Bitwerte)   |
| FB_READ_166_SNVT_nv_type [▶ 191]     | Netzwerkvariablentyp                           |
| FB_READ_168_SNVT_ent_opmode [▶ 192]  | Betriebsart für Zutrittsobjekte                |
| FB_READ_169_SNVT_ent_state [▶ 193]   | Zustand für Zutrittsobjekte                    |



| Bausteine   | Beschreibung  |
|---|---|
| FB_READ_170_SNVT_ent_status [ <a href="#">▶ 193</a> ]     | Status von Zutrittsobjekten                           |
| FB_READ_171_SNVT_flow_dir [ <a href="#">▶ 194</a> ]       | Flow direction  |
| FB_READ_172_SNVT_hvac_satsts [ <a href="#">▶ 195</a> ]    | HLK Sättigungsstatus                                  |
| FB_READ_173_SNVT_dev_status [ <a href="#">▶ 196</a> ]     | Gerätestatus  |
| FB_READ_174_SNVT_dev_fault [ <a href="#">▶ 197</a> ]      | Fehlerstatus  |
| FB_READ_175_SNVT_dev_maint [ <a href="#">▶ 197</a> ]      | Geräte Wartungsstatus                                 |
| FB_READ_176_SNVT_date_event [ <a href="#">▶ 198</a> ]     | Status eines Ereignisses                              |
| FB_READ_177_SNVT_sched_val [ <a href="#">▶ 199</a> ]      | Kalenderwert  |
| FB_READ_180_SNVT_sbld_state [ <a href="#">▶ 200</a> ]     | Jalousie Status                                       |
| FB_READ_181_SNVT_rac_ctrl [ <a href="#">▶ 201</a> ]       | Steuerung Soundfunktion                               |
| FB_READ_182_SNVT_rac_req [ <a href="#">▶ 201</a> ]        | Anforderung Soundfunktion                             |
| FB_READ_183_SNVT_count_32 [ <a href="#">▶ 202</a> ]       | Absoluter Zähler                                      |
| FB_READ_184_SNVT_clothes_w_c [ <a href="#">▶ 203</a> ]    | Waschmaschine / Kommandos                             |
| FB_READ_185_SNVT_clothes_w_m [ <a href="#">▶ 204</a> ]    | Waschmaschine / Management-Status                     |
| FB_READ_186_SNVT_clothes_w_s [ <a href="#">▶ 205</a> ]    | Waschmaschine / Status                                |
| FB_READ_187_SNVT_clothes_w_a [ <a href="#">▶ 205</a> ]    | Waschmaschine / Alarmmeldungen                        |
| FB_READ_188_SNVT_multiplier_s [ <a href="#">▶ 206</a> ]   | Multiplikator   |
| FB_READ_189_SNVT_switch_2 [ <a href="#">▶ 207</a> ]       | Schalter zum einstellen von Szenen und Einstellungen. |
| FB_READ_190_SNVT_color_2 [ <a href="#">▶ 208</a> ]        | Farbe   |
| FB_READ_191_SNVT_log_status [ <a href="#">▶ 209</a> ]     | Log Status  |
| FB_READ_192_SNVT_time_stamp_p [ <a href="#">▶ 209</a> ]   | Präziser Zeitstempel                                  |
| FB_READ_193_SNVT_log_fx_request [ <a href="#">▶ 210</a> ] | Log file transfer Anforderung                         |
| FB_READ_194_SNVT_log_fx_status [ <a href="#">▶ 211</a> ]  | Log file transfer Status                              |
| FB_READ_195_SNVT_log_request [ <a href="#">▶ 212</a> ]    | Log Status Anforderung                                |
| FB_READ_196_SNVT_enthalpy_d [ <a href="#">▶ 213</a> ]     | Enthalpie Unterschied(kJ/kg)                          |
| FB_READ_197_SNVT_amp_ac_mil [ <a href="#">▶ 213</a> ]     | Elektrischer Strom (Milliampere)                      |
| FB_READ_198_SNVT_time_hour_p [ <a href="#">▶ 214</a> ]    | Zeit in Stunden                                       |
| FB_READ_199_SNVT_lamp_status [ <a href="#">▶ 215</a> ]    | Lampen Status   |
| FB_READ_200_SNVT_environment [ <a href="#">▶ 216</a> ]    | Umgebung  |
| FB_READ_201_SNVT_geo_loc [ <a href="#">▶ 216</a> ]        | Geographische Lage                                    |

**Bausteine/Send**

| Bausteine   | Beschreibung                               |
|---|--|
| FB_SEND_001_SNVT_amp [ <a href="#">▶ 217</a> ]        | Elektrischer Strom (Ampere)                |
| FB_SEND_002_SNVT_amp_mil [ <a href="#">▶ 219</a> ]    | Elektrischer Strom (Milliampere)           |
| FB_SEND_003_SNVT_angle [ <a href="#">▶ 220</a> ]      | Winkelabstand (Bogenmaß)                   |
| FB_SEND_004_SNVT_angle_vel [ <a href="#">▶ 221</a> ]  | Winkelgeschwindigkeit (Bogenmaß / Sekunde) |
| FB_SEND_005_SNVT_btu_kilo [ <a href="#">▶ 223</a> ]   | Thermische Energie (Kilo-Btus)             |
| FB_SEND_006_SNVT_btu_mega [ <a href="#">▶ 224</a> ]   | Thermische Energie (Mega-Btus)             |
| FB_SEND_007_SNVT_char_ascii [ <a href="#">▶ 225</a> ] | ASCII Zeichen (8-bit ASCII character)      |
| FB_SEND_008_SNVT_count [ <a href="#">▶ 227</a> ]      | Absolute Anzahl (Stück)                    |
| FB_SEND_009_SNVT_count_inc [ <a href="#">▶ 228</a> ]  | Inkrement Zähler (Stück(delta))            |
| FB_SEND_011_SNVT_date_day [ <a href="#">▶ 229</a> ]   | Wochentag                                  |

| <b>Bausteine</b>                                     | <b>Beschreibung</b>                                |
|--|--|
| <a href="#">FB_SEND_013_SNVT_elec_kwh [▶ 231]</a>    | Elektrische Energie (Kilowatt / Stunde)            |
| <a href="#">FB_SEND_014_SNVT_elec_whr [▶ 232]</a>    | Elektrische Energie (Watt / Stunde)                |
| <a href="#">FB_SEND_015_SNVT_flow [▶ 233]</a>        | Volumenstrom (Liter / Sekunde)                     |
| <a href="#">FB_SEND_016_SNVT_flow_mil [▶ 235]</a>    | Volumenstrom (Milliliter / Sekunde)                |
| <a href="#">FB_SEND_017_SNVT_length [▶ 236]</a>      | Länge (Meter)                                      |
| <a href="#">FB_SEND_018_SNVT_length_kilo [▶ 237]</a> | Länge (Kilometer)                                  |
| <a href="#">FB_SEND_019_SNVT_length_micr [▶ 239]</a> | Länge (Mikrometer)                                 |
| <a href="#">FB_SEND_020_SNVT_length_mil [▶ 240]</a>  | Länge (Millimeter)                                 |
| <a href="#">FB_SEND_021_SNVT_lev_cont [▶ 241]</a>    | Stufenlose Wertänderung (% vom obersten Niveau)    |
| <a href="#">FB_SEND_023_SNVT_mass [▶ 243]</a>        | Masse (Gramm)                                      |
| <a href="#">FB_SEND_024_SNVT_mass_kilo [▶ 244]</a>   | Masse (Kilogramm)                                  |
| <a href="#">FB_SEND_025_SNVT_mass_mega [▶ 245]</a>   | Masse (Metrische Tonne)                            |
| <a href="#">FB_SEND_026_SNVT_mass_mil [▶ 247]</a>    | Masse (Milligramm)                                 |
| <a href="#">FB_SEND_027_SNVT_power [▶ 248]</a>       | Leistung (Watt)                                    |
| <a href="#">FB_SEND_028_SNVT_power_kilo [▶ 249]</a>  | Leistung (Kilowatt)                                |
| <a href="#">FB_SEND_029_SNVT_ppm [▶ 251]</a>         | Konzentration (ppm)                                |
| <a href="#">FB_SEND_030_SNVT_press [▶ 252]</a>       | Druck (Überdruck) (Kilopascal)                     |
| <a href="#">FB_SEND_031_SNVT_res [▶ 253]</a>         | Elektrischer Widerstand (Ohm)                      |
| <a href="#">FB_SEND_032_SNVT_res_kilo [▶ 255]</a>    | Elektrischer Widerstand (Kilohm)                   |
| <a href="#">FB_SEND_033_SNVT_sound_db [▶ 256]</a>    | Schalldruckpegel (dB)                              |
| <a href="#">FB_SEND_034_SNVT_speed [▶ 257]</a>       | Lineare Geschwindigkeit (Meter / Sekunde)          |
| <a href="#">FB_SEND_035_SNVT_speed_mil [▶ 259]</a>   | Lineare Geschwindigkeit (Meter / Sekunde)          |
| <a href="#">FB_SEND_036_SNVT_str_asc [▶ 260]</a>     | Zeichenkette (30 Zeichen max) (ASCII-Zeichenkette) |
| <a href="#">FB_SEND_037_SNVT_str_int [▶ 261]</a>     | Wide Character String                              |
| <a href="#">FB_SEND_038_SNVT_telcom [▶ 262]</a>      | Telefonstatus (Telefonstatus Namen)                |
| <a href="#">FB_SEND_039_SNVT_temp [▶ 264]</a>        | Temperatur (Grad Celsius)                          |
| <a href="#">FB_SEND_041_SNVT_vol [▶ 265]</a>         | Volumen (Liter)                                    |
| <a href="#">FB_SEND_042_SNVT_vol_kilo [▶ 266]</a>    | Volumen (Kiloliter)                                |
| <a href="#">FB_SEND_043_SNVT_vol_mil [▶ 268]</a>     | Volumen (Milliliters)                              |
| <a href="#">FB_SEND_044_SNVT_volt [▶ 269]</a>        | Elektrische Spannung (Volt)                        |
| <a href="#">FB_SEND_045_SNVT_volt_dbmv [▶ 270]</a>   | Elektrische Spannung (dB Mikrovolt)                |
| <a href="#">FB_SEND_046_SNVT_volt_kilo [▶ 272]</a>   | Elektrische Spannung (Kilovolt)                    |
| <a href="#">FB_SEND_047_SNVT_volt_mil [▶ 273]</a>    | Elektrische Spannung (Millivolt)                   |
| <a href="#">FB_SEND_048_SNVT_amp_f [▶ 274]</a>       | Elektrischer Strom (Ampere)                        |
| <a href="#">FB_SEND_049_SNVT_angle_f [▶ 276]</a>     | Winkelabstand (Bogenmaß)                           |
| <a href="#">FB_SEND_050_SNVT_angle_vel_f [▶ 277]</a> | Winkelgeschwindigkeit (Radiant / Sekunde)          |
| <a href="#">FB_SEND_051_SNVT_count_f [▶ 278]</a>     | Absolute Anzahl (Stück)                            |
| <a href="#">FB_SEND_052_SNVT_count_inc_f [▶ 280]</a> | Inkrement Zähler (Stück(delta))                    |
| <a href="#">FB_SEND_053_SNVT_flow_f [▶ 281]</a>      | Volumenstrom (Liter / Sekunde)                     |
| <a href="#">FB_SEND_054_SNVT_length_f [▶ 282]</a>    | Länge (Meter)                                      |
| <a href="#">FB_SEND_055_SNVT_lev_cont_f [▶ 284]</a>  | Stufenlose Wertänderung (% vom obersten Niveau)    |
| <a href="#">FB_SEND_056_SNVT_mass_f [▶ 285]</a>      | Masse (Gramm)                                      |
| <a href="#">FB_SEND_057_SNVT_power_f [▶ 286]</a>     | Leistung (Watt)                                    |



| <b>Bausteine</b>                     | <b>Beschreibung</b>                               |
|--------------------------------------|---|
| FB_SEND_058_SNVT_ppm_f [▶ 288]       | Konzentration (ppm)                               |
| FB_SEND_059_SNVT_press_f [▶ 289]     | Druck (Überdruck) (Pascal)                        |
| FB_SEND_060_SNVT_res_f [▶ 290]       | Elektrischer Widerstand (Ohm)                     |
| FB_SEND_061_SNVT_sound_db_f [▶ 292]  | Schalldruckpegel (dBspl)                          |
| FB_SEND_062_SNVT_speed_f [▶ 293]     | Lineare Geschwindigkeit (Meter / Sekunde)         |
| FB_SEND_063_SNVT_temp_f [▶ 294]      | Temperatur (Grad Celsius)                         |
| FB_SEND_064_SNVT_time_f [▶ 296]      | Abgelaufene Zeit (Sekunden)                       |
| FB_SEND_065_SNVT_vol_f [▶ 297]       | Volumen (Liter)                                   |
| FB_SEND_066_SNVT_volt_f [▶ 298]      | Elektrische Spannung (Volt)                       |
| FB_SEND_067_SNVT_btu_f [▶ 300]       | Thermische Energie (Btus)                         |
| FB_SEND_068_SNVT_elec_whr_f [▶ 301]  | Elektrische Energie (Watt / Stunden)              |
| FB_SEND_069_SNVT_config_src [▶ 302]  | Konfigurationseigenschaften                       |
| FB_SEND_070_SNVT_color [▶ 304]       | Farbe nach CIE Norm                               |
| FB_SEND_071_SNVT_grammage [▶ 305]    | amerikanisches Mass für Papiergewichte und Dichte |
| FB_SEND_072_SNVT_grammage_f [▶ 306]  | amerikanisches Mass für Papiergewichte und Dichte |
| FB_SEND_073_SNVT_file_req [▶ 307]    | Dateianforderung                                  |
| FB_SEND_074_SNVT_file_status [▶ 309] | Dateiinformation                                  |
| FB_SEND_075_SNVT_freq_f [▶ 310]      | Frequenz (Hertz)                                  |
| FB_SEND_076_SNVT_freq_hz [▶ 311]     | Frequenz (Hertz)                                  |
| FB_SEND_077_SNVT_freq_kilohz [▶ 313] | Frequenz (Kilohertz)                              |
| FB_SEND_078_SNVT_freq_milhz [▶ 314]  | Frequenz (Millihertz)                             |
| FB_SEND_079_SNVT_lux [▶ 315]         | Beleuchtungsstärke (Lux)                          |
| FB_SEND_081_SNVT_lev_percent [▶ 317] | Prozentangabe                                     |
| FB_SEND_082_SNVT_multiplier [▶ 318]  | Multiplikator                                     |
| FB_SEND_083_SNVT_state [▶ 319]       | Statusinformation                                 |
| FB_SEND_084_SNVT_time_stamp [▶ 321]  | Zeitstempel                                       |
| FB_SEND_085_SNVT_zerospan [▶ 322]    | Nullpunkt und Proportionalitätsfaktor             |
| FB_SEND_086_SNVT_magcard [▶ 323]     | ISO 7811 (40 Hexadezimal Zahlen)                  |
| FB_SEND_087_SNVT_elapsed_tm [▶ 324]  | Abgelaufene Zeit                                  |
| FB_SEND_088_SNVT_alarm [▶ 325]       | Alarm Status                                      |
| FB_SEND_089_SNVT_currency [▶ 327]    | Währung   |
| FB_SEND_090_SNVT_file_pos [▶ 328]    | Dateiposition                                     |
| FB_SEND_091_SNVT_muldiv [▶ 329]      | Verstärkungsfaktor/Dämpfungsfaktor                |
| FB_SEND_092_SNVT_obj_request [▶ 330] | Funktionsauswahl                                  |
| FB_SEND_093_SNVT_obj_status [▶ 331]  | Objektstatus                                      |
| FB_SEND_094_SNVT_preset [▶ 333]      | Voreinstellung                                    |
| FB_SEND_095_SNVT_switch [▶ 334]      | Schalter  |
| FB_SEND_096_SNVT_trans_table [▶ 335] | Übersetzungstabelle                               |
| FB_SEND_097_SNVT_override [▶ 336]    | Übersteuerungsmodus                               |
| FB_SEND_098_SNVT_pwr_fact [▶ 337]    | Leistungsfaktor                                   |
| FB_SEND_099_SNVT_pwr_fact_f [▶ 339]  | Leistungsfaktor                                   |
| FB_SEND_100_SNVT_density [▶ 340]     | Dichte (kg/m <sup>3</sup> )                       |
| FB_SEND_101_SNVT_density_f [▶ 341]   | Dichte (kg/m <sup>3</sup> )                       |

| Bausteine                            | Beschreibung                                     |
|--------------------------------------|--|
| FB_SEND_102_SNVT_rpm [▶ 343]         | Winkelgeschwindigkeit (Umdrehungen/Minute (RPM)) |
| FB_SEND_103_SNVT_hvac_emerg [▶ 344]  | HLK Notbetrieb                                   |
| FB_SEND_104_SNVT_angle_deg [▶ 345]   | Winkelangabe                                     |
| FB_SEND_105_SNVT_temp_p [▶ 347]      | Temperatur (Grad Celsius)                        |
| FB_SEND_106_SNVT_temp_setpt [▶ 348]  | Temperatur (6 Temperaturwerte)                   |
| FB_SEND_107_SNVT_time_sec [▶ 349]    | Abgelaufene Zeit (Sekunde)                       |
| FB_SEND_108_SNVT_hvac_mode [▶ 350]   | HLK Betriebsmodus                                |
| FB_SEND_109_SNVT_occupancy [▶ 352]   | Präsenzmeldung (Stati)                           |
| FB_SEND_110_SNVT_area [▶ 353]        | Fläche (Quadratmeter)                            |
| FB_SEND_111_SNVT_hvac_overid [▶ 354] | HLK Übersteuerungsmodus                          |
| FB_SEND_112_SNVT_hvac_status [▶ 355] | HLK Status                                       |
| FB_SEND_113_SNVT_press_p [▶ 357]     | Druck (Überdruck) (Pascal)                       |
| FB_SEND_114_SNVT_address [▶ 358]     | Neuron Adresse (16-bit Adress Wert)              |
| FB_SEND_115_SNVT_scene [▶ 359]       | Szenen   |
| FB_SEND_116_SNVT_scene_cfg [▶ 361]   | Szenen Einstellung                               |
| FB_SEND_117_SNVT_setting [▶ 362]     | Einstellungen                                    |
| FB_SEND_118_SNVT_evap_state [▶ 363]  | Verdampferstatus                                 |
| FB_SEND_119_SNVT_therm_mode [▶ 364]  | Thermostatmodus                                  |
| FB_SEND_120_SNVT_defr_mode [▶ 365]   | Abtaumodus                                       |
| FB_SEND_121_SNVT_defr_term [▶ 367]   | Abschluss des Abtauvorgangs                      |
| FB_SEND_122_SNVT_defr_state [▶ 368]  | Abtaustatus (Enumeration)                        |
| FB_SEND_123_SNVT_time_min [▶ 369]    | Abgelaufene Zeit (Minuten)                       |
| FB_SEND_124_SNVT_time_hour [▶ 370]   | Abgelaufen Zeit (Stunde)                         |
| FB_SEND_125_SNVT_ph [▶ 372]          | Säuregehalt (pH)                                 |
| FB_SEND_126_SNVT_ph_f [▶ 373]        | Säuregehalt (pH)                                 |
| FB_SEND_127_SNVT_chlr_status [▶ 374] | Kälteerzeugerstatus                              |
| FB_SEND_128_SNVT_tod_event [▶ 376]   | Präsenzzeit                                      |
| FB_SEND_129_SNVT_smo_obscur [▶ 377]  | Verdunkelung durch Rauch                         |
| FB_SEND_130_SNVT_fire_test [▶ 378]   | Anforderung Test Brandmeldeanlage                |
| FB_SEND_131_SNVT_temp_ror [▶ 379]    | Wert der Temperaturänderung/-erhöhung            |
| FB_SEND_132_SNVT_fire_init [▶ 381]   | Brandmeldertyp                                   |
| FB_SEND_133_SNVT_fire_indcte [▶ 382] | Brandmeldeanzeige                                |
| FB_SEND_134_SNVT_time_zone [▶ 383]   | Zeitzonebeschreibung                             |
| FB_SEND_135_SNVT_earth_pos [▶ 384]   | Position auf der Erde                            |
| FB_SEND_136_SNVT_reg_val [▶ 386]     | Registerwert                                     |
| FB_SEND_137_SNVT_reg_val_ts [▶ 387]  | Registerwert                                     |
| FB_SEND_138_SNVT_volt_ac [▶ 388]     | Wechselspannung                                  |
| FB_SEND_139_SNVT_amp_ac [▶ 389]      | Wechselstrom                                     |
| FB_SEND_143_SNVT_turbidity [▶ 391]   | Trübung  |
| FB_SEND_144_SNVT_turbidity_f [▶ 392] | Trübung  |
| FB_SEND_145_SNVT_hvac_type [▶ 393]   | HLK Anlagentyp                                   |
| FB_SEND_146_SNVT_elec_kwh_l [▶ 395]  | Elektrische Energie (Kilowatt / Stunde)          |

| <b>Bausteine</b>                      | <b>Beschreibung</b>                                   |
|---------------------------------------|---|
| FB_SEND 147 SNVT temp diff p [▶ 396]  | Temperaturdifferenz (Grad Celsius)                    |
| FB_SEND 148 SNVT ctrl req [▶ 397]     | Steuerungsanfrage                                     |
| FB_SEND 149 SNVT ctrl resp [▶ 399]    | Steuerungsantwort                                     |
| FB_SEND 150 SNVT ptz [▶ 400]          | Kamera PTZ (SNZ)                                      |
| FB_SEND 151 SNVT privacyzone [▶ 401]  | Privatzone  |
| FB_SEND 152 SNVT pos ctrl [▶ 402]     | Positionseinstellung für Kameras in Netzwerken        |
| FB_SEND 153 SNVT enthalpy [▶ 404]     | Enthalpie (kiloJoules/kg)                             |
| FB_SEND 154 SNVT gfci status [▶ 405]  | Fehlerstromschutzschalter Status                      |
| FB_SEND 155 SNVT motor state [▶ 406]  | Motorstatus (Motorstatus Benennung)                   |
| FB_SEND 156 SNVT pumpset mn [▶ 407]   | Pumpengruppe  |
| FB_SEND 157 SNVT ex control [▶ 409]   | Exklusive Kontrolle                                   |
| FB_SEND 158 SNVT pumpset sn [▶ 410]   | Pumpengruppensensor                                   |
| FB_SEND 159 SNVT pump sensor [▶ 411]  | Pumpensensor (Geschwindigkeit, Temperatur, Status)    |
| FB_SEND 160 SNVT abs humid [▶ 412]    | Absolute Feuchtigkeit                                 |
| FB_SEND 161 SNVT flow p [▶ 414]       | Volumenstrom (Kubikmeter / Stunde)                    |
| FB_SEND 162 SNVT dev c mode [▶ 415]   | Gerätebetriebsart                                     |
| FB_SEND 163 SNVT valve mode [▶ 416]   | Ventilzustand   |
| FB_SEND 164 SNVT alarm 2 [▶ 417]      | Alarmstatus 2   |
| FB_SEND 165 SNVT state 64 [▶ 419]     | Statusinformation (64 individuelle Bitwerte)          |
| FB_SEND 166 SNVT nv_type [▶ 420]      | Netzwerkvariablentyp                                  |
| FB_SEND 168 SNVT ent_opmode [▶ 421]   | Betriebsart für Zutrittsobjekte                       |
| FB_SEND 169 SNVT ent_state [▶ 422]    | Zustand für Zutrittsobjekte                           |
| FB_SEND 170 SNVT ent_status [▶ 423]   | Status von Zutrittsobjekten                           |
| FB_SEND 171 SNVT flow_dir [▶ 425]     | Gangrichtung  |
| FB_SEND 172 SNVT hvac satsts [▶ 426]  | HLK Sättigungsstatus                                  |
| FB_SEND 173 SNVT dev_status [▶ 427]   | Gerätestatus  |
| FB_SEND 174 SNVT dev_fault [▶ 428]    | Fehlerstatus  |
| FB_SEND 175 SNVT dev_maint [▶ 430]    | Geräte Wartungsstatus                                 |
| FB_SEND 176 SNVT date_event [▶ 431]   | Status eines Ereignisses                              |
| FB_SEND 177 SNVT sched_val [▶ 432]    | Kalenderwert  |
| FB_SEND 180 SNVT sbld state [▶ 433]   | Jalousie Status                                       |
| FB_SEND 181 SNVT rac_ctrl [▶ 435]     | Steuerung Soundfunktion                               |
| FB_SEND 182 SNVT rac_req [▶ 436]      | Anforderung Soundfunktion                             |
| FB_SEND 183 SNVT count 32 [▶ 437]     | Absoluter Zähler                                      |
| FB_SEND 184 SNVT clothes_w_c [▶ 438]  | Waschmaschine / Kommandos                             |
| FB_SEND 185 SNVT clothes_w_m [▶ 440]  | Waschmaschine /Management-Status.                     |
| FB_SEND 186 SNVT clothes_w_s [▶ 441]  | Waschmaschine / Status                                |
| FB_SEND 187 SNVT clothes_w_a [▶ 442]  | Waschmaschine / Status                                |
| FB_SEND 188 SNVT multiplier_s [▶ 443] | Multiplikator   |
| FB_SEND 189 SNVT switch_2 [▶ 445]     | Schalter zum einstellen von Szenen und Einstellungen. |
| FB_SEND 190 SNVT color_2 [▶ 446]      | Farbe   |
| FB_SEND 191 SNVT log_status [▶ 447]   | Log Status  |

| Bausteine                               | Beschreibung                     |
|---|----------------------------------|
| FB SEND 192 SNVT time_stamp_p [▶ 448]   | Präziser Zeitstempel             |
| FB SEND 193 SNVT log_fx_request [▶ 450] | Log file transfer Anforderung    |
| FB SEND 194 SNVT log_fx_status [▶ 451]  | Log file transfer Status.        |
| FB SEND 195 SNVT log_request [▶ 452]    | Log Status Anforderung           |
| FB SEND 196 SNVT enthalpy_d [▶ 453]     | Enthalpie Unterschied(kJ/kg)     |
| FB SEND 197 SNVT amp_ac_mil [▶ 455]     | Elektrischer Strom (Milliampere) |
| FB SEND 198 SNVT time_hour_p [▶ 456]    | Zeit in Stunden                  |
| FB SEND 199 SNVT lamp_status [▶ 457]    | Lampen Status                    |
| FB SEND 200 SNVT environment [▶ 459]    | Umgebung                         |
| FB SEND 201 SNVT geo_loc [▶ 460]        | Geographische Lage               |

**Datentypen/Enums**

| Datentypen                      | Beschreibung   |
|---------------------------------|--|
| E LON_alarm_type_t [▶ 493]      | Genutzt durch: SNVT_alarm / SNVT_alarm_2   |
| E LON_appl_cwc_t [▶ 495]        | Genutzt durch: SNVT_clothes_w_c / SNVT_clothes_w_s   |
| E LON_appl_cwp_t [▶ 495]        | Genutzt durch: SNVT_clothes_w_c  |
| E LON_appl_cws_t [▶ 496]        | Genutzt durch: SNVT_clothes_w_c / SNVT_clothes_w_s   |
| E LON_appl_rin_t [▶ 497]        | Genutzt durch: SNVT_clothes_w_c  |
| E LON_boolean_t [▶ 497]         | Genutzt durch: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPThighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn |
| E LON_calendar_type_t [▶ 497]   | Genutzt durch: SNVT_time_zone  |
| E LON_cam_act_t [▶ 497]         | Genutzt durch: SNVT_pos_ctrl   |
| E LON_cam_func_t [▶ 498]        | Genutzt durch: SNVT_pos_ctrl   |
| E LON_chiller_t [▶ 498]         | Genutzt durch: SNVT_chlr_status  |
| E LON_color_encoding_t [▶ 498]  | Genutzt durch: SNVT_color_2  |
| E LON_config_source_t [▶ 499]   | Genutzt durch: SNVT_config_src   |
| E LON_control_resp_t [▶ 499]    | Genutzt durch: SNVT_ctrl_resp  |
| E LON_currency_t [▶ 499]        | Genutzt durch: SNVT_currency   |
| E LON_days_of_week_t [▶ 502]    | Genutzt durch: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone   |
| E LON_defrost_mode_t [▶ 502]    | Genutzt durch: SNVT_defr_mode  |
| E LON_defrost_state_t [▶ 503]   | Genutzt durch: SNVT_defr_state   |
| E LON_defrost_term_t [▶ 503]    | Genutzt durch: SNVT_defr_term  |
| E LON_device_c_mode_t [▶ 504]   | Genutzt durch: SNVT_dev_c_mode   |
| E LON_device_select_t [▶ 505]   | Genutzt durch: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status   |
| E LON_discrete_levels_t [▶ 505] | Genutzt durch: SNVT_clothes_w_c / SNVT_lev_disc  |
| E LON_emerg_t [▶ 505]           | Genutzt durch: SNVT_hvac_emerg   |

| <b>Datentypen</b>                                      | <b>Beschreibung</b>   |
|--|---|
| <a href="#">E_LON_ent_cmd_t [► 506]</a>                | Genutzt durch: SNVT_ent_state                                       |
| <a href="#">E_LON_ent_opmode_cmd_t [► 506]</a>         | Genutzt durch: SNVT_ent_opmode / SNVT_ent_status                    |
| <a href="#">E_LON_evap_t [► 507]</a>                   | Genutzt durch: SNVT_evap_state                                      |
| <a href="#">E_LON_ex_control_t [► 508]</a>             | Genutzt durch: SNVT_ex_control                                      |
| <a href="#">E_LON_file_request_t [► 508]</a>           | Genutzt durch: SNVT_file_req  |
| <a href="#">E_LON_file_status_t [► 508]</a>            | Genutzt durch: SNVT_file_status                                     |
| <a href="#">E_LON_fire_indicator_t [► 509]</a>         | Genutzt durch: SNVT_fire_indcte                                     |
| <a href="#">E_LON_fire_initiator_t [► 510]</a>         | Genutzt durch: SNVT_fire_init                                       |
| <a href="#">E_LON_fire_test_t [► 510]</a>              | Genutzt durch: SNVT_fire_test                                       |
| <a href="#">E_LON_flow_direction_t [► 511]</a>         | Genutzt durch: SNVT_flow_dir  |
| <a href="#">E_LON_gfci_status_t [► 511]</a>            | Genutzt durch: SNVT_gfci_status                                     |
| <a href="#">E_LON_hvac_hvt_t [► 511]</a>               | Genutzt durch: SNVT_hvac_type                                       |
| <a href="#">E_LON_hvac_overid_t [► 512]</a>            | Genutzt durch: SNVT_hvac_overid                                     |
| <a href="#">E_LON_hvac_t [► 514]</a>                   | Genutzt durch: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status |
| <a href="#">E_LON_learn_mode_t [► 515]</a>             | Genutzt durch: SNVT_preset  |
| <a href="#">E_LON_log_status_t [► 516]</a>             | Genutzt durch: SCPTlogRecord / SNVT_log_status                      |
| <a href="#">E_LON_motor_state_t [► 516]</a>            | Genutzt durch: SNVT_motor_state / SNVT_pumpset_mn                   |
| <a href="#">E_LON_nv_type_category_t [► 516]</a>       | Genutzt durch: SNVT_nv_type   |
| <a href="#">E_LON_object_request_t [► 517]</a>         | Genutzt durch: SNVT_obj_request                                     |
| <a href="#">E_LON_occup_t [► 518]</a>                  | Genutzt durch: SNVT_occupancy / SNVT_tod_event                      |
| <a href="#">E_LON_override_t [► 518]</a>               | Genutzt durch: SNVT_override  |
| <a href="#">E_LON_pan_dir_t [► 519]</a>                | Genutzt durch: SNVT_ptz   |
| <a href="#">E_LON_priority_level_t [► 519]</a>         | Genutzt durch: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn          |
| <a href="#">E_LON_privacyzone_t [► 520]</a>            | Genutzt durch: SNVT_privacyzone                                     |
| <a href="#">E_LON_rail_audio_sensor_type_t [► 520]</a> | Genutzt durch: SNVT_rac_ctrl / SNVT_rac_req                         |
| <a href="#">E_LON_rail_audio_type_t [► 521]</a>        | Genutzt durch: SNVT_rac_ctrl / SNVT_rac_req                         |
| <a href="#">E_LON_reg_val_unit_t [► 522]</a>           | Genutzt durch: SNVT_reg_val / SNVT_reg_val_ts                       |
| <a href="#">E_LON_sbld cmd source_t [► 524]</a>        | Genutzt durch: SNVT_sbld_state                                      |
| <a href="#">E_LON_sbld_error_t [► 526]</a>             | Genutzt durch: SNVT_sbld_state                                      |
| <a href="#">E_LON_scene_config_t [► 526]</a>           | Genutzt durch: SNVT_scene_cfg                                       |
| <a href="#">E_LON_scene_t [► 527]</a>                  | Genutzt durch: SNVT_scene   |
| <a href="#">E_LON_sec_state_t [► 528]</a>              | Genutzt durch: SNVT_sec_state                                       |
| <a href="#">E_LON_sec_status_t [► 529]</a>             | Genutzt durch: SNVT_sec_status                                      |
| <a href="#">E_LON_setting_t [► 530]</a>                | Genutzt durch: SNVT_setting   |
| <a href="#">E_LON_switch_state_t [► 531]</a>           | Genutzt durch: SNVT_switch_2  |
| <a href="#">E_LON_telcom_states_t [► 533]</a>          | Genutzt durch: SNVT_telcom  |
| <a href="#">E_LON_therm_mode_t [► 534]</a>             | Genutzt durch: SNVT_therm_mode                                      |
| <a href="#">E_LON_tilt_dir_t [► 534]</a>               | Genutzt durch: SNVT_ptz   |
| <a href="#">E_LON_unit_temp_t [► 534]</a>              | Genutzt durch: SNVT_pump_sensor                                     |
| <a href="#">E_LON_valve_mode_t [► 535]</a>             | Genutzt durch: SNVT_valve_mode                                      |
| <a href="#">E_LON_zoom_t [► 535]</a>                   | Genutzt durch: SNVT_ptz   |

**Datentypen/Hardware Types**

| Datentypen                                       | Beschreibung               |
|--|----------------------------|
| <a href="#">ST_LON_Parameter_IN_36B [▶ 560]</a>  | Prozessabbild der Eingänge |
| <a href="#">ST_LON_Parameter_OUT_36B [▶ 561]</a> | Prozessabbild der Ausgänge |

**Datentypen/LON\_TYPES**

| Datentypen  | Beschreibung     |
|---|------------------|
| <a href="#">E_LON_Parameter_Datatypes [▶ 485]</a> | Enums SNVT Typen |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_chlr\_status**

| Datentypen                                | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_chlr_state [▶ 538]</a> | Genutzt durch: SNVT_chlr_status |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_clothes\_w\_c**

| Datentypen                              | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_action [▶ 538]</a>   | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_dry [▶ 539]</a>      | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_duration [▶ 539]</a> | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_function [▶ 539]</a> | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_rinse [▶ 539]</a>    | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_spin [▶ 540]</a>     | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_wash [▶ 540]</a>     | Genutzt durch: SNVT_clothes_w_c |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_clothes\_w\_s**

| Datentypen                           | Beschreibung                    |
|--------------------------------------|---------------------------------|
| <a href="#">ST_LON_alarm [▶ 540]</a> | Genutzt durch: SNVT_clothes_w_s |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_color\_2**

| Datentypen                                     | Beschreibung                |
|--|-----------------------------|
| <a href="#">ST_LON_CIE1931_lumen [▶ 542]</a>   | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_CIE1931_percent [▶ 542]</a> | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_color_value [▶ 543]</a>     | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_RGB [▶ 543]</a>             | Genutzt durch: SNVT_color_2 |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_ctrl\_resp**

| Datentypen                            | Beschreibung                  |
|---------------------------------------|-------------------------------|
| <a href="#">ST_LON_range [▶ 543]</a>  | Genutzt durch: SNVT_ctrl_resp |
| <a href="#">ST_LON_sender [▶ 544]</a> | Genutzt durch: SNVT_ctrl_resp |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_dev\_fault**

| Datentypen                                | Beschreibung                  |
|---|-------------------------------|
| <a href="#">ST_LON_Dev_type1 [▶ 544]</a>  | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_pump_ctrl1 [▶ 544]</a> | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_valve_pos1 [▶ 545]</a> | Genutzt durch: SNVT_dev_fault |



**Datentypen/Structure/AuxiliaryStructure/SNVT\_dev\_maint**

| Datentypen  | Beschreibung                  |
|---|-------------------------------|
| <a href="#">ST_LON_Dev_type2</a> [ <a href="#">▶ 546</a> ]  | Genutzt durch: SNVT_dev_maint |
| <a href="#">ST_LON_pump_ctrl2</a> [ <a href="#">▶ 546</a> ] | Genutzt durch: SNVT_dev_maint |
| <a href="#">ST_LON_valve_pos2</a> [ <a href="#">▶ 547</a> ] | Genutzt durch: SNVT_dev_maint |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_dev\_status**

| Datentypen  | Beschreibung                   |
|---|--------------------------------|
| <a href="#">ST_LON_Dev_type3</a> [ <a href="#">▶ 547</a> ]  | Genutzt durch: SNVT_dev_status |
| <a href="#">ST_LON_pump_ctrl3</a> [ <a href="#">▶ 548</a> ] | Genutzt durch: SNVT_dev_status |
| <a href="#">ST_LON_valve_pos3</a> [ <a href="#">▶ 549</a> ] | Genutzt durch: SNVT_dev_status |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_ex\_control**

| Datentypen   | Beschreibung                   |
|--|--------------------------------|
| <a href="#">ST_LON_Control_device_addr</a> [ <a href="#">▶ 549</a> ] | Genutzt durch: SNVT_ex_control |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_file\_req**

| Datentypen  | Beschreibung                 |
|---|------------------------------|
| <a href="#">ST_LON_addrt</a> [ <a href="#">▶ 550</a> ]        | Genutzt durch: SNVT_file_req |
| <a href="#">ST_LON_dest_address</a> [ <a href="#">▶ 550</a> ] | Genutzt durch: SNVT_file_req |
| <a href="#">ST_LON_gp</a> [ <a href="#">▶ 550</a> ]           | Genutzt durch: SNVT_file_req |
| <a href="#">ST_LON_sn</a> [ <a href="#">▶ 551</a> ]           | Genutzt durch: SNVT_file_req |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_file\_status**

| Datentypen  | Beschreibung  |
|---|---|
| <a href="#">ST_LON_address</a> [ <a href="#">▶ 551</a> ]    | Genutzt durch: FB_Write_Address_Table / FB_Read_Address_Table |
| <a href="#">ST_LON_adr</a> [ <a href="#">▶ 551</a> ]        | Genutzt durch: SNVT_file_status                               |
| <a href="#">ST_LON_descriptor</a> [ <a href="#">▶ 552</a> ] | Genutzt durch: SNVT_file_status                               |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_lamp\_status**

| Datentypen  | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_Alarm_actual</a> [ <a href="#">▶ 552</a> ]   | Genutzt durch: SNVT_lamp_status |
| <a href="#">ST_LON_alarm_previous</a> [ <a href="#">▶ 554</a> ] | Genutzt durch: SNVT_lamp_status |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_pos\_ctrl**

| Datentypen  | Beschreibung                 |
|---|------------------------------|
| <a href="#">ST_LON_abspos</a> [ <a href="#">▶ 556</a> ] | Genutzt durch: SNVT_pos_ctrl |
| <a href="#">ST_LON_Value</a> [ <a href="#">▶ 556</a> ]  | Genutzt durch: SNVT_pos_ctrl |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_rac\_ctrl**

| Datentypen   | Beschreibung                                |
|--|---|
| <a href="#">ST_LON_addr_dest</a> [ <a href="#">▶ 556</a> ] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_addr_init</a> [ <a href="#">▶ 557</a> ] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |

| Datentypen                               | Beschreibung                                |
|--|---|
| <a href="#">ST_LON_addr_talk [▶ 557]</a> | Genutzt durch: SNVT_rac_ctrl                |
| <a href="#">ST_LON_p2m [▶ 557]</a>       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_p2p [▶ 557]</a>       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_rac\_req**

| Datentypen                                       | Beschreibung |
|--|--------------|
| <a href="#">ST_LON_rac_req_addr_dest [▶ 558]</a> |              |
| <a href="#">ST_LON_rac_req_addr_init [▶ 558]</a> |              |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_switch\_2**

| Datentypen                             | Beschreibung                 |
|--|------------------------------|
| <a href="#">ST_LON_setting [▶ 559]</a> | Genutzt durch: SNVT_switch_2 |

**Datentypen/Structure/AuxiliaryStructure/SNVT\_time\_zone**

| Datentypen                                 | Beschreibung                  |
|--|-------------------------------|
| <a href="#">ST_LON_end_DST [▶ 559]</a>     | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_M_end_DST [▶ 560]</a>   | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_M_start_DST [▶ 560]</a> | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_start_DST [▶ 560]</a>   | Genutzt durch: SNVT_time_zone |

**Datentypen/Structure**

| Datentypen                                      | Beschreibung  |
|---|---|
| <a href="#">ST_KL6401 [▶ 563]</a>               | Struktur für die Konfiguration                                |
| <a href="#">ST_LON_AddressTable [▶ 564]</a>     | Genutzt durch: FB_Write_Address_Table / FB_Read_Address_Table |
| <a href="#">ST_LON_ConfigTable [▶ 564]</a>      | Genutzt durch: FB_Write_Config_Table / FB_Read_Config_Table   |
| <a href="#">ST_LON_DomainTable [▶ 565]</a>      | Genutzt durch: FB_Write_Domain_Table / FB_Read_Domain_Table   |
| <a href="#">ST_LON_SNVT_alarm [▶ 565]</a>       | Genutzt durch: SNVT_alarm                                     |
| <a href="#">ST_LON_SNVT_alarm_2 [▶ 566]</a>     | Genutzt durch: SNVT_alarm_2                                   |
| <a href="#">ST_LON_SNVT_chlr_status [▶ 566]</a> | Genutzt durch: SNVT_chlr_status                               |
| <a href="#">ST_LON_SNVT_clothes_w_a [▶ 566]</a> | Genutzt durch: SNVT_clothes_w_a                               |
| <a href="#">ST_LON_SNVT_clothes_w_c [▶ 568]</a> | Genutzt durch: SNVT_clothes_w_c                               |
| <a href="#">ST_LON_SNVT_clothes_w_m [▶ 568]</a> | Genutzt durch: SNVT_clothes_w_m                               |
| <a href="#">ST_LON_SNVT_clothes_w_s [▶ 569]</a> | Genutzt durch: SNVT_clothes_w_s                               |
| <a href="#">ST_LON_SNVT_color [▶ 569]</a>       | Genutzt durch: SNVT_color                                     |
| <a href="#">ST_LON_SNVT_color_2 [▶ 569]</a>     | Genutzt durch: SNVT_color_2                                   |
| <a href="#">ST_LON_SNVT_ctrl_req [▶ 569]</a>    | Genutzt durch: SNVT_ctrl_req                                  |
| <a href="#">ST_LON_SNVT_ctrl_resp [▶ 570]</a>   | Genutzt durch: SNVT_ctrl_resp                                 |
| <a href="#">ST_LON_SNVT_currency [▶ 570]</a>    | Genutzt durch: SNVT_currency                                  |
| <a href="#">ST_LON_SNVT_date_event [▶ 570]</a>  | Genutzt durch: SNVT_date_event                                |
| <a href="#">ST_LON_SNVT_dev_fault [▶ 570]</a>   | Genutzt durch: SNVT_dev_fault                                 |
| <a href="#">ST_LON_SNVT_dev_maint [▶ 571]</a>   | Genutzt durch: SNVT_dev_maint                                 |



| <b>Datentypen</b>                                  | <b>Beschreibung</b>                |
|--|------------------------------------|
| <a href="#">ST_LON_SNVT_dev_status [▶ 571]</a>     | Genutzt durch: SNVT_dev_status     |
| <a href="#">ST_LON_SNVT_earth_pos [▶ 571]</a>      | Genutzt durch: SNVT_earth_pos      |
| <a href="#">ST_LON_SNVT_elapsed_tm [▶ 572]</a>     | Genutzt durch: SNVT_elapsed_tm     |
| <a href="#">ST_LON_SNVT_ent_status [▶ 572]</a>     | Genutzt durch: SNVT_ent_status     |
| <a href="#">ST_LON_SNVT_environment [▶ 573]</a>    | Genutzt durch: SNVT_environment    |
| <a href="#">ST_LON_SNVT_ex_control [▶ 574]</a>     | Genutzt durch: SNVT_ex_control     |
| <a href="#">ST_LON_SNVT_file_pos [▶ 574]</a>       | Genutzt durch: SNVT_file_pos       |
| <a href="#">ST_LON_SNVT_file_req [▶ 574]</a>       | Genutzt durch: SNVT_file_req       |
| <a href="#">ST_LON_SNVT_file_status [▶ 575]</a>    | Genutzt durch: SNVT_file_status    |
| <a href="#">ST_LON_SNVT_geo_loc [▶ 575]</a>        | Genutzt durch: SNVT_geo_loc        |
| <a href="#">ST_LON_SNVT_hvac_overid [▶ 575]</a>    | Genutzt durch: SNVT_hvac_overid    |
| <a href="#">ST_LON_SNVT_hvac_satsts [▶ 575]</a>    | Genutzt durch: SNVT_hvac_satsts    |
| <a href="#">ST_LON_SNVT_hvac_status [▶ 576]</a>    | Genutzt durch: SNVT_hvac_status    |
| <a href="#">ST_LON_SNVT_lamp_status [▶ 577]</a>    | Genutzt durch: SNVT_lamp_status    |
| <a href="#">ST_LON_SNVT_log_fx_request [▶ 577]</a> | Genutzt durch: SNVT_log_fx_request |
| <a href="#">ST_LON_SNVT_log_fx_status [▶ 577]</a>  | Genutzt durch: SNVT_log_fx_status  |
| <a href="#">ST_LON_SNVT_log_status [▶ 578]</a>     | Genutzt durch: SNVT_log_status     |
| <a href="#">ST_LON_SNVT_muldiv [▶ 578]</a>         | Genutzt durch: SNVT_muldiv         |
| <a href="#">ST_LON_SNVT_nv_type [▶ 578]</a>        | Genutzt durch: SNVT_nv_type        |
| <a href="#">ST_LON_SNVT_obj_request [▶ 579]</a>    | Genutzt durch: SNVT_obj_request    |
| <a href="#">ST_LON_SNVT_obj_status [▶ 579]</a>     | Genutzt durch: SNVT_obj_status     |
| <a href="#">ST_LON_SNVT_pos_ctrl [▶ 580]</a>       | Genutzt durch: SNVT_pos_ctrl       |
| <a href="#">ST_LON_SNVT_preset [▶ 581]</a>         | Genutzt durch: SNVT_preset         |
| <a href="#">ST_LON_SNVT_privacyzone [▶ 581]</a>    | Genutzt durch: SNVT_privacyzone    |
| <a href="#">ST_LON_SNVT_ptz [▶ 582]</a>            | Genutzt durch: SNVT_ptz            |
| <a href="#">ST_LON_SNVT_pump_sensor [▶ 582]</a>    | Genutzt durch: SNVT_pump_sensor    |
| <a href="#">ST_LON_SNVT_pumpset_mn [▶ 583]</a>     | Genutzt durch: SNVT_pumpset_mn     |
| <a href="#">ST_LON_SNVT_pumpset_sn [▶ 583]</a>     | Genutzt durch: SNVT_pumpset_sn     |
| <a href="#">ST_LON_SNVT_rac_ctrl [▶ 584]</a>       | Genutzt durch: SNVT_rac_ctrl       |
| <a href="#">ST_LON_SNVT_rac_req [▶ 584]</a>        | Genutzt durch: SNVT_rac_req        |
| <a href="#">ST_LON_SNVT_reg_val [▶ 585]</a>        | Genutzt durch: SNVT_rac_val        |
| <a href="#">ST_LON_SNVT_reg_val_ts [▶ 585]</a>     | Genutzt durch: SNVT_rac_val_ts     |
| <a href="#">ST_LON_SNVT_sbld state [▶ 585]</a>     | Genutzt durch: SNVT_sbld_state     |
| <a href="#">ST_LON_SNVT_scene [▶ 586]</a>          | Genutzt durch: SNVT_scene          |
| <a href="#">ST_LON_SNVT_scene_cfg [▶ 586]</a>      | Genutzt durch: SNVT_scene_cfg      |
| <a href="#">ST_LON_SNVT_setting [▶ 586]</a>        | Genutzt durch: SNVT_setting        |
| <a href="#">ST_LON_SNVT_str_int [▶ 587]</a>        | Genutzt durch: SNVT_str_int        |
| <a href="#">ST_LON_SNVT_switch [▶ 587]</a>         | Genutzt durch: SNVT_switch         |
| <a href="#">ST_LON_SNVT_switch_2 [▶ 587]</a>       | Genutzt durch: SNVT_switch_2       |
| <a href="#">ST_LON_SNVT_temp_setpt [▶ 587]</a>     | Genutzt durch: SNVT_temp_setpt     |
| <a href="#">ST_LON_SNVT_time_zone [▶ 588]</a>      | Genutzt durch: SNVT_time_zone      |
| <a href="#">ST_LON_SNVT_tod_event [▶ 588]</a>      | Genutzt durch: SNVT_tod_event      |
| <a href="#">ST_LON_SNVT_trans_table [▶ 589]</a>    | Genutzt durch: SNVT_trans_table    |

| Datentypen                                     | Beschreibung                 |
|--|------------------------------|
| ST_LON_SNVT_zerospan [ <a href="#">▶ 589</a> ] | Genutzt durch: SNVT_zerospan |
| str_AddressTable [ <a href="#">▶ 589</a> ]     | Adress Tabelle               |

**Datentypen**

| Datentypen  | Beschreibung   |
|---|--|
| E_LON_ERROR [ <a href="#">▶ 467</a> ]               | Fehlermeldungen  |
| ST_ExplicitMessage [ <a href="#">▶ 561</a> ]        | Explicit Message   |
| ST_LON_Communication [ <a href="#">▶ 562</a> ]      | Verbindung zwischen "FB_LON_KL6401" und den Sende-/ Empfangsbausteinen |
| ST_LON_ParameterInterface [ <a href="#">▶ 562</a> ] | LON Parameterinterface   |
| ST_LON_WriteData [ <a href="#">▶ 563</a> ]          | Struktur des Sendepuffer   |
| ST_Prm [ <a href="#">▶ 563</a> ]                    | Struktur für Konfiguration   |

**Ressourcen**

| Ressourcen                                      | Beschreibung                         |
|---|--------------------------------------|
| Globale Variablen_LON [ <a href="#">▶ 590</a> ] | Defaultwerte für alle Sendebausteine |

## 7.1 Allgemeine Informationen

● **Installation**

**i** Ab TwinCAT 2.11 Build 2229 (R3 und x64 Engineering) werden die Bibliotheken "TcLON.lib/.lb6/.lbox" standardmäßig mitinstalliert.

● **Name der Bibliothek**

**i** Diese Bibliothek ersetzt die "TcKL6401.lib/.lb6/.lbox". Es hat sich nur der Name der Bibliothek geändert. Die Bausteine sind kompatibel.

Hardware Dokumentation im Beckhoff Information System: [KL6401 - LON-Busklemme](#)

**Weitere erforderliche Bibliotheken**

Für PC-Systeme (x86) und Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib
- TcUtilities.lib

Für Busklemmen-Controller der Serie BCxx00:

- Standard.lb6
- TcPlcUtilitiesBC.lb6
- PlcHelperBC.lb6
- PlcSystemBC.lb6

Für Busklemmen-Controller der Serie BCxx50, BCxx20 und BC9191:

- Standard.lbx
- TcBaseBCxx50.lbx
- TcSystemBCxx50.lbx

Für Busklemmen-Controller der Serie BXxx00:

- Standard.lbx

- TcBaseBX.lbx
- TcSystemBX.lbx

**● Speicherauslastung**

**I** Durch Einbinden der Bibliothek wird bereits SPS-Programmspeicher verbraucht. Abhängig vom Applikationsprogramm kann daher der verbleibende Speicher nicht ausreichend sein.

## 7.2 Bausteine

| Bausteine                            | Beschreibung              |
|--------------------------------------|---------------------------|
| FB_LON_KL6401 <a href="#">[▶ 68]</a> | Sende- / Empfangsbaustein |

**Read**

| Bausteine   | Beschreibung                                    |
|---|---|
| FB_READ_001_SNVT_amp <a href="#">[▶ 70]</a>         | Elektrischer Strom (Ampere)                     |
| FB_READ_002_SNVT_amp_mil <a href="#">[▶ 70]</a>     | Elektrischer Strom (Milliampere)                |
| FB_READ_003_SNVT_angle <a href="#">[▶ 71]</a>       | Winkelabstand (Bogenmaß)                        |
| FB_READ_004_SNVT_angle_vel <a href="#">[▶ 72]</a>   | Winkelgeschwindigkeit (Bogenmaß / Sekunde)      |
| FB_READ_005_SNVT_btu_kilo <a href="#">[▶ 73]</a>    | Thermische Energie (Kilo-Btus)                  |
| FB_READ_006_SNVT_btu_mega <a href="#">[▶ 73]</a>    | Thermische Energie (Mega-Btus)                  |
| FB_READ_007_SNVT_char_ascii <a href="#">[▶ 74]</a>  | ASCII Zeichen (8-bit ASCII character)           |
| FB_READ_008_SNVT_count <a href="#">[▶ 75]</a>       | Absolute Anzahl (Stück)                         |
| FB_READ_009_SNVT_count_inc <a href="#">[▶ 76]</a>   | Inkrement Zähler (Stück(delta))                 |
| FB_READ_011_SNVT_date_day <a href="#">[▶ 76]</a>    | Wochentag                                       |
| FB_READ_013_SNVT_elec_kwh <a href="#">[▶ 77]</a>    | Elektrische Energie (kW/h)                      |
| FB_READ_014_SNVT_elec_whr <a href="#">[▶ 78]</a>    | Elektrische Energie (W/h)                       |
| FB_READ_015_SNVT_flow <a href="#">[▶ 79]</a>        | Volumenstrom (Liter / Sekunde)                  |
| FB_READ_016_SNVT_flow_mil <a href="#">[▶ 80]</a>    | Volumenstrom (Milliliter / Sekunde)             |
| FB_READ_017_SNVT_length <a href="#">[▶ 80]</a>      | Länge (Meter)                                   |
| FB_READ_018_SNVT_length_kilo <a href="#">[▶ 81]</a> | Länge (Kilometer)                               |
| FB_READ_019_SNVT_length_micr <a href="#">[▶ 82]</a> | Länge (Mikrometer)                              |
| FB_READ_020_SNVT_length_mil <a href="#">[▶ 83]</a>  | Länge (Millimeter)                              |
| FB_READ_021_SNVT_lev_cont <a href="#">[▶ 83]</a>    | Stufenlose Wertänderung (% vom obersten Niveau) |
| FB_READ_023_SNVT_mass <a href="#">[▶ 84]</a>        | Masse (Gramm)                                   |
| FB_READ_024_SNVT_mass_kilo <a href="#">[▶ 85]</a>   | Masse (Kilogramm)                               |
| FB_READ_025_SNVT_mass_mega <a href="#">[▶ 86]</a>   | Masse (Metrische Tonne)                         |
| FB_READ_026_SNVT_mass_mil <a href="#">[▶ 86]</a>    | Masse (Milligramm)                              |
| FB_READ_027_SNVT_power <a href="#">[▶ 87]</a>       | Leistung (Watt)                                 |
| FB_READ_028_SNVT_power_kilo <a href="#">[▶ 88]</a>  | Leistung (Kilowatt)                             |
| FB_READ_029_SNVT_ppm <a href="#">[▶ 89]</a>         | Konzentration (ppm)                             |
| FB_READ_030_SNVT_press <a href="#">[▶ 89]</a>       | Druck (Überdruck) (Kilopascal)                  |
| FB_READ_031_SNVT_res <a href="#">[▶ 90]</a>         | Elektrischer Widerstand (Ohm)                   |
| FB_READ_032_SNVT_res_kilo <a href="#">[▶ 91]</a>    | Elektrischer Widerstand (Kilohm)                |
| FB_READ_033_SNVT_sound_db <a href="#">[▶ 92]</a>    | Schalldruckpegel (dB)                           |
| FB_READ_034_SNVT_speed <a href="#">[▶ 92]</a>       | Lineare Geschwindigkeit (Meter / Sekunde)       |
| FB_READ_035_SNVT_speed_mil <a href="#">[▶ 93]</a>   | Lineare Geschwindigkeit (Millimeter / Sekunde)  |

| Bausteine  | Beschreibung                                      |
|--|---|
| <a href="#">FB_READ_036_SNVT_str_asc [▶ 94]</a>      | Zeichenkette                                      |
| <a href="#">FB_READ_037_SNVT_str_int [▶ 95]</a>      | Wide Character String                             |
| <a href="#">FB_READ_038_SNVT_telcom [▶ 95]</a>       | Telefonstatus (Telefonstatus Namen)               |
| <a href="#">FB_READ_039_SNVT_temp [▶ 96]</a>         | Temperatur (Grad Celsius)                         |
| <a href="#">FB_READ_041_SNVT_vol [▶ 97]</a>          | Volumen (Liter)                                   |
| <a href="#">FB_READ_042_SNVT_vol_kilo [▶ 98]</a>     | Volumen (Kiloliter)                               |
| <a href="#">FB_READ_043_SNVT_vol_mil [▶ 99]</a>      | Volumen (Milliliter)                              |
| <a href="#">FB_READ_044_SNVT_volt [▶ 99]</a>         | Elektrische Spannung (Volt)                       |
| <a href="#">FB_READ_045_SNVT_volt_dbmv [▶ 100]</a>   | Elektrische Spannung (dB Mikrovolt)               |
| <a href="#">FB_READ_046_SNVT_volt_kilo [▶ 101]</a>   | Elektrische Spannung (Kilovolt)                   |
| <a href="#">FB_READ_047_SNVT_volt_mil [▶ 102]</a>    | Elektrische Spannung (Millivolt)                  |
| <a href="#">FB_READ_048_SNVT_amp_f [▶ 102]</a>       | Elektrischer Strom (Ampere)                       |
| <a href="#">FB_READ_049_SNVT_angle_f [▶ 103]</a>     | Winkelabstand (Bogenmaß)                          |
| <a href="#">FB_READ_050_SNVT_angle_vel_f [▶ 104]</a> | Winkelgeschwindigkeit (Radiant / Sekunde)         |
| <a href="#">FB_READ_051_SNVT_count_f [▶ 105]</a>     | Absolute Anzahl (Stück)                           |
| <a href="#">FB_READ_052_SNVT_count_inc_f [▶ 105]</a> | Inkrement Zähler (Stück(delta))                   |
| <a href="#">FB_READ_053_SNVT_flow_f [▶ 106]</a>      | Volumenstrom (Liter / Sekunde)                    |
| <a href="#">FB_READ_054_SNVT_length_f [▶ 107]</a>    | Länge (Meter)                                     |
| <a href="#">FB_READ_055_SNVT_lev_cont_f [▶ 108]</a>  | Stufenlose Wertänderung (% vom obersten Niveau)   |
| <a href="#">FB_READ_056_SNVT_mass_f [▶ 108]</a>      | Masse (Gramm)                                     |
| <a href="#">FB_READ_057_SNVT_power_f [▶ 109]</a>     | Leistung (Watt)                                   |
| <a href="#">FB_READ_058_SNVT_ppm_f [▶ 110]</a>       | Konzentration (ppm)                               |
| <a href="#">FB_READ_059_SNVT_press_f [▶ 111]</a>     | Druck (Überdruck) (Pascal)                        |
| <a href="#">FB_READ_060_SNVT_res_f [▶ 111]</a>       | Elektrischer Widerstand (Ohm)                     |
| <a href="#">FB_READ_061_SNVT_sound_db_f [▶ 112]</a>  | Schalldruckpegel (dBspl)                          |
| <a href="#">FB_READ_062_SNVT_speed_f [▶ 113]</a>     | Lineare Geschwindigkeit (Meter / Sekunde)         |
| <a href="#">FB_READ_063_SNVT_temp_f [▶ 114]</a>      | Temperatur (Grad Celsius)                         |
| <a href="#">FB_READ_064_SNVT_time_f [▶ 114]</a>      | Abgelaufene Zeit (Sekunden)                       |
| <a href="#">FB_READ_065_SNVT_vol_f [▶ 115]</a>       | Volumen (Liter)                                   |
| <a href="#">FB_READ_066_SNVT_volt_f [▶ 116]</a>      | Elektrische Spannung (Volt)                       |
| <a href="#">FB_READ_067_SNVT_btu_f [▶ 117]</a>       | Thermische Energie (Btus)                         |
| <a href="#">FB_READ_068_SNVT_elec_whr_f [▶ 117]</a>  | Elektrische Energie (Watt / Stunden)              |
| <a href="#">FB_READ_069_SNVT_config_src [▶ 118]</a>  | Konfigurationseigenschaften                       |
| <a href="#">FB_READ_070_SNVT_color [▶ 119]</a>       | Farbe nach CIE Norm                               |
| <a href="#">FB_READ_071_SNVT_grammage [▶ 120]</a>    | amerikanisches Mass für Papiergewichte und Dichte |
| <a href="#">FB_READ_072_SNVT_grammage_f [▶ 120]</a>  | amerikanisches Mass für Papiergewichte und Dichte |
| <a href="#">FB_READ_073_SNVT_file_req [▶ 121]</a>    | Dateianforderung                                  |
| <a href="#">FB_READ_074_SNVT_file_status [▶ 122]</a> | Dateiinformaton                                   |
| <a href="#">FB_READ_075_SNVT_freq_f [▶ 123]</a>      | Frequenz (Hertz)                                  |
| <a href="#">FB_READ_076_SNVT_freq_hz [▶ 123]</a>     | Frequenz (Hertz)                                  |
| <a href="#">FB_READ_077_SNVT_freq_kilohz [▶ 124]</a> | Frequenz (Kilohertz)                              |
| <a href="#">FB_READ_078_SNVT_freq_milhz [▶ 125]</a>  | Frequenz (Millihertz)                             |
| <a href="#">FB_READ_079_SNVT_lux [▶ 126]</a>         | Beleuchtungsstärke (Lux)                          |

| Bausteine                            | Beschreibung                                     |
|--------------------------------------|--|
| FB_READ_081_SNVT_lev_percent [▶_126] | Prozentangabe                                    |
| FB_READ_082_SNVT_multiplier [▶_127]  | Multiplikator                                    |
| FB_READ_083_SNVT_state [▶_128]       | Statusinformation                                |
| FB_READ_084_SNVT_time_stamp [▶_129]  | Zeitstempel                                      |
| FB_READ_085_SNVT_zerospan [▶_130]    | Nullpunkt und Proportionalitätsfaktor            |
| FB_READ_086_SNVT_magcard [▶_130]     | ISO 7811 (40 Hexadezimal Zahlen)                 |
| FB_READ_087_SNVT_elapsed_tm [▶_131]  | Abgelaufene Zeit                                 |
| FB_READ_088_SNVT_alarm [▶_132]       | Alarm Status                                     |
| FB_READ_089_SNVT_currency [▶_133]    | Währung  |
| FB_READ_090_SNVT_file_pos [▶_134]    | Dateiposition                                    |
| FB_READ_091_SNVT_muldiv [▶_134]      | Verstärkungsfaktor/Dämpfungsfaktor               |
| FB_READ_092_SNVT_obj_request [▶_135] | Funktionsauswahl                                 |
| FB_READ_093_SNVT_obj_status [▶_136]  | Objektstatus                                     |
| FB_READ_094_SNVT_preset [▶_137]      | Voreinstellung                                   |
| FB_READ_095_SNVT_switch [▶_138]      | Schalter   |
| FB_READ_096_SNVT_trans_table [▶_138] | Übersetzungstabelle                              |
| FB_READ_097_SNVT_override [▶_139]    | Übersteuerungsmodus                              |
| FB_READ_098_SNVT_pwr_fact [▶_140]    | Leistungsfaktor                                  |
| FB_READ_099_SNVT_pwr_fact_f [▶_141]  | Leistungsfaktor                                  |
| FB_READ_100_SNVT_density [▶_142]     | Dichte (kg/m <sup>3</sup> )                      |
| FB_READ_101_SNVT_density_f [▶_142]   | Dichte (kg/m <sup>3</sup> )                      |
| FB_READ_102_SNVT_rpm [▶_143]         | Winkelgeschwindigkeit (Umdrehungen/Minute (RPM)) |
| FB_READ_103_SNVT_hvac_emerg [▶_144]  | HLK Notbetrieb                                   |
| FB_READ_104_SNVT_angle_deg [▶_145]   | Winkelangabe                                     |
| FB_READ_105_SNVT_temp_p [▶_145]      | Temperatur (Grad Celsius)                        |
| FB_READ_106_SNVT_temp_setpt [▶_146]  | Temperatur (6 Temperaturwerte)                   |
| FB_READ_107_SNVT_time_sec [▶_147]    | Abgelaufene Zeit (Sekunde)                       |
| FB_READ_108_SNVT_hvac_mode [▶_148]   | HLK Betriebsmodus                                |
| FB_READ_109_SNVT_occupancy [▶_148]   | Präsenzmeldung (Stati)                           |
| FB_READ_110_SNVT_area [▶_149]        | Fläche (Quadratmeter)                            |
| FB_READ_111_SNVT_hvac_overid [▶_150] | HLK Übersteuerungsmodus                          |
| FB_READ_112_SNVT_hvac_status [▶_151] | HLK Status                                       |
| FB_READ_113_SNVT_press_p [▶_152]     | Druck (Überdruck) (Pascal)                       |
| FB_READ_114_SNVT_address [▶_152]     | Neuron Adresse                                   |
| FB_READ_115_SNVT_scene [▶_153]       | Szenen   |
| FB_READ_116_SNVT_scene_cfg [▶_154]   | Szenen Einstellung                               |
| FB_READ_117_SNVT_setting [▶_155]     | Einstellungen                                    |
| FB_READ_118_SNVT_evap_state [▶_155]  | Verdampferstatus                                 |
| FB_READ_119_SNVT_therm_mode [▶_156]  | Thermostatmodus                                  |
| FB_READ_120_SNVT_defr_mode [▶_157]   | Abtaumodus                                       |
| FB_READ_121_SNVT_defr_term [▶_158]   | Abschluss des Abtauvorgangs                      |
| FB_READ_122_SNVT_defr_state [▶_159]  | Abtaustatus (Enumeration)                        |

| Bausteine                            | Beschreibung                                   |
|--------------------------------------|--|
| FB_READ_123_SNVT_time_min [▶ 159]    | Abgelaufen Zeit (Minuten)                      |
| FB_READ_124_SNVT_time_hour [▶ 160]   | Abgelaufene Zeit (Stunde)                      |
| FB_READ_125_SNVT_ph [▶ 161]          | Säuregehalt (pH)                               |
| FB_READ_126_SNVT_ph_f [▶ 162]        | Säuregehalt (pH)                               |
| FB_READ_127_SNVT_chlr_status [▶ 162] | Kälteerzeugerstatus                            |
| FB_READ_128_SNVT_tod_event [▶ 163]   | Präsenzzeit                                    |
| FB_READ_129_SNVT_smo_obscur [▶ 164]  | Verdunkelung durch Rauch                       |
| FB_READ_130_SNVT_fire_test [▶ 165]   | Anforderung Test Brandmeldeanlage              |
| FB_READ_131_SNVT_temp_ror [▶ 166]    | Wert der Temperaturänderung/-erhöhung          |
| FB_READ_132_SNVT_fire_init [▶ 166]   | Brandmeldertyp                                 |
| FB_READ_133_SNVT_fire_indcte [▶ 167] | Brandmeldeanzeige                              |
| FB_READ_134_SNVT_time_zone [▶ 168]   | Zeitzonebeschreibung                           |
| FB_READ_135_SNVT_earth_pos [▶ 169]   | Position auf der Erde                          |
| FB_READ_136_SNVT_reg_val [▶ 170]     | Registerwert                                   |
| FB_READ_137_SNVT_reg_val_ts [▶ 170]  | Registerwert                                   |
| FB_READ_138_SNVT_volt_ac [▶ 171]     | Wechselspannung (Volt AC)                      |
| FB_READ_139_SNVT_amp_ac [▶ 172]      | Wechselstrom                                   |
| FB_READ_143_SNVT_turbidity [▶ 173]   | Trübung  |
| FB_READ_144_SNVT_turbidity_f [▶ 174] | Trübung  |
| FB_READ_145_SNVT_hvac_type [▶ 174]   | HLK Anlagentyp                                 |
| FB_READ_146_SNVT_elec_kwh_l [▶ 175]  | Elektrische Energie (Kilowatt / Stunde)        |
| FB_READ_147_SNVT_temp_diff_p [▶ 176] | Temperaturdifferenz (Grad Celsius)             |
| FB_READ_148_SNVT_ctrl_req [▶ 177]    | Steuerungsanfrage                              |
| FB_READ_149_SNVT_ctrl_resp [▶ 178]   | Steuerungsantwort                              |
| FB_READ_150_SNVT_ptz [▶ 178]         | Kamera PTZ (SNZ)                               |
| FB_READ_151_SNVT_privacyzone [▶ 179] | Privatzone                                     |
| FB_READ_152_SNVT_pos_ctrl [▶ 180]    | Positionseinstellung für Kameras in Netzwerken |
| FB_READ_153_SNVT_enthalpy [▶ 181]    | Enthalpie (kiloJoules/kg)                      |
| FB_READ_154_SNVT_gfci_status [▶ 182] | Fehlerstromschutzschalter Status               |
| FB_READ_155_SNVT_motor_state [▶ 182] | Motorstatus (Motorstatus Benennung)            |
| FB_READ_156_SNVT_pumpset_mn [▶ 183]  | Pumpengruppe                                   |
| FB_READ_157_SNVT_ex_control [▶ 184]  | Exklusive Kontrolle                            |
| FB_READ_158_SNVT_pumpset_sn [▶ 185]  | Pumpengruppensor                               |
| FB_READ_159_SNVT_pump_sensor [▶ 186] | Pumpensor                                      |
| FB_READ_160_SNVT_abs_humid [▶ 186]   | Absolute Feuchtigkeit                          |
| FB_READ_161_SNVT_flow_p [▶ 187]      | Volumenstrom (Kubikmeter / Stunde)             |
| FB_READ_162_SNVT_dev_c_mode [▶ 188]  | Gerätebetriebsart                              |
| FB_READ_163_SNVT_valve_mode [▶ 189]  | Ventilzustand                                  |
| FB_READ_164_SNVT_alarm_2 [▶ 189]     | Alarmstatus 2                                  |
| FB_READ_165_SNVT_state_64 [▶ 190]    | Statusinformation (64 individuelle Bitwerte)   |
| FB_READ_166_SNVT_nv_type [▶ 191]     | Netzwerkvariablentyp                           |
| FB_READ_168_SNVT_ent_opmode [▶ 192]  | Betriebsart für Zutrittsobjekte                |
| FB_READ_169_SNVT_ent_state [▶ 193]   | Zustand für Zutrittsobjekte                    |



| Bausteine                               | Beschreibung  |
|---|---|
| FB_READ_170_SNVT_ent_status [▶ 193]     | Status von Zutrittsobjekten                           |
| FB_READ_171_SNVT_flow_dir [▶ 194]       | Flow direction  |
| FB_READ_172_SNVT_hvac_satsts [▶ 195]    | HLK Sättigungsstatus                                  |
| FB_READ_173_SNVT_dev_status [▶ 196]     | Gerätestatus  |
| FB_READ_174_SNVT_dev_fault [▶ 197]      | Fehlerstatus  |
| FB_READ_175_SNVT_dev_maint [▶ 197]      | Geräte Wartungsstatus                                 |
| FB_READ_176_SNVT_date_event [▶ 198]     | Status eines Ereignisses                              |
| FB_READ_177_SNVT_sched_val [▶ 199]      | Kalenderwert  |
| FB_READ_180_SNVT_sbld_state [▶ 200]     | Jalousie Status                                       |
| FB_READ_181_SNVT_rac_ctrl [▶ 201]       | Steuerung Soundfunktion                               |
| FB_READ_182_SNVT_rac_req [▶ 201]        | Anforderung Soundfunktion                             |
| FB_READ_183_SNVT_count_32 [▶ 202]       | Absoluter Zähler                                      |
| FB_READ_184_SNVT_clothes_w_c [▶ 203]    | Waschmaschine / Kommandos                             |
| FB_READ_185_SNVT_clothes_w_m [▶ 204]    | Waschmaschine / Management-Status                     |
| FB_READ_186_SNVT_clothes_w_s [▶ 205]    | Waschmaschine / Status                                |
| FB_READ_187_SNVT_clothes_w_a [▶ 205]    | Waschmaschine / Alarmmeldungen                        |
| FB_READ_188_SNVT_multiplier_s [▶ 206]   | Multiplikator   |
| FB_READ_189_SNVT_switch_2 [▶ 207]       | Schalter zum einstellen von Szenen und Einstellungen. |
| FB_READ_190_SNVT_color_2 [▶ 208]        | Farbe   |
| FB_READ_191_SNVT_log_status [▶ 209]     | Log Status  |
| FB_READ_192_SNVT_time_stamp_p [▶ 209]   | Präziser Zeitstempel                                  |
| FB_READ_193_SNVT_log_fx_request [▶ 210] | Log file transfer Anforderung                         |
| FB_READ_194_SNVT_log_fx_status [▶ 211]  | Log file transfer Status                              |
| FB_READ_195_SNVT_log_request [▶ 212]    | Log Status Anforderung                                |
| FB_READ_196_SNVT_enthalpy_d [▶ 213]     | Enthalpie Unterschied(kJ/kg)                          |
| FB_READ_197_SNVT_amp_ac_mil [▶ 213]     | Elektrischer Strom (Milliampere)                      |
| FB_READ_198_SNVT_time_hour_p [▶ 214]    | Zeit in Stunden                                       |
| FB_READ_199_SNVT_lamp_status [▶ 215]    | Lampen Status   |
| FB_READ_200_SNVT_environment [▶ 216]    | Umgebung  |
| FB_READ_201_SNVT_geo_loc [▶ 216]        | Geographische Lage                                    |

**Send**

| Bausteine                           | Beschreibung                               |
|-------------------------------------|--|
| FB_SEND_001_SNVT_amp [▶ 217]        | Elektrischer Strom (Ampere)                |
| FB_SEND_002_SNVT_amp_mil [▶ 219]    | Elektrischer Strom (Milliampere)           |
| FB_SEND_003_SNVT_angle [▶ 220]      | Winkelabstand (Bogenmaß)                   |
| FB_SEND_004_SNVT_angle_vel [▶ 221]  | Winkelgeschwindigkeit (Bogenmaß / Sekunde) |
| FB_SEND_005_SNVT_btu_kilo [▶ 223]   | Thermische Energie (Kilo-Btus)             |
| FB_SEND_006_SNVT_btu_mega [▶ 224]   | Thermische Energie (Mega-Btus)             |
| FB_SEND_007_SNVT_char_ascii [▶ 225] | ASCII Zeichen (8-bit ASCII character)      |
| FB_SEND_008_SNVT_count [▶ 227]      | Absolute Anzahl (Stück)                    |
| FB_SEND_009_SNVT_count_inc [▶ 228]  | Inkrement Zähler (Stück(delta))            |
| FB_SEND_011_SNVT_date_day [▶ 229]   | Wochentag                                  |

| <b>Bausteine</b>                                     | <b>Beschreibung</b>                                |
|--|--|
| <a href="#">FB_SEND_013_SNVT_elec_kwh [▶ 231]</a>    | Elektrische Energie (Kilowatt / Stunde)            |
| <a href="#">FB_SEND_014_SNVT_elec_whr [▶ 232]</a>    | Elektrische Energie (Watt / Stunde)                |
| <a href="#">FB_SEND_015_SNVT_flow [▶ 233]</a>        | Volumenstrom (Liter / Sekunde)                     |
| <a href="#">FB_SEND_016_SNVT_flow_mil [▶ 235]</a>    | Volumenstrom (Milliliter / Sekunde)                |
| <a href="#">FB_SEND_017_SNVT_length [▶ 236]</a>      | Länge (Meter)                                      |
| <a href="#">FB_SEND_018_SNVT_length_kilo [▶ 237]</a> | Länge (Kilometer)                                  |
| <a href="#">FB_SEND_019_SNVT_length_micr [▶ 239]</a> | Länge (Mikrometer)                                 |
| <a href="#">FB_SEND_020_SNVT_length_mil [▶ 240]</a>  | Länge (Millimeter)                                 |
| <a href="#">FB_SEND_021_SNVT_lev_cont [▶ 241]</a>    | Stufenlose Wertänderung (% vom obersten Niveau)    |
| <a href="#">FB_SEND_023_SNVT_mass [▶ 243]</a>        | Masse (Gramm)                                      |
| <a href="#">FB_SEND_024_SNVT_mass_kilo [▶ 244]</a>   | Masse (Kilogramm)                                  |
| <a href="#">FB_SEND_025_SNVT_mass_mega [▶ 245]</a>   | Masse (Metrische Tonne)                            |
| <a href="#">FB_SEND_026_SNVT_mass_mil [▶ 247]</a>    | Masse (Milligramm)                                 |
| <a href="#">FB_SEND_027_SNVT_power [▶ 248]</a>       | Leistung (Watt)                                    |
| <a href="#">FB_SEND_028_SNVT_power_kilo [▶ 249]</a>  | Leistung (Kilowatt)                                |
| <a href="#">FB_SEND_029_SNVT_ppm [▶ 251]</a>         | Konzentration (ppm)                                |
| <a href="#">FB_SEND_030_SNVT_press [▶ 252]</a>       | Druck (Überdruck) (Kilopascal)                     |
| <a href="#">FB_SEND_031_SNVT_res [▶ 253]</a>         | Elektrischer Widerstand (Ohm)                      |
| <a href="#">FB_SEND_032_SNVT_res_kilo [▶ 255]</a>    | Elektrischer Widerstand (Kilohm)                   |
| <a href="#">FB_SEND_033_SNVT_sound_db [▶ 256]</a>    | Schalldruckpegel (dB)                              |
| <a href="#">FB_SEND_034_SNVT_speed [▶ 257]</a>       | Lineare Geschwindigkeit (Meter / Sekunde)          |
| <a href="#">FB_SEND_035_SNVT_speed_mil [▶ 259]</a>   | Lineare Geschwindigkeit (Meter / Sekunde)          |
| <a href="#">FB_SEND_036_SNVT_str_asc [▶ 260]</a>     | Zeichenkette (30 Zeichen max) (ASCII-Zeichenkette) |
| <a href="#">FB_SEND_037_SNVT_str_int [▶ 261]</a>     | Wide Character String                              |
| <a href="#">FB_SEND_038_SNVT_telcom [▶ 262]</a>      | Telefonstatus (Telefonstatus Namen)                |
| <a href="#">FB_SEND_039_SNVT_temp [▶ 264]</a>        | Temperatur (Grad Celsius)                          |
| <a href="#">FB_SEND_041_SNVT_vol [▶ 265]</a>         | Volumen (Liter)                                    |
| <a href="#">FB_SEND_042_SNVT_vol_kilo [▶ 266]</a>    | Volumen (Kiloliter)                                |
| <a href="#">FB_SEND_043_SNVT_vol_mil [▶ 268]</a>     | Volumen (Milliliters)                              |
| <a href="#">FB_SEND_044_SNVT_volt [▶ 269]</a>        | Elektrische Spannung (Volt)                        |
| <a href="#">FB_SEND_045_SNVT_volt_dbmv [▶ 270]</a>   | Elektrische Spannung (dB Mikrovolt)                |
| <a href="#">FB_SEND_046_SNVT_volt_kilo [▶ 272]</a>   | Elektrische Spannung (Kilovolt)                    |
| <a href="#">FB_SEND_047_SNVT_volt_mil [▶ 273]</a>    | Elektrische Spannung (Millivolt)                   |
| <a href="#">FB_SEND_048_SNVT_amp_f [▶ 274]</a>       | Elektrischer Strom (Ampere)                        |
| <a href="#">FB_SEND_049_SNVT_angle_f [▶ 276]</a>     | Winkelabstand (Bogenmaß)                           |
| <a href="#">FB_SEND_050_SNVT_angle_vel_f [▶ 277]</a> | Winkelgeschwindigkeit (Radiant / Sekunde)          |
| <a href="#">FB_SEND_051_SNVT_count_f [▶ 278]</a>     | Absolute Anzahl (Stück)                            |
| <a href="#">FB_SEND_052_SNVT_count_inc_f [▶ 280]</a> | Inkrement Zähler (Stück(delta))                    |
| <a href="#">FB_SEND_053_SNVT_flow_f [▶ 281]</a>      | Volumenstrom (Liter / Sekunde)                     |
| <a href="#">FB_SEND_054_SNVT_length_f [▶ 282]</a>    | Länge (Meter)                                      |
| <a href="#">FB_SEND_055_SNVT_lev_cont_f [▶ 284]</a>  | Stufenlose Wertänderung (% vom obersten Niveau)    |
| <a href="#">FB_SEND_056_SNVT_mass_f [▶ 285]</a>      | Masse (Gramm)                                      |
| <a href="#">FB_SEND_057_SNVT_power_f [▶ 286]</a>     | Leistung (Watt)                                    |



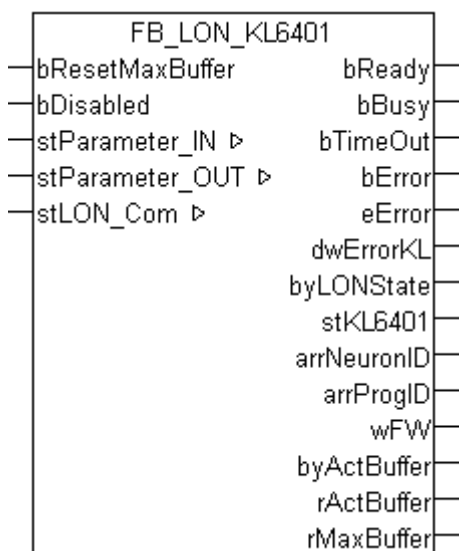
| <b>Bausteine</b>                     | <b>Beschreibung</b>                               |
|--------------------------------------|---|
| FB_SEND_058_SNVT_ppm_f [▶ 288]       | Konzentration (ppm)                               |
| FB_SEND_059_SNVT_press_f [▶ 289]     | Druck (Überdruck) (Pascal)                        |
| FB_SEND_060_SNVT_res_f [▶ 290]       | Elektrischer Widerstand (Ohm)                     |
| FB_SEND_061_SNVT_sound_db_f [▶ 292]  | Schalldruckpegel (dBspl)                          |
| FB_SEND_062_SNVT_speed_f [▶ 293]     | Lineare Geschwindigkeit (Meter / Sekunde)         |
| FB_SEND_063_SNVT_temp_f [▶ 294]      | Temperatur (Grad Celsius)                         |
| FB_SEND_064_SNVT_time_f [▶ 296]      | Abgelaufene Zeit (Sekunden)                       |
| FB_SEND_065_SNVT_vol_f [▶ 297]       | Volumen (Liter)                                   |
| FB_SEND_066_SNVT_volt_f [▶ 298]      | Elektrische Spannung (Volt)                       |
| FB_SEND_067_SNVT_btu_f [▶ 300]       | Thermische Energie (Btus)                         |
| FB_SEND_068_SNVT_elec_whr_f [▶ 301]  | Elektrische Energie (Watt / Stunden)              |
| FB_SEND_069_SNVT_config_src [▶ 302]  | Konfigurationseigenschaften                       |
| FB_SEND_070_SNVT_color [▶ 304]       | Farbe nach CIE Norm                               |
| FB_SEND_071_SNVT_grammage [▶ 305]    | amerikanisches Mass für Papiergewichte und Dichte |
| FB_SEND_072_SNVT_grammage_f [▶ 306]  | amerikanisches Mass für Papiergewichte und Dichte |
| FB_SEND_073_SNVT_file_req [▶ 307]    | Dateianforderung                                  |
| FB_SEND_074_SNVT_file_status [▶ 309] | Dateiinformatio                                   |
| FB_SEND_075_SNVT_freq_f [▶ 310]      | Frequenz (Hertz)                                  |
| FB_SEND_076_SNVT_freq_hz [▶ 311]     | Frequenz (Hertz)                                  |
| FB_SEND_077_SNVT_freq_kilohz [▶ 313] | Frequenz (Kilohertz)                              |
| FB_SEND_078_SNVT_freq_milhz [▶ 314]  | Frequenz (Millihertz)                             |
| FB_SEND_079_SNVT_lux [▶ 315]         | Beleuchtungsstärke (Lux)                          |
| FB_SEND_081_SNVT_lev_percent [▶ 317] | Prozentangabe                                     |
| FB_SEND_082_SNVT_multiplier [▶ 318]  | Multiplikator                                     |
| FB_SEND_083_SNVT_state [▶ 319]       | Statusinformation                                 |
| FB_SEND_084_SNVT_time_stamp [▶ 321]  | Zeitstempel                                       |
| FB_SEND_085_SNVT_zerospan [▶ 322]    | Nullpunkt und Proportionalitätsfaktor             |
| FB_SEND_086_SNVT_magcard [▶ 323]     | ISO 7811 (40 Hexadezimal Zahlen)                  |
| FB_SEND_087_SNVT_elapsed_tm [▶ 324]  | Abgelaufene Zeit                                  |
| FB_SEND_088_SNVT_alarm [▶ 325]       | Alarm Status                                      |
| FB_SEND_089_SNVT_currency [▶ 327]    | Währung   |
| FB_SEND_090_SNVT_file_pos [▶ 328]    | Dateiposition                                     |
| FB_SEND_091_SNVT_muldiv [▶ 329]      | Verstärkungsfaktor/Dämpfungsfaktor                |
| FB_SEND_092_SNVT_obj_request [▶ 330] | Funktionsauswahl                                  |
| FB_SEND_093_SNVT_obj_status [▶ 331]  | Objektstatus                                      |
| FB_SEND_094_SNVT_preset [▶ 333]      | Voreinstellung                                    |
| FB_SEND_095_SNVT_switch [▶ 334]      | Schalter  |
| FB_SEND_096_SNVT_trans_table [▶ 335] | Übersetzungstabelle                               |
| FB_SEND_097_SNVT_override [▶ 336]    | Übersteuerungsmodus                               |
| FB_SEND_098_SNVT_pwr_fact [▶ 337]    | Leistungsfaktor                                   |
| FB_SEND_099_SNVT_pwr_fact_f [▶ 339]  | Leistungsfaktor                                   |
| FB_SEND_100_SNVT_density [▶ 340]     | Dichte (kg/m <sup>3</sup> )                       |
| FB_SEND_101_SNVT_density_f [▶ 341]   | Dichte (kg/m <sup>3</sup> )                       |

| Bausteine                            | Beschreibung                                     |
|--------------------------------------|--|
| FB_SEND_102_SNVT_rpm [▶ 343]         | Winkelgeschwindigkeit (Umdrehungen/Minute (RPM)) |
| FB_SEND_103_SNVT_hvac_emerg [▶ 344]  | HLK Notbetrieb                                   |
| FB_SEND_104_SNVT_angle_deg [▶ 345]   | Winkelangabe                                     |
| FB_SEND_105_SNVT_temp_p [▶ 347]      | Temperatur (Grad Celsius)                        |
| FB_SEND_106_SNVT_temp_setpt [▶ 348]  | Temperatur (6 Temperaturwerte)                   |
| FB_SEND_107_SNVT_time_sec [▶ 349]    | Abgelaufene Zeit (Sekunde)                       |
| FB_SEND_108_SNVT_hvac_mode [▶ 350]   | HLK Betriebsmodus                                |
| FB_SEND_109_SNVT_occupancy [▶ 352]   | Präsenzmeldung (Stati)                           |
| FB_SEND_110_SNVT_area [▶ 353]        | Fläche (Quadratmeter)                            |
| FB_SEND_111_SNVT_hvac_overid [▶ 354] | HLK Übersteuerungsmodus                          |
| FB_SEND_112_SNVT_hvac_status [▶ 355] | HLK Status                                       |
| FB_SEND_113_SNVT_press_p [▶ 357]     | Druck (Überdruck) (Pascal)                       |
| FB_SEND_114_SNVT_address [▶ 358]     | Neuron Adresse (16-bit Adress Wert)              |
| FB_SEND_115_SNVT_scene [▶ 359]       | Szenen   |
| FB_SEND_116_SNVT_scene_cfg [▶ 361]   | Szenen Einstellung                               |
| FB_SEND_117_SNVT_setting [▶ 362]     | Einstellungen                                    |
| FB_SEND_118_SNVT_evap_state [▶ 363]  | Verdampferstatus                                 |
| FB_SEND_119_SNVT_therm_mode [▶ 364]  | Thermostatmodus                                  |
| FB_SEND_120_SNVT_defr_mode [▶ 365]   | Abtaumodus                                       |
| FB_SEND_121_SNVT_defr_term [▶ 367]   | Abschluss des Abtauvorgangs                      |
| FB_SEND_122_SNVT_defr_state [▶ 368]  | Abtaustatus (Enumeration)                        |
| FB_SEND_123_SNVT_time_min [▶ 369]    | Abgelaufene Zeit (Minuten)                       |
| FB_SEND_124_SNVT_time_hour [▶ 370]   | Abgelaufen Zeit (Stunde)                         |
| FB_SEND_125_SNVT_ph [▶ 372]          | Säuregehalt (pH)                                 |
| FB_SEND_126_SNVT_ph_f [▶ 373]        | Säuregehalt (pH)                                 |
| FB_SEND_127_SNVT_chlr_status [▶ 374] | Kälteerzeugerstatus                              |
| FB_SEND_128_SNVT_tod_event [▶ 376]   | Präsenzzeit                                      |
| FB_SEND_129_SNVT_smo_obscur [▶ 377]  | Verdunkelung durch Rauch                         |
| FB_SEND_130_SNVT_fire_test [▶ 378]   | Anforderung Test Brandmeldeanlage                |
| FB_SEND_131_SNVT_temp_ror [▶ 379]    | Wert der Temperaturänderung/-erhöhung            |
| FB_SEND_132_SNVT_fire_init [▶ 381]   | Brandmeldertyp                                   |
| FB_SEND_133_SNVT_fire_indcte [▶ 382] | Brandmeldeanzeige                                |
| FB_SEND_134_SNVT_time_zone [▶ 383]   | Zeitzonebeschreibung                             |
| FB_SEND_135_SNVT_earth_pos [▶ 384]   | Position auf der Erde                            |
| FB_SEND_136_SNVT_reg_val [▶ 386]     | Registerwert                                     |
| FB_SEND_137_SNVT_reg_val_ts [▶ 387]  | Registerwert                                     |
| FB_SEND_138_SNVT_volt_ac [▶ 388]     | Wechselspannung                                  |
| FB_SEND_139_SNVT_amp_ac [▶ 389]      | Wechselstrom                                     |
| FB_SEND_143_SNVT_turbidity [▶ 391]   | Trübung  |
| FB_SEND_144_SNVT_turbidity_f [▶ 392] | Trübung  |
| FB_SEND_145_SNVT_hvac_type [▶ 393]   | HLK Anlagentyp                                   |
| FB_SEND_146_SNVT_elec_kwh_l [▶ 395]  | Elektrische Energie (Kilowatt / Stunde)          |

| <b>Bausteine</b>                      | <b>Beschreibung</b>                                   |
|---------------------------------------|---|
| FB_SEND 147 SNVT temp diff p [▶ 396]  | Temperaturdifferenz (Grad Celsius)                    |
| FB_SEND 148 SNVT ctrl req [▶ 397]     | Steuerungsanfrage                                     |
| FB_SEND 149 SNVT ctrl resp [▶ 399]    | Steuerungsantwort                                     |
| FB_SEND 150 SNVT ptz [▶ 400]          | Kamera PTZ (SNZ)                                      |
| FB_SEND 151 SNVT privacyzone [▶ 401]  | Privatzone  |
| FB_SEND 152 SNVT pos ctrl [▶ 402]     | Positionseinstellung für Kameras in Netzwerken        |
| FB_SEND 153 SNVT enthalpy [▶ 404]     | Enthalpie (kiloJoules/kg)                             |
| FB_SEND 154 SNVT gfci status [▶ 405]  | Fehlerstromschutzschalter Status                      |
| FB_SEND 155 SNVT motor state [▶ 406]  | Motorstatus (Motorstatus Benennung)                   |
| FB_SEND 156 SNVT pumpset mn [▶ 407]   | Pumpengruppe  |
| FB_SEND 157 SNVT ex control [▶ 409]   | Exklusive Kontrolle                                   |
| FB_SEND 158 SNVT pumpset sn [▶ 410]   | Pumpengruppensensor                                   |
| FB_SEND 159 SNVT pump sensor [▶ 411]  | Pumpensensor (Geschwindigkeit, Temperatur, Status)    |
| FB_SEND 160 SNVT abs humid [▶ 412]    | Absolute Feuchtigkeit                                 |
| FB_SEND 161 SNVT flow p [▶ 414]       | Volumenstrom (Kubikmeter / Stunde)                    |
| FB_SEND 162 SNVT dev c mode [▶ 415]   | Gerätebetriebsart                                     |
| FB_SEND 163 SNVT valve mode [▶ 416]   | Ventilzustand   |
| FB_SEND 164 SNVT alarm 2 [▶ 417]      | Alarmstatus 2   |
| FB_SEND 165 SNVT state 64 [▶ 419]     | Statusinformation (64 individuelle Bitwerte)          |
| FB_SEND 166 SNVT nv_type [▶ 420]      | Netzwerkvariablentyp                                  |
| FB_SEND 168 SNVT ent_opmode [▶ 421]   | Betriebsart für Zutrittsobjekte                       |
| FB_SEND 169 SNVT ent_state [▶ 422]    | Zustand für Zutrittsobjekte                           |
| FB_SEND 170 SNVT ent_status [▶ 423]   | Status von Zutrittsobjekten                           |
| FB_SEND 171 SNVT flow_dir [▶ 425]     | Gangrichtung  |
| FB_SEND 172 SNVT hvac satsts [▶ 426]  | HLK Sättigungsstatus                                  |
| FB_SEND 173 SNVT dev_status [▶ 427]   | Gerätestatus  |
| FB_SEND 174 SNVT dev_fault [▶ 428]    | Fehlerstatus  |
| FB_SEND 175 SNVT dev_maint [▶ 430]    | Geräte Wartungsstatus                                 |
| FB_SEND 176 SNVT date_event [▶ 431]   | Status eines Ereignisses                              |
| FB_SEND 177 SNVT sched_val [▶ 432]    | Kalenderwert  |
| FB_SEND 180 SNVT sbld state [▶ 433]   | Jalousie Status                                       |
| FB_SEND 181 SNVT rac_ctrl [▶ 435]     | Steuerung Soundfunktion                               |
| FB_SEND 182 SNVT rac_req [▶ 436]      | Anforderung Soundfunktion                             |
| FB_SEND 183 SNVT count 32 [▶ 437]     | Absoluter Zähler                                      |
| FB_SEND 184 SNVT clothes_w_c [▶ 438]  | Waschmaschine / Kommandos                             |
| FB_SEND 185 SNVT clothes_w_m [▶ 440]  | Waschmaschine /Management-Status.                     |
| FB_SEND 186 SNVT clothes_w_s [▶ 441]  | Waschmaschine / Status                                |
| FB_SEND 187 SNVT clothes_w_a [▶ 442]  | Waschmaschine / Status                                |
| FB_SEND 188 SNVT multiplier_s [▶ 443] | Multiplikator   |
| FB_SEND 189 SNVT switch_2 [▶ 445]     | Schalter zum einstellen von Szenen und Einstellungen. |
| FB_SEND 190 SNVT color_2 [▶ 446]      | Farbe   |
| FB_SEND 191 SNVT log_status [▶ 447]   | Log Status  |

| Bausteine                               | Beschreibung                     |
|---|----------------------------------|
| FB_SEND_192_SNVT_time_stamp_p [▶ 448]   | Präziser Zeitstempel             |
| FB_SEND_193_SNVT_log_fx_request [▶ 450] | Log file transfer Anforderung    |
| FB_SEND_194_SNVT_log_fx_status [▶ 451]  | Log file transfer Status.        |
| FB_SEND_195_SNVT_log_request [▶ 452]    | Log Status Anforderung           |
| FB_SEND_196_SNVT_enthalpy_d [▶ 453]     | Enthalpie Unterschied(kJ/kg)     |
| FB_SEND_197_SNVT_amp_ac_mil [▶ 455]     | Elektrischer Strom (Milliampere) |
| FB_SEND_198_SNVT_time_hour_p [▶ 456]    | Zeit in Stunden                  |
| FB_SEND_199_SNVT_lamp_status [▶ 457]    | Lampen Status                    |
| FB_SEND_200_SNVT_environment [▶ 459]    | Umgebung                         |
| FB_SEND_201_SNVT_geo_loc [▶ 460]        | Geographische Lage               |

### 7.2.1 FB\_LON\_KL6401



Dieser Funktionsbaustein dient zum Senden/ Empfangen von LON SNVT's über die Busklemme KL6401. Für jede Klemme wird eine Instanz dieses Bausteins benötigt. Mit einer Instanz können maximal 62 SNVT's gesendet oder empfangen werden.

Über die VAR\_IN\_OUT-Struktur stLON\_Com [▶ 562] muss der FB an die Sende/Empfangsbausteine gebunden werden.



#### Einschränkungen

- Nur ein Aufruf pro Instanz
- Aufruf muss einmal pro PLC-Zyklus erfolgen
- Instanz muss in derselben PLC-Task aufgerufen werden, wie die ihm zugeordneten Sende- und Empfangsbausteine

#### VAR\_INPUT

```
bResetMaxBuffer : BOOL;
bDisabled       : BOOL := FALSE;
```

**bResetMaxBuffer:** Löscht den Wert *rMaxBuffer* für die maximale Auslastung des Sendepuffers.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bReady      : BOOL;
bBusy       : BOOL;
bTimeOut    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
dwErrorKL   : DWORD;
byLONState  : BYTE;
stKL6401    : ST_KL6401;
arrNeuronID : ARRAY [0..5] OF BYTE;
arrProgID   : ARRAY [0..7] OF BYTE;
wFW         : WORD;
byActBuffer : BYTE;
rActBuffer  : REAL;
rMaxBuffer  : REAL;

```

**bReady:** Initialisierung ist abgeschlossen.

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bTimeOut:** Zeitüberschreitung während der Initialisierung.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [\[► 613\]](#) des Bausteins. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**byLONState:** LON Status.

**stKL6401:** Struktur für die Konfiguration (für zukünftige Anwendungen) (siehe [ST\\_KL6401 \[► 563\]](#)).

**arrNeuronID:** Neuron ID der Klemme (wird erst ab Firmware 4C unterstützt).

**arrProgID:** Programm ID der Klemme (wird erst ab Firmware 4C unterstützt).

**wFW:** Firmware der Klemme (wird erst ab Firmware 4C unterstützt).

**byActBuffer:** Anzahl der Aufträge im Sendepuffer.

**rActBuffer:** Aktuelle Auslastung des Sendepuffers in Prozent.

**rMaxBuffer:** Maximale Auslastung des Sendepuffers in Prozent. Der Wert kann mit der Eingangsvariable *bResetMaxBuffer* gelöscht werden.

**VAR\_IN\_OUT**

```

stParameter_IN   : ST_LON_Parameter_IN_36B;
stParameter_OUT  : ST_LON_Parameter_OUT_36B;
stLON_Com        : ST_LON_Communication;

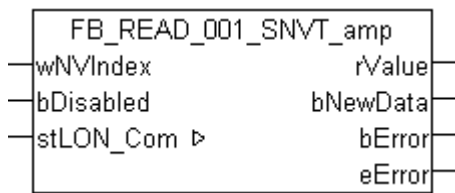
```

**stParameter\_IN:** Eingangsvariable der Hardware (siehe [ST\\_LON\\_Parameter\\_IN\\_36B \[► 560\]](#)).

**stParameter\_OUT:** Ausgangsvariable der Hardware (siehe [ST\\_LON\\_Parameter\\_OUT\\_36B \[► 561\]](#)).

**stLON\_Com:** Über diese Struktur wird der Baustein mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)). Das eigentliche Senden und Empfangen wird in diesem Baustein realisiert. Die Sende-/Empfangsbausteine übernehmen nur die Vorbereitung/Auswertung der Daten.

## 7.2.2 FB\_READ\_001\_SNVT\_amp



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_amp.

**SNVT Nummer:** 001.

**Beschreibung:** Elektrischer Strom (Ampere).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

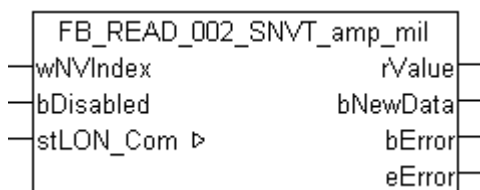
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.3 FB\_READ\_002\_SNVT\_amp\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_amp\_mil.

**SNVT Nummer:** 002.

**Beschreibung:** Elektrischer Strom (Milliampere).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser Index wird für das Binding der LON Knoten benötigt. Pro Klemme (KL6401) sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

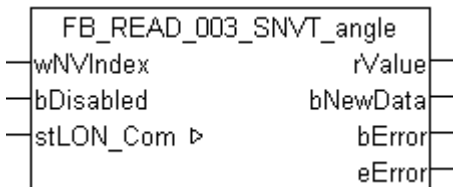
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.4 FB\_READ\_003\_SNVT\_angle**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_angle.

**SNVT Nummer:** 003.

**Beschreibung:** Winkelabstand (Bogenmaß).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```



**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

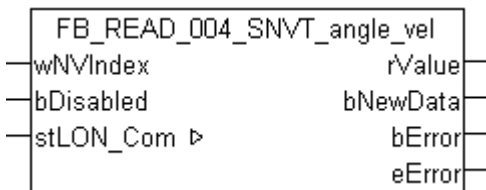
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.5 FB\_READ\_004\_SNVT\_angle\_vel**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_angle\_vel.

**SNVT Nummer:** 004.

**Beschreibung:** Winkelgeschwindigkeit (Bogenmaß / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

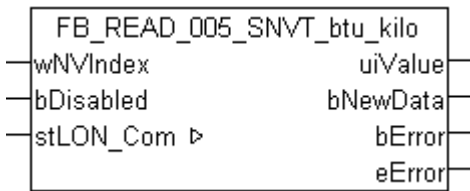
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).



### 7.2.6 FB\_READ\_005\_SNVT\_btu\_kilo



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_btu\_kilo.

**SNVT Nummer:** 005.

**Beschreibung:** Thermische Energie (Kilo BTU).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

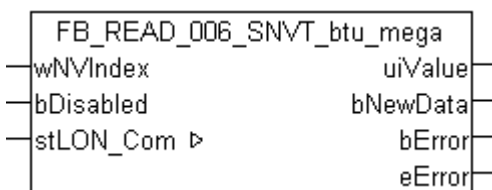
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

### 7.2.7 FB\_READ\_006\_SNVT\_btu\_mega



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_btu\_mega.

**SNVT Nummer:** 006.

**Beschreibung:** ASCII Zeichen (8-bit ASCII character).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

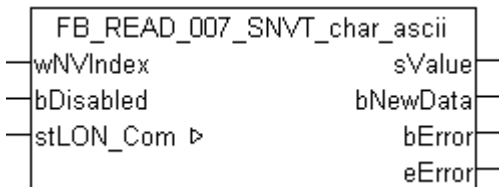
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.8 FB\_READ\_007\_SNVT\_char\_ascii**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_char\_ascii.

**SNVT Nummer:** 007.

**Beschreibung:** ASCII Zeichen (8-bit ASCII character).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
sValue        : STRING(1);
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**sValue:** STRING(1).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

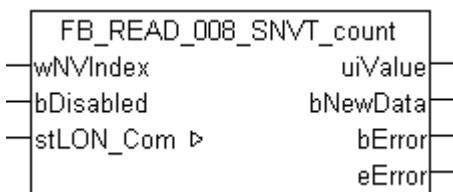
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.9 FB\_READ\_008\_SNVT\_count**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_count.

**SNVT Nummer:** 008.

**Beschreibung:** Zähler (nur positive Werte).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

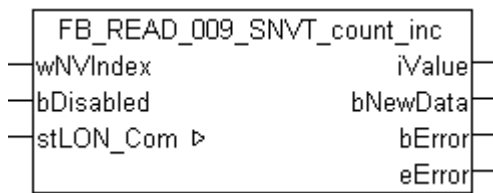
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.10 FB\_READ\_009\_SNVT\_count\_inc



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_count\_inc.

**SNVT Nummer:** 009.

**Beschreibung:** Zähler (negative und positive Werte).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
iValue       : INT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**iValue:** Min: -32768 / Max: 32767.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

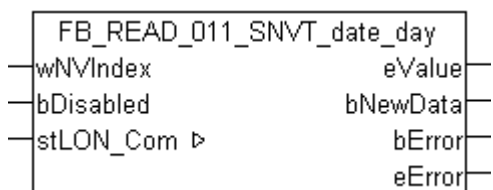
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.11 FB\_READ\_011\_SNVT\_date\_day



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_date\_day.

**SNVT Nummer:** 011.

**Beschreibung:** Wochentag.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue       : E_LON_days_of_week_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll, (siehe [E\\_LON\\_days\\_of\\_week\\_t](#) [▶ 502]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

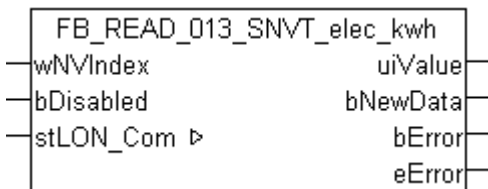
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.12 FB\_READ\_013\_SNVT\_elec\_kwh**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_elec\_kwh.

**SNVT Nummer:** 013.

**Beschreibung:** Elektrische Energie (kW/h).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue      : UINT;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

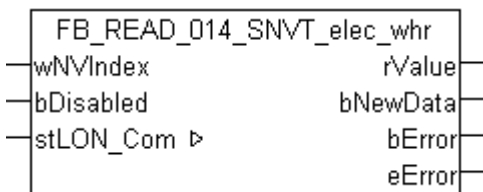
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.13 FB\_READ\_014\_SNVT\_elec\_whr**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_elec\_whr.

**SNVT Nummer:** 014.

**Beschreibung:** Elektrische Energie (W/h).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

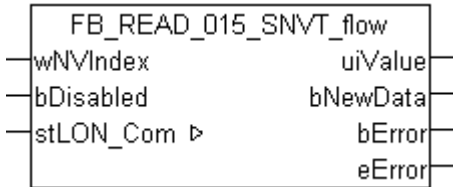
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.14 FB\_READ\_015\_SNVT\_flow**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_flow.

**SNVT Nummer:** 015.

**Beschreibung:** Volumenstrom (Liter / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

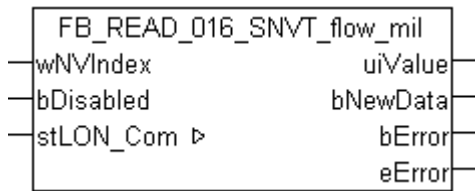
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).



## 7.2.15 FB\_READ\_016\_SNVT\_flow\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_flow\_mil.

**SNVT Nummer:** 016.

**Beschreibung:** Volumenstrom (Milliliter / Sekunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

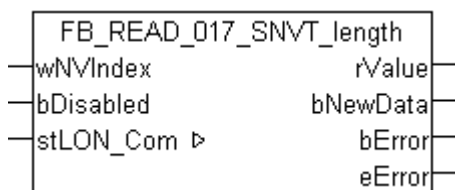
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.16 FB\_READ\_017\_SNVT\_length



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_length.

**SNVT Nummer:** 017.

**Beschreibung:** Länge (Meter).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

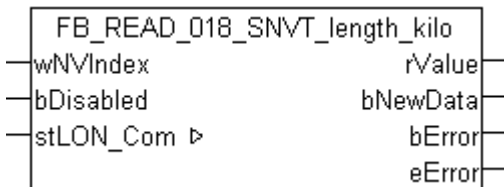
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.17 FB\_READ\_018\_SNVT\_length\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_length\_kilo.

**SNVT Nummer:** 018.

**Beschreibung:** Länge (Kilometer).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

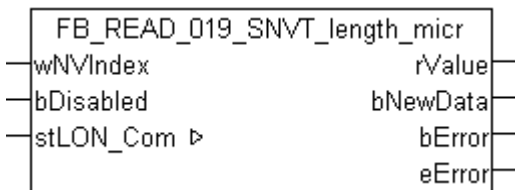
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.18 FB\_READ\_019\_SNVT\_length\_micr**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_length\_micr.

**SNVT Nummer:** 019.

**Beschreibung:** Länge (Mikrometer).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

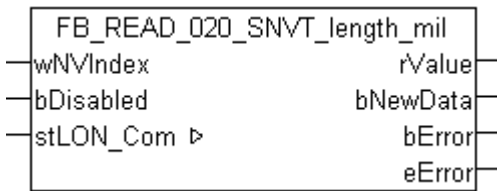
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.19 FB\_READ\_020\_SNVT\_length\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_length\_mil.

**SNVT Nummer:** 020.

**Beschreibung:** Länge (Millimeter).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

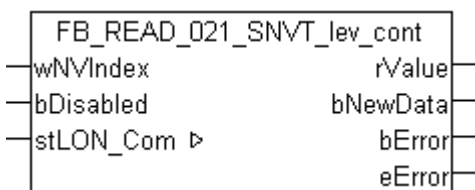
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.20 FB\_READ\_021\_SNVT\_lev\_cont



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_lev\_cont.

**SNVT Nummer:** 021.

**Beschreibung:** Stufenlose Wertänderung (% vom obersten Niveau).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 100.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

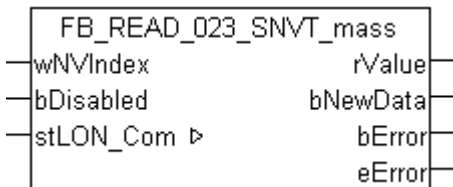
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.21 FB\_READ\_023\_SNVT\_mass**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_mass.

**SNVT Nummer:** 023.

**Beschreibung:** Masse (Gramm).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

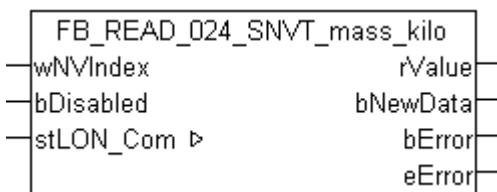
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.22 FB\_READ\_024\_SNVT\_mass\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_mass\_kilo.

**SNVT Nummer:** 024.

**Beschreibung:** Masse (Kilogramm).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

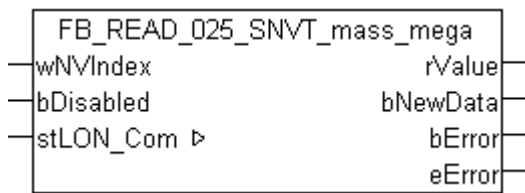
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.23 FB\_READ\_025\_SNVT\_mass\_mega



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_mass\_mega.

**SNVT Nummer:** 025.

**Beschreibung:** Masse (Metrische Tonne).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

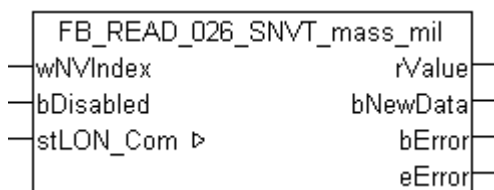
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.24 FB\_READ\_026\_SNVT\_mass\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_mass\_mil.

**SNVT Nummer:** 026.



**Beschreibung:** Masse (Milligramm).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

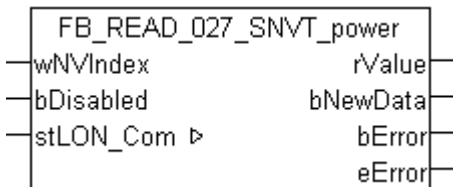
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.25 FB\_READ\_027\_SNVT\_power**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_power.

**SNVT Nummer:** 027.

**Beschreibung:** Leistung (Watt).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

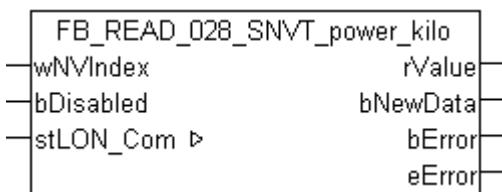
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.26 FB\_READ\_028\_SNVT\_power\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_power\_kilo.

**SNVT Nummer:** 028.

**Beschreibung:** Leistung (Kilowatt).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

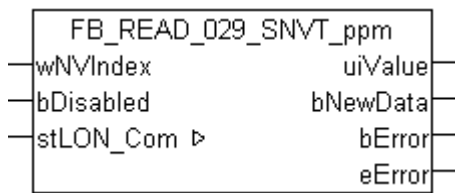
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.27 FB\_READ\_029\_SNVT\_ppm



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ppm.

**SNVT Nummer:** 029.

**Beschreibung:** Konzentration (ppm).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

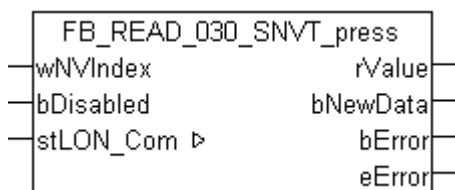
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.28 FB\_READ\_030\_SNVT\_press



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_press.

**SNVT Nummer:** 030.

**Beschreibung:** Druck (Überdruck) (Kilopascal).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

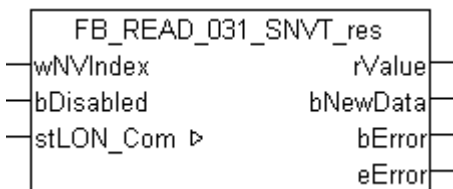
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Send-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.29 FB\_READ\_031\_SNVT\_res**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_res

**SNVT Nummer:** 031

**Beschreibung:** Elektrischer Widerstand (Ohm)

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

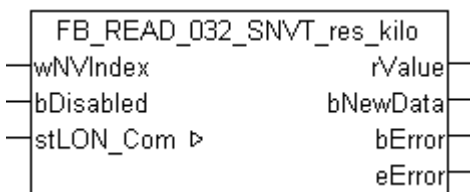
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.30 FB\_READ\_032\_SNVT\_res\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_res\_kilo.

**SNVT Nummer:** 032.

**Beschreibung:** Elektrischer Widerstand (Kiloohm).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

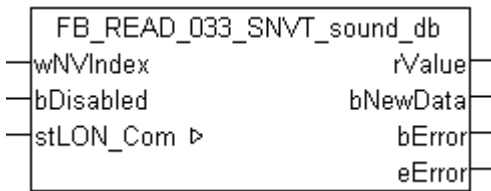
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.31 FB\_READ\_033\_SNVT\_sound\_db



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_sound\_db.

**SNVT Nummer:** 033.

**Beschreibung:** Schalldruckpegel (dB).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.67.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

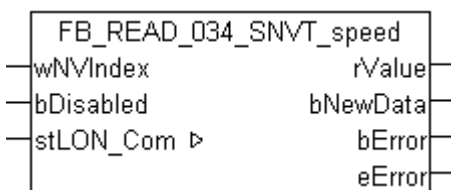
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.32 FB\_READ\_034\_SNVT\_speed



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_speed.

**SNVT Nummer:** 034.

**Beschreibung:** Lineare Geschwindigkeit (Meter / Sekunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

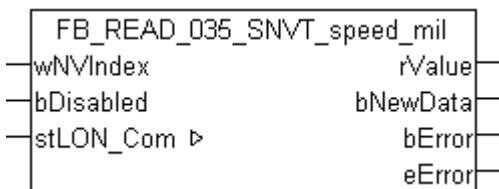
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

### 7.2.33 FB\_READ\_035\_SNVT\_speed\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_speed\_mil.

**SNVT Nummer:** 035.

**Beschreibung:** Lineare Geschwindigkeit (Millimeter / Sekunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```



**rValue:** Min: 0 / Max: 65.535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

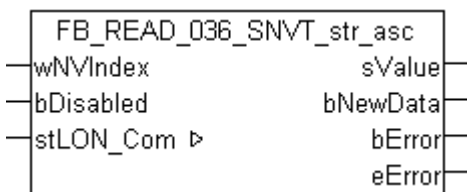
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gle

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

**7.2.34 FB\_READ\_036\_SNVT\_str\_asc**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_str\_asc.

**SNVT Nummer:** 036.

**Beschreibung:** Zeichenkette (30 Zeichen max) (ASCII-Zeichenkette).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
sValue : STRING(31);
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**sValue:** STRING(31).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

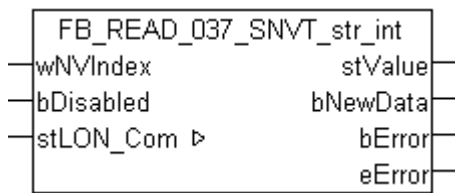
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

### 7.2.35 FB\_READ\_037\_SNVT\_str\_int



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_str\_int.

**SNVT Nummer:** 037.

**Beschreibung:** Wide Character String mit eigenem Code (Max. 15 Zeichen) (Wide character string).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_str_int;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur der Daten, die empfangen wurden (siehe [ST\\_LON\\_SNVT\\_str\\_int \[▶ 587\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

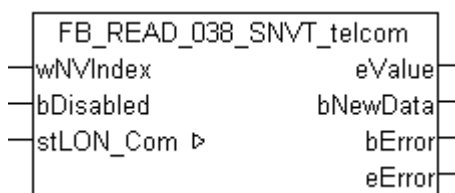
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.36 FB\_READ\_038\_SNVT\_telcom



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_telcom.

**SNVT Nummer:** 038.

**Beschreibung:** Telefonstatus (Telefonstatus Namen).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue        : E_LON_telcom_states_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_telcom\\_states t \[▶ 533\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

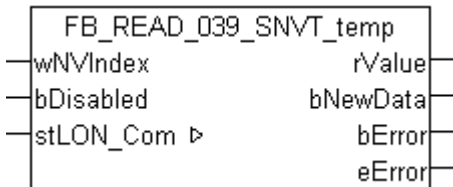
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.37 FB\_READ\_039\_SNVT\_temp



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp.

**SNVT Nummer:** 039.

**Beschreibung:** Temperatur (Grad Celsius).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

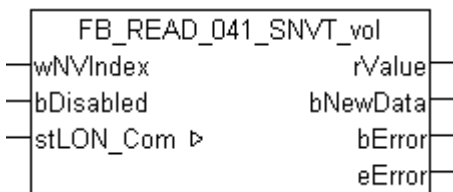
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.38 FB\_READ\_041\_SNVT\_vol**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_vol.

**SNVT Nummer:** 041.

**Beschreibung:** Volumen (Liter).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

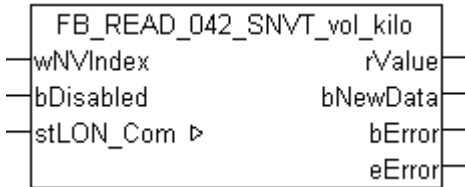
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.39 FB\_READ\_042\_SNVT\_vol\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_vol\_kilo.

**SNVT Nummer:** 042.

**Beschreibung:** Volumen (Kiloliter).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

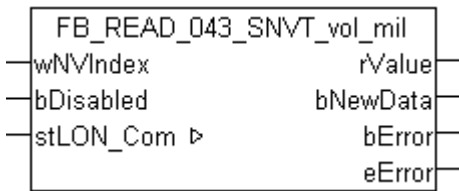
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

### 7.2.40 FB\_READ\_043\_SNVT\_vol\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_vol\_mil.

**SNVT Nummer:** 043.

**Beschreibung:** Volumen (Milliliter).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue        : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

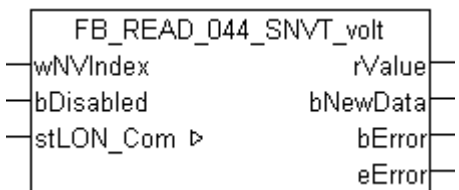
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

### 7.2.41 FB\_READ\_044\_SNVT\_volt



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt.

**SNVT Nummer:** 044.

**Beschreibung:** Elektrische Spannung (Volt).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

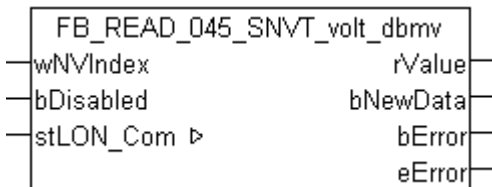
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.42 FB\_READ\_045\_SNVT\_volt\_dbmv**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt\_dbmv.

**SNVT Nummer:** 045.

**Beschreibung:** Elektrische Spannung (dB Mikrovolt).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.67.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

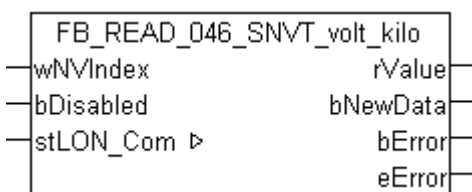
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.43 FB\_READ\_046\_SNVT\_volt\_kilo**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt\_kilo.

**SNVT Nummer:** 046.

**Beschreibung:** Elektrische Spannung (Kilovolt).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

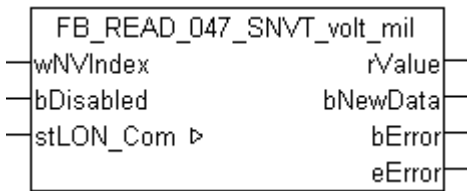
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).



## 7.2.44 FB\_READ\_047\_SNVT\_volt\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt\_mil.

**SNVT Nummer:** 047.

**Beschreibung:** Elektrische Spannung (Millivolt).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: -3276.8 / Max: 3276.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

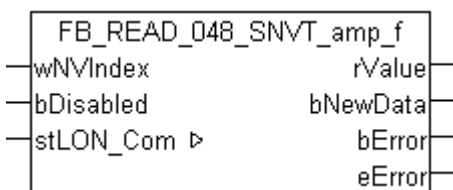
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.45 FB\_READ\_048\_SNVT\_amp\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_amp\_f.

**SNVT Nummer:** 048.

**Beschreibung:** Elektrischer Strom (Ampere).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

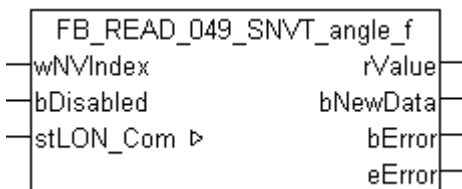
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.46 FB\_READ\_049\_SNVT\_angle\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_angle\_f.

**SNVT Nummer:** 049.

**Beschreibung:** Winkelabstand (Bogenmaß).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

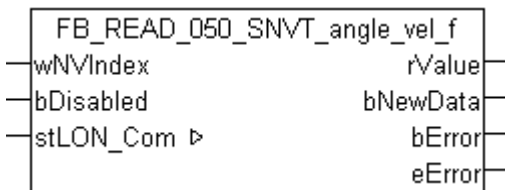
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.47 FB\_READ\_050\_SNVT\_angle\_vel\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_angle\_vel\_f.

**SNVT Nummer:** 050.

**Beschreibung:** Winkelgeschwindigkeit (Radiant / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

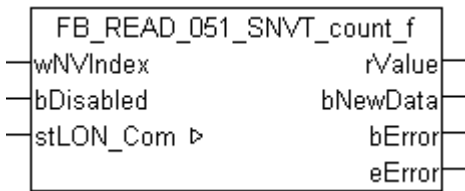
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.48 FB\_READ\_051\_SNVT\_count\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_count\_f.

**SNVT Nummer:** 051.

**Beschreibung:** Absolute Anzahl (Stück).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

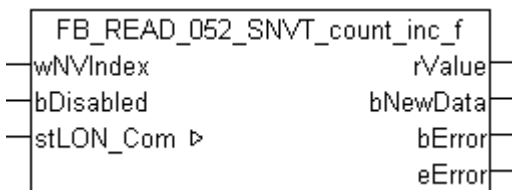
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.49 FB\_READ\_052\_SNVT\_count\_inc\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_count\_inc\_f.

**SNVT Nummer:** 052.

**Beschreibung:** Inkrement Zähler (Stück (delta)).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

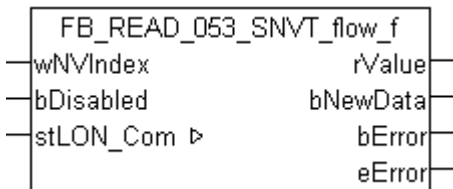
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.50 FB\_READ\_053\_SNVT\_flow\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_flow\_f.

**SNVT Nummer:** 053.

**Beschreibung:** Volumenstrom (Liter / Sekunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

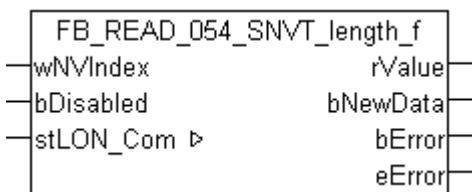
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.51 FB\_READ\_054\_SNVT\_length\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_length\_f.

**SNVT Nummer:** 054.

**Beschreibung:** Länge (Meter).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

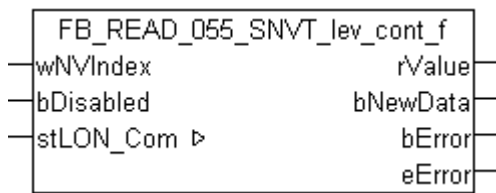
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.52 FB\_READ\_055\_SNVT\_lev\_cont\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_lev\_cont\_f.

**SNVT Nummer:** 055.

**Beschreibung:** Stufenlose Wertänderung (% vom obersten Niveau).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 100.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

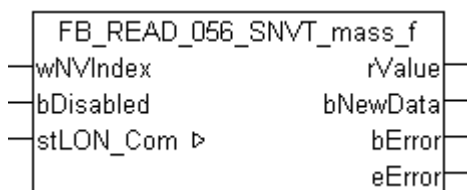
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.53 FB\_READ\_056\_SNVT\_mass\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_mass\_f.

**SNVT Nummer:** 056.

**Beschreibung:** Masse (Gramm).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

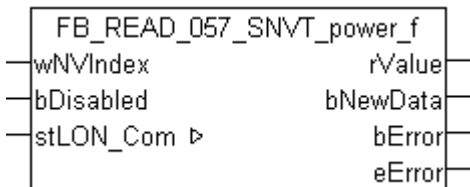
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.54 FB\_READ\_057\_SNVT\_power\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_power\_f.

**SNVT Nummer:** 057.

**Beschreibung:** Leistung (Watt).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```



**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

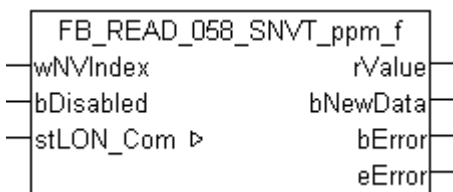
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.55 FB\_READ\_058\_SNVT\_ppm\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ppm\_f.

**SNVT Nummer:** 058.

**Beschreibung:** Konzentration (ppm).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

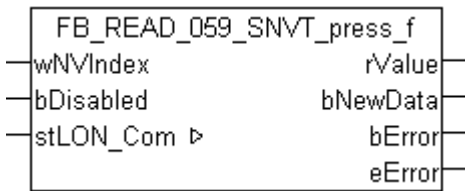
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.56 FB\_READ\_059\_SNVT\_press\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_press\_f.

**SNVT Nummer:** 059.

**Beschreibung:** Druck (Überdruck) (Pascal).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue        : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

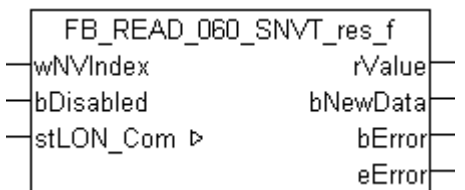
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.57 FB\_READ\_060\_SNVT\_res\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_res\_f.

**SNVT Nummer:** 060.

**Beschreibung:** Elektrischer Widerstand (Ohm).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

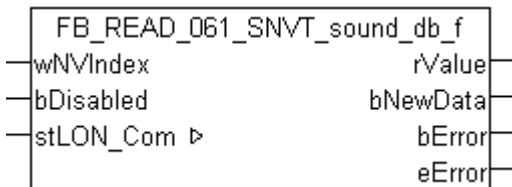
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.58 FB\_READ\_061\_SNVT\_sound\_db\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_sound\_db\_f.

**SNVT Nummer:** 061.

**Beschreibung:** Schalldruckpegel (dB spl).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

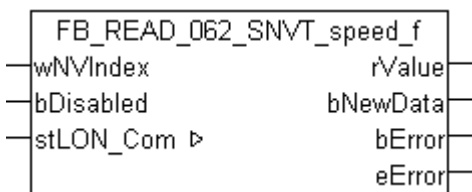
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.59 FB\_READ\_062\_SNVT\_speed\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_speed\_f.

**SNVT Nummer:** 062.

**Beschreibung:** Lineare Geschwindigkeit (Meter / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

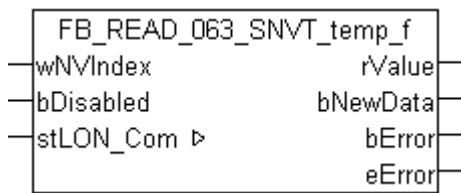
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.60 FB\_READ\_063\_SNVT\_temp\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp\_f.

**SNVT Nummer:** 063.

**Beschreibung:** Temperatur (Grad Celsius).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -273,17 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

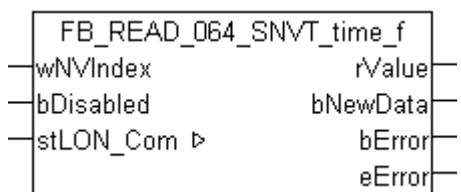
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.61 FB\_READ\_064\_SNVT\_time\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_f.

**SNVT Nummer:** 064.

**Beschreibung:** Abgelaufene Zeit (Sekunden).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

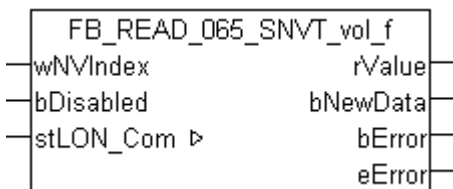
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.62 FB\_READ\_065\_SNVT\_vol\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_vol\_f.

**SNVT Nummer:** 065.

**Beschreibung:** Volumen (Liter).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

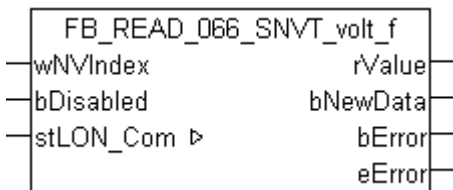
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.63 FB\_READ\_066\_SNVT\_volt\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt\_f.

**SNVT Nummer:** 066.

**Beschreibung:** Elektrische Spannung (Volt).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

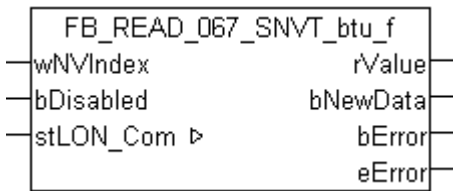
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.64 FB\_READ\_067\_SNVT\_btu\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_btu\_f.

**SNVT Nummer:** 067.

**Beschreibung:** Thermische Energie (BTU).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

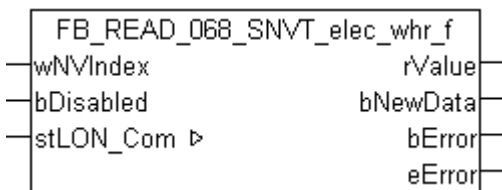
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.65 FB\_READ\_068\_SNVT\_elec\_whr\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_elec\_whr\_f.

**SNVT Nummer:** 068.



**Beschreibung:** Elektrische Energie (Watt / Stunden).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

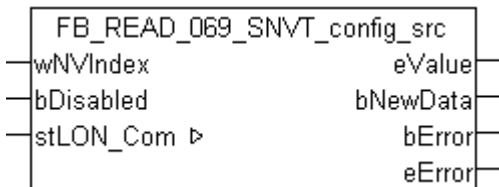
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.66 FB\_READ\_069\_SNVT\_config\_src**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_config\_src.

**SNVT Nummer:** 069.

**Beschreibung:** Konfigurationseigenschaften (Name der Konfigurationsquelle (0 = eigene, 1 = fremde)).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue       : E_LON_config_source_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_config\\_source\\_t](#) [▶ 499]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

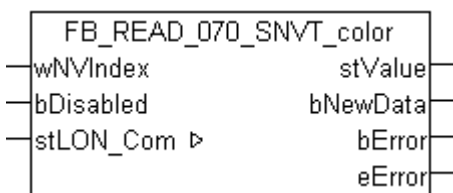
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.67 FB\_READ\_070\_SNVT\_color**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_color.

**SNVT Nummer:** 070.

**Beschreibung:** Farbe nach CIE Norm (L\*,a\*,b).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_color;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur der Daten, die empfangen wurden (siehe [ST\\_LON\\_SNVT\\_color](#) [▶ 569]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

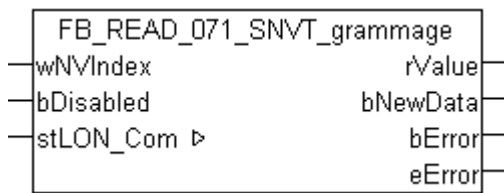
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.68 FB\_READ\_071\_SNVT\_grammage



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_grammage.

**SNVT Nummer:** 071.

**Beschreibung:** Amerikanisches Maß für Papiergewichte und Dichte (Gramm/m<sup>2</sup>).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

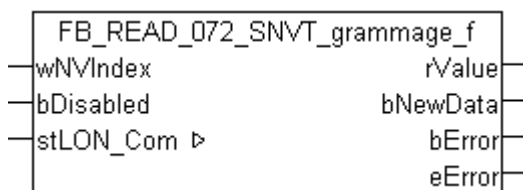
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.69 FB\_READ\_072\_SNVT\_grammage\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_grammage\_f.

**SNVT Nummer:** 072.

**Beschreibung:** Amerikanisches Maß für Papiergewichte und Dichte (Gramm/m<sup>2</sup>).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

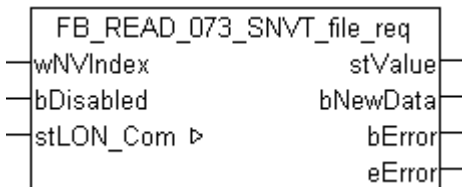
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.70 FB\_READ\_073\_SNVT\_file\_req**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_file\_req.

**SNVT Nummer:** 073.

**Beschreibung:** Dateianforderung.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_file_req;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT file req \[▶ 574\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

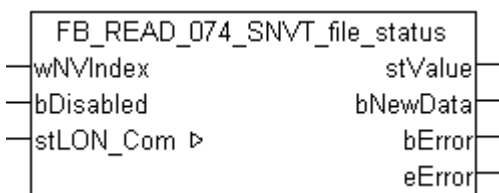
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.71 FB\_READ\_074\_SNVT\_file\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_file\_status.

**SNVT Nummer:** 074.

**Beschreibung:** Dateiinformation (Teil des LONWORKS File-Transfer Protokoll (LW-FTP)).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_file_status;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT file status \[▶ 575\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

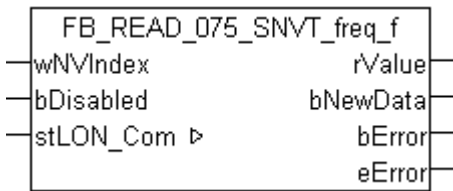
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.72 FB\_READ\_075\_SNVT\_freq\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_freq\_f.

**SNVT Nummer:** 075.

**Beschreibung:** Frequenz (Hertz).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

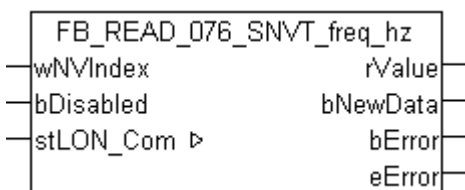
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.73 FB\_READ\_076\_SNVT\_freq\_hz



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_freq\_hz.

**SNVT Nummer:** 076.

**Beschreibung:** Frequenz (Hertz).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

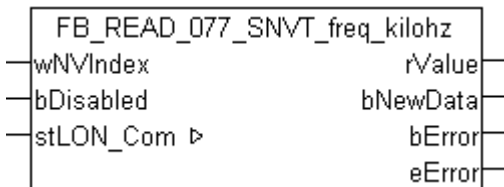
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.74 FB\_READ\_077\_SNVT\_freq\_kilohz**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_freq\_kilohz.

**SNVT Nummer:** 077.

**Beschreibung:** Frequenz (Kilohertz).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

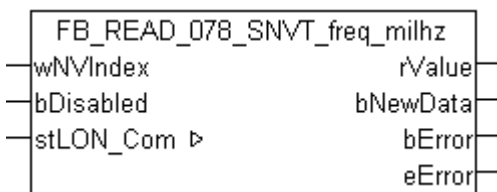
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.75 FB\_READ\_078\_SNVF\_freq\_milhz**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_freq\_milhz.

**SNVT Nummer:** 078.

**Beschreibung:** Frequenz (Millihertz).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6.5535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

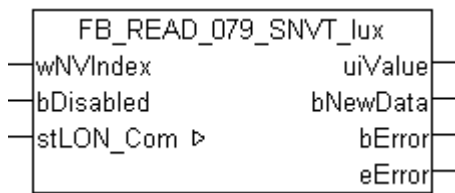
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).



## 7.2.76 FB\_READ\_079\_SNVT\_lux



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_lux.

**SNVT Nummer:** 079.

**Beschreibung:** Beleuchtungsstärke (Lux) 1 lux = 1 lumen/m<sup>2</sup>.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

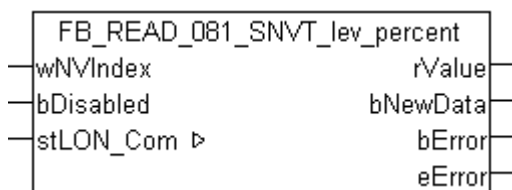
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.77 FB\_READ\_081\_SNVT\_lev\_percent



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_lev\_percent.

**SNVT Nummer:** 081.

**Beschreibung:** Prozentangabe in 0,005-%-Schritten mit Vorzeichen. Anstelle von SNVT\_switch sollte SNVT\_lev\_percent benutzt werden. Ausnahmen bilden Netzwerkvariablen die einen Prozentwert übertragen und die von SNVT\_lev\_percent unterstützte zusätzlich Auflösung erfordern oder Netzwerkvariablen von Funktionsprofilen, die hauptsächlich für das Verknüpfen mit SNVT\_lev\_percent Variablen anderer Funktionsprofile gedacht sind.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue        : REAL;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**rValue:** Min: -163.84 / Max: 163.835.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

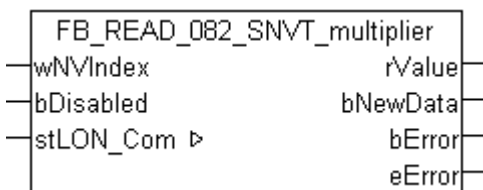
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.78 FB\_READ\_082\_SNVT\_multiplier**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_multiplier.

**SNVT Nummer:** 082.

**Beschreibung:** Multiplikator in 0,0005- Schritten (16-bit Vorzeichenloser Wert).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 32.7675.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

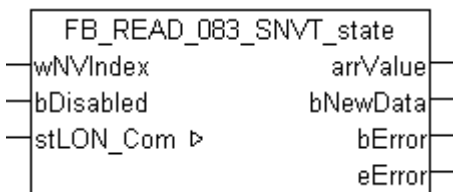
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.79 FB\_READ\_083\_SNVT\_state**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_state.

**SNVT Nummer:** 083.

**Beschreibung:** Statusinformation (16 individuelle Bitwerte). Jeder Status ist ein boolescher Wert.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
arrValue     : ARRAY [0..15] OF BOOL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**arrValue:** 0-15 Bit.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

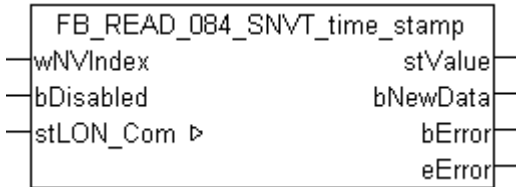
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.80 FB\_READ\_084\_SNVT\_time\_stamp**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_stamp.

**SNVT Nummer:** 084.

**Beschreibung:** Zeitstempel (Jahr, Monat, Tag, Stunde, Minute, Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : Timestruct;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_Error;
```

**stValue:** Struktur, der empfangen werden soll (siehe [Timestruct](#)). Die Strukturvariablen *wDayOfWeek* und *wMilliseconds* sind hier nicht gültig und immer null.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

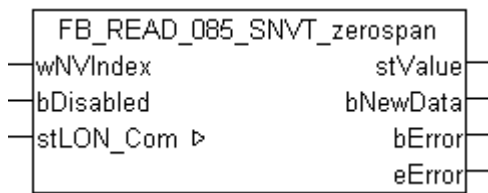
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_Error](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.81 FB\_READ\_085\_SNVT\_zerospan



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_zerospan.

**SNVT Nummer:** 085.

**Beschreibung:** Nullpunkt und Proportionalitätsfaktor. Lineare Transformationsparameter: Multiplizieren mit Proportionalitätsfaktor, dann addieren mit Nullpunkt Offset.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_zerospan;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_zerospan \[▶ 589\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

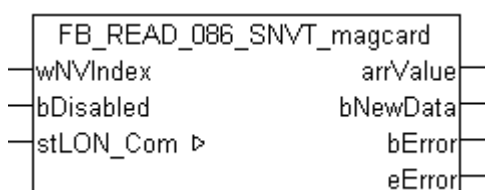
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.82 FB\_READ\_086\_SNVT\_magcard



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_magcard.

**SNVT Nummer:** 086.

**Beschreibung:** ISO 7811 (40 Hexadezimal Zahlen). Daten entsprechend des ISO 7811 Standards für Magnetkartenleser.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
arrValue      : ARRAY [0..40] OF BYTE;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**arrValue:** 1-40 BYTE.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

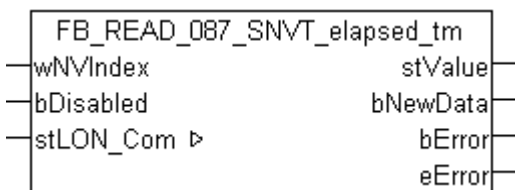
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.83 FB\_READ\_087\_SNVT\_elapsed\_tm**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_elapsed\_tm.

**SNVT Nummer:** 087.

**Beschreibung:** Abgelaufene Zeit (Tag, Stunde, Minute, Sekunde, Millisekunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_elapsed_tm;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_elapsed\\_tm](#) [▶ 572]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

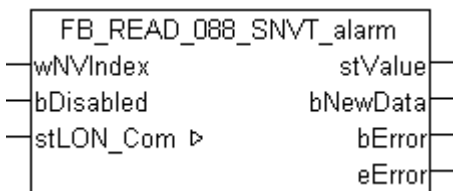
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.84 FB\_READ\_088\_SNVT\_alarm**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_alarm.

**SNVT Nummer:** 088.

**Beschreibung:** Alarm Status.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_alarm;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_alarm](#) [▶ 565]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

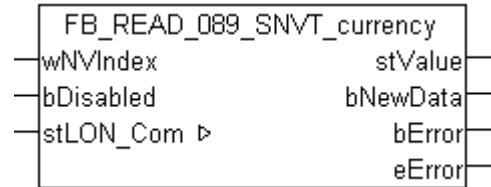
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.85 FB\_READ\_089\_SNVT\_currency**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_currency.

**SNVT Nummer:** 089.

**Beschreibung:** Währung (Einheit (Euro,...), Multiplikator, Wert).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_currency;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_currency](#) [▶ 570]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

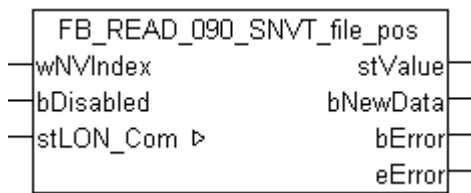
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).



## 7.2.86 FB\_READ\_090\_SNVT\_file\_pos



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_file\_pos.

**SNVT Nummer:** 090.

**Beschreibung:** Dateiposition (Pointer, Länge).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_file_pos;
bNewData     : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_file\\_pos \[▶ 574\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

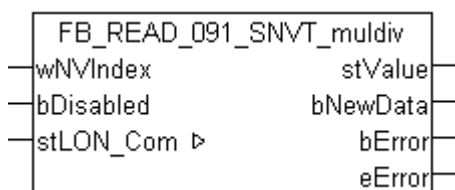
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.87 FB\_READ\_091\_SNVT\_muldiv



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_muldiv.

**SNVT Nummer:** 091.

**Beschreibung:** Verstärkungsfaktor/Dämpfungsfaktor (Multiplikator, Divisor).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_muldiv;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_muldiv](#) [▶ 578]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

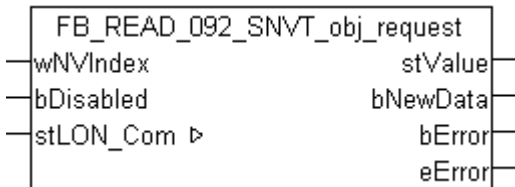
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.88 FB\_READ\_092\_SNVT\_obj\_request**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_obj\_request.

**SNVT Nummer:** 092.

**Beschreibung:** Funktionsauswahl (ID, Request).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_obj_request;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_obj\\_request \[▶ 579\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

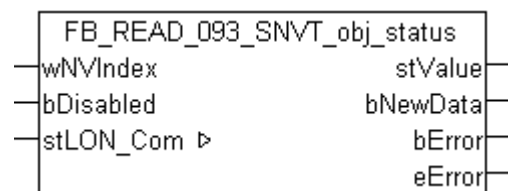
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.89 FB\_READ\_093\_SNVT\_obj\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_obj\_status.

**SNVT Nummer:** 093.

**Beschreibung:** Objektstatus (ID, Status (4 Byte)).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_obj_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_obj\\_status \[▶ 579\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

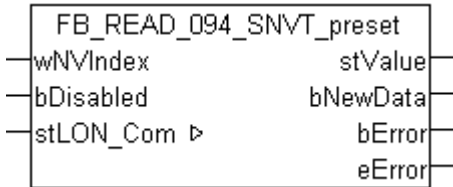
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.90 FB\_READ\_094\_SNVT\_preset**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_preset.

**SNVT Nummer:** 094.

**Beschreibung:** Voreinstellung (Lernmodus, Wert, Zeit).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_preset;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_preset](#) [▶ 581]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

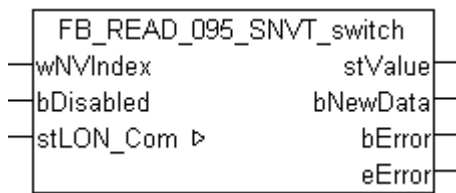
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.91 FB\_READ\_095\_SNVT\_switch



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_switch.

**SNVT Nummer:** 095.

**Beschreibung:** Schalter (Wert, Status).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_switch;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_switch \[▶ 587\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

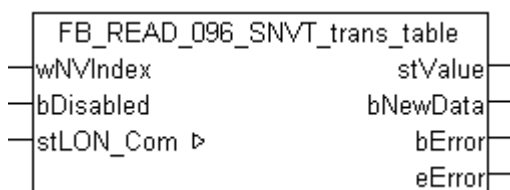
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.92 FB\_READ\_096\_SNVT\_trans\_table



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_trans\_table.

**SNVT Nummer:** 096.

**Beschreibung:** Übersetzungstabelle (Anzahl Werte, Interpolation).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_trans_table;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_trans\\_table](#) [▶ 589]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

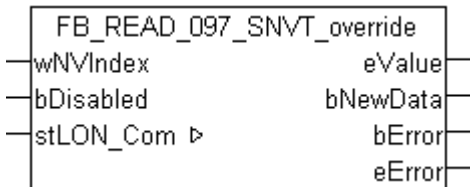
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.93 FB\_READ\_097\_SNVT\_override**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_override.

**SNVT Nummer:** 097.

**Beschreibung:** Übersteuerungsmodus (Enumeration: 0 = aktuellen Wert halten, 1 = spezifischen Wert setzen, 2 = default Wert setzen).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue      : E_LON_override_t;
bNewData   : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_override\\_t](#) [▶ 518]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

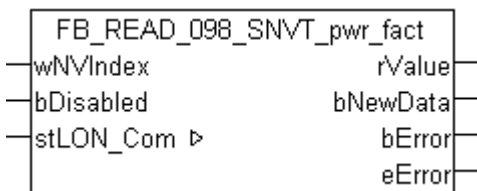
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com   : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.94 FB\_READ\_098\_SNVT\_pwr\_fact**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pwr\_fact.

**SNVT Nummer:** 098.

**Beschreibung:** Leistungsfaktor (Multiplikator).

**VAR\_INPUT**

```
wNVIndex    : WORD;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData   : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
```

**rValue:** Min: -1 / Max: 1.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

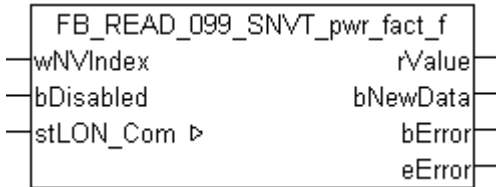
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.95 FB\_READ\_099\_SNVT\_pwr\_fact\_f**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pwr\_fact\_f.

**SNVT Nummer:** 099.

**Beschreibung:** Leistungsfaktor (Multiplikator).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -1 / Max: 1.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

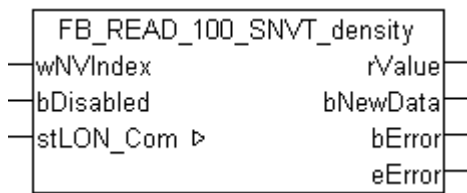
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).



## 7.2.96 FB\_READ\_100\_SNVT\_density



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_density.

**SNVT Nummer:** 100.

**Beschreibung:** Dichte (kg/m<sup>3</sup>).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 32767.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

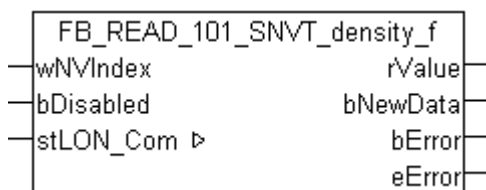
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Send-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.97 FB\_READ\_101\_SNVT\_density\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_density\_f.

**SNVT Nummer:** 101.

**Beschreibung:** Dichte (kg/m³).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

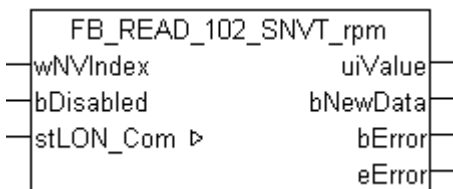
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.98 FB\_READ\_102\_SNVT\_rpm**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_rpm.

**SNVT Nummer:** 102.

**Beschreibung:** Drehzahl (Umdrehungen/Minute (RPM)).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

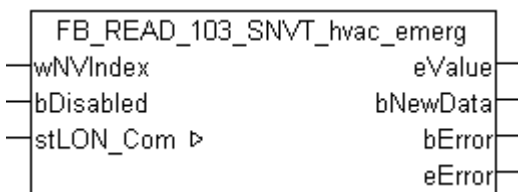
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.99 FB\_READ\_103\_SNVT\_hvac\_emerg**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_emerg.

**SNVT Nummer:** 103.

**Beschreibung:** HLK Notbetrieb (Betriebsarten).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue : E_LON_emerg_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_emerg\\_t](#) [▶ 505]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

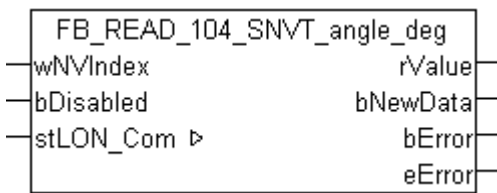
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.100 FB\_READ\_104\_SNVT\_angle\_deg



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_angle\_deg.

**SNVT Nummer:** 104.

**Beschreibung:** Winkelangabe in 1/50-Grad-Schritten.

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -359.98 / Max: 360.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

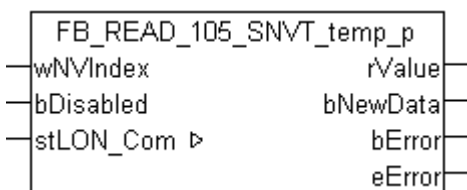
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.101 FB\_READ\_105\_SNVT\_temp\_p



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp\_p.

**SNVT Nummer:** 105.

**Beschreibung:** Temperatur (Grad Celsius).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -273.17 / Max: 327.67.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

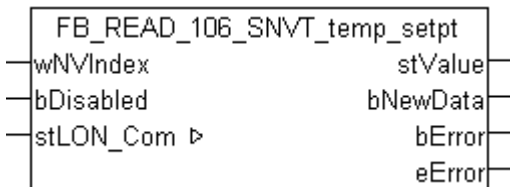
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.102 FB\_READ\_106\_SNVT\_temp\_setpt**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp\_setpt.

**SNVT Nummer:** 106.

**Beschreibung:** Temperatur (6 Temperaturwerte).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_temp_setpt;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT temp\\_setpt \[▶ 587\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

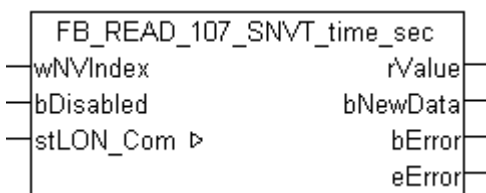
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.103 FB\_READ\_107\_SNVT\_time\_sec**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_sec.

**SNVT Nummer:** 107.

**Beschreibung:** Abgelaufene Zeit (Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 6553.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

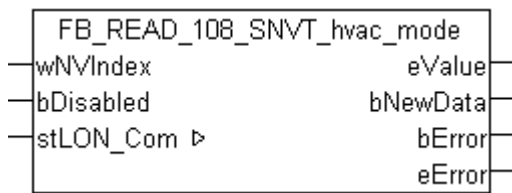
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.104 FB\_READ\_108\_SNVT\_hvac\_mode



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_mode.

**SNVT Nummer:** 108.

**Beschreibung:** HLK Betriebsmodus (Betriebsarten).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
eValue       : E_LON_hvac_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_hvac\\_t](#) [▶ 514]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

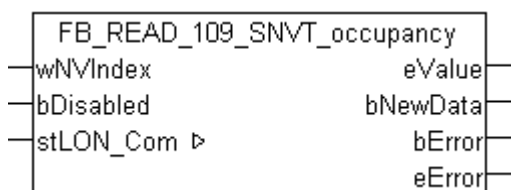
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.105 FB\_READ\_109\_SNVT\_occupancy



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_occupancy.

**SNVT Nummer:** 109.

**Beschreibung:** Präsenzmeldung (Status).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue       : E_LON_occup_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_occup\\_t](#) [▶ 518]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

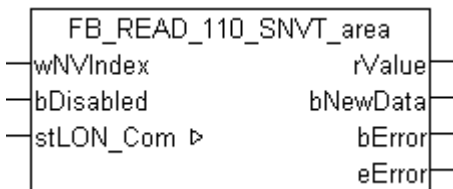
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.106 FB\_READ\_110\_SNVT\_area



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_area.

**SNVT Nummer:** 110.

**Beschreibung:** Fläche (Quadratmeter).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.



**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 13.107.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

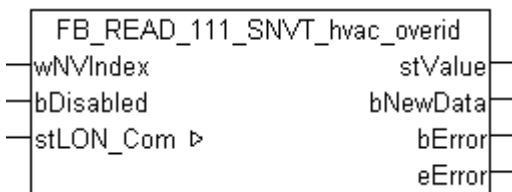
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.107 FB\_READ\_111\_SNVT\_hvac\_overid**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_overid.

**SNVT Nummer:** 111.

**Beschreibung:** HLK Übersteuerungsmodus (Modus, Position/Durchfluss, Min/Max Durchflusswert).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_hvac_overid;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_hvac\\_overid \[▶ 575\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

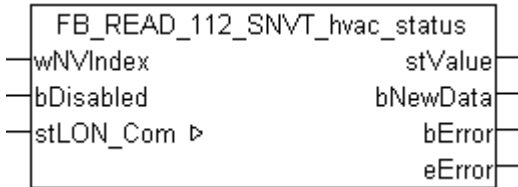
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.108 FB\_READ\_112\_SNVT\_hvac\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_status.

**SNVT Nummer:** 112.

**Beschreibung:** HLK Status (Modus, 5 Leistungsangaben, Alarmflag).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_hvac_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_hvac\\_status](#) [▶ 576]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

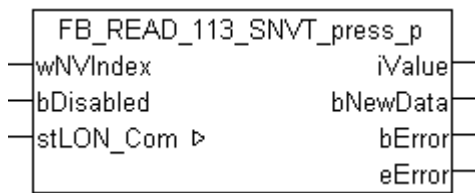
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.109 FB\_READ\_113\_SNVT\_press\_p



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_press\_p.

**SNVT Nummer:** 113.

**Beschreibung:** Druck (Überdruck) (Pascal).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
iValue       : INT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**iValue:** Min: -32768 / Max: 32767.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

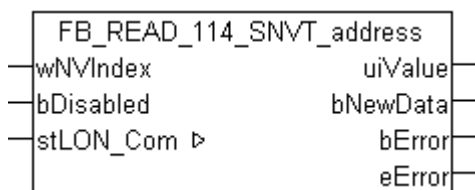
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.110 FB\_READ\_114\_SNVT\_address



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_address.

**SNVT Nummer:** 114.

**Beschreibung:** Neuron Adresse (16-bit Adress Wert).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**uiValue:** Min: 16384 / Max: 64767.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

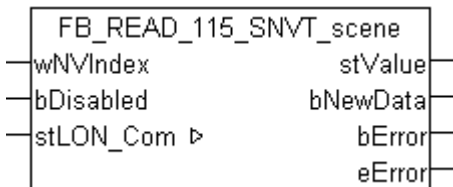
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.111 FB\_READ\_115\_SNVT\_scene**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_scene.

**SNVT Nummer:** 115.

**Beschreibung:** Szenen (Funktion (abrufen/lernen), Szenennummer).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_scene;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_scene](#) [▶ 586]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

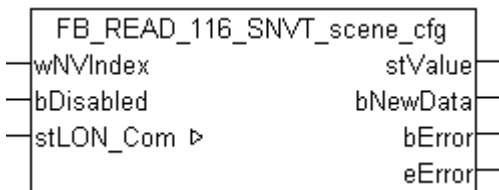
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.112 FB\_READ\_116\_SNVT\_scene\_cfg**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_scene\_cfg.

**SNVT Nummer:** 116.

**Beschreibung:** Szenen Einstellung (Funktion , Szenennummer, Einstellung, Übergangsnummer, Übergangszeit, Verzögerungszeit, Priorität).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_scene_cfg;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_scene\\_cfg](#) [▶ 586]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

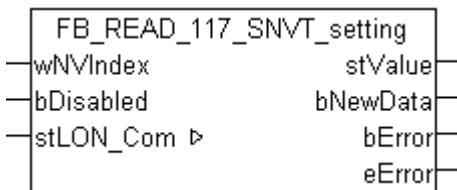
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.113 FB\_READ\_117\_SNVT\_setting



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_setting.

**SNVT Nummer:** 117.

**Beschreibung:** Einstellungen (Funktion, Wert, Übergangsnummer).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_setting;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_setting](#) [▶ 586]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

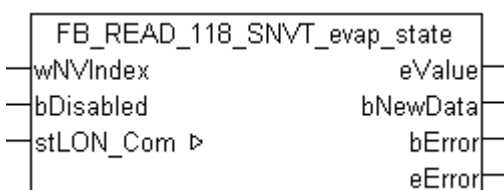
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.114 FB\_READ\_118\_SNVT\_evap\_state



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_evap\_state.

**SNVT Nummer:** 118.

**Beschreibung:** Verdampferstatus (Enumeration).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue        : E_LON_evap_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_evap\\_t \[▶ 507\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

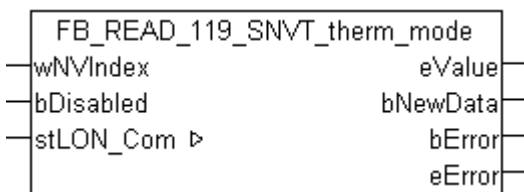
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.115 FB\_READ\_119\_SNVT\_therm\_mode



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_therm\_mode.

**SNVT Nummer:** 119.

**Beschreibung:** Thermostatmodus (Enumeration (Freigabe, Modulation)).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue      : E_LON_therm_mode_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_therm\\_mode\\_t](#) [► 534]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

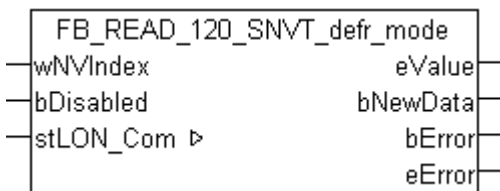
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

**7.2.116 FB\_READ\_120\_SNVT\_defr\_mode**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_defr\_mode.

**SNVT Nummer:** 120.

**Beschreibung:** Abtaumodus (Enumeration).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue      : E_LON_defrost_mode_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_defrost\\_mode\\_t](#) [► 502]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

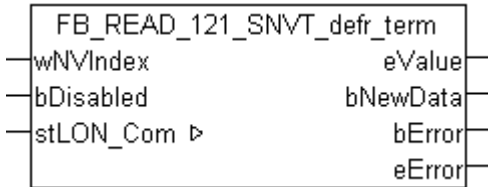


**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.117 FB\_READ\_121\_SNVT\_defr\_term**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_defr\_term.

**SNVT Nummer:** 121.

**Beschreibung:** Abschluss des Abtauvorgangs (Enumeration).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue : E_LON_defrost_term_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_defrost\\_term\\_t](#) [[▶ 503](#)]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

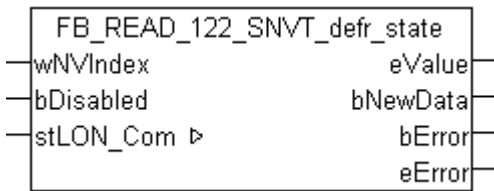
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

### 7.2.118 FB\_READ\_122\_SNVT\_defr\_state



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_defr\_state.

**SNVT Nummer:** 122.

**Beschreibung:** Abtaustatus (Enumeration).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
eValue       : E_LON_defrost_state_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_defrost\\_state\\_t](#) [▶ 503]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

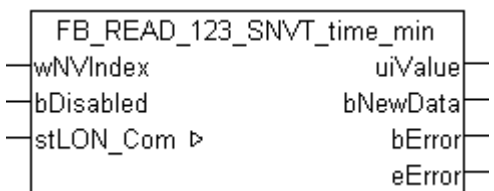
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.119 FB\_READ\_123\_SNVT\_time\_min



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_min.

**SNVT Nummer:** 123.

**Beschreibung:** Abgelaufen Zeit (Minuten).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

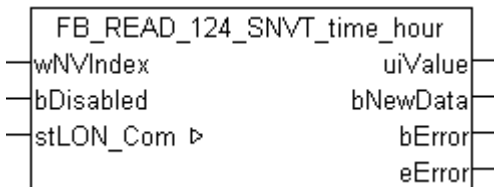
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.120 FB\_READ\_124\_SNVT\_time\_hour



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_hour.

**SNVT Nummer:** 124.

**Beschreibung:** Abgelaufene Zeit (Stunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

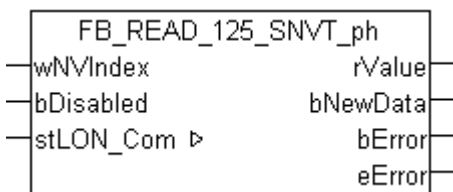
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.121 FB\_READ\_125\_SNVT\_ph**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ph.

**SNVT Nummer:** 125.

**Beschreibung:** Säuregehalt (pH). Verhältnis der Ionenkonzentration.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -32.768 / Max: 32.767.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

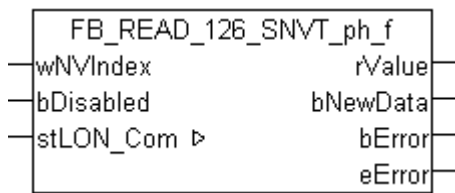
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.122 FB\_READ\_126\_SNVT\_ph\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ph\_f.

**SNVT Nummer:** 126.

**Beschreibung:** Säuregehalt (pH). Verhältnis der Ionenkonzentration.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

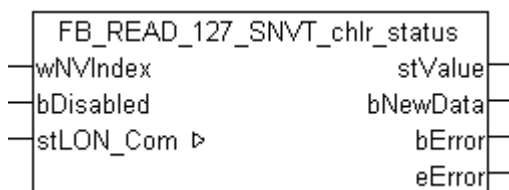
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.123 FB\_READ\_127\_SNVT\_chlr\_status



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_chlr\_status.

**SNVT Nummer:** 127.

**Beschreibung:** Kälteerzeugerstatus (run mode, op mode, Statusbits).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_chlr_status;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_chlr\\_status](#) [▶ 566]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

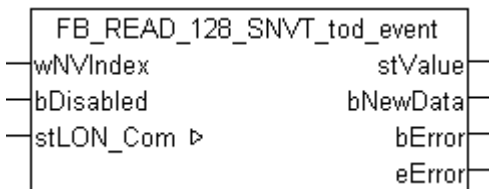
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.124 FB\_READ\_128\_SNVT\_tod\_event**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_tod\_event.

**SNVT Nummer:** 128.

**Beschreibung:** Präsenzzeit (Besetztstatus, nächste Besetztzeit).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_tod_event;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_tod\\_event \[▶ 588\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

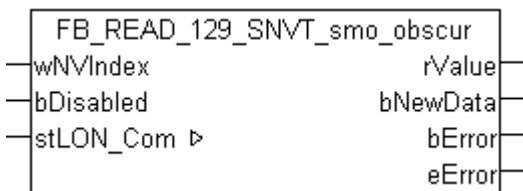
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.125 FB\_READ\_129\_SNVT\_smo\_obscur**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_smo\_obscur.

**SNVT Nummer:** 129.

**Beschreibung:** Verdunkelung durch Rauch (Verdunkelung %).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

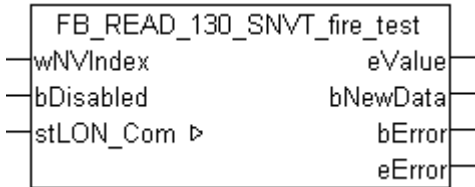
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.126 FB\_READ\_130\_SNVT\_fire\_test**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_fire\_test.

**SNVT Nummer:** 130.

**Beschreibung:** Anforderung Test Brandmeldeanlage (Brandmeldetest Benennungen).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue : E_LON_fire_test_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_fire\\_test\\_t](#) [▶ 510]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

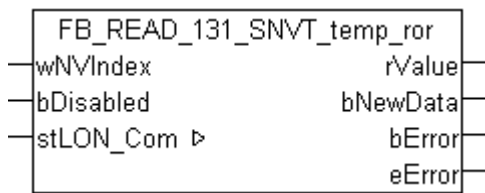
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).



## 7.2.127 FB\_READ\_131\_SNVT\_temp\_ror



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp\_ror.

**SNVT Nummer:** 131.

**Beschreibung:** Wert der Temperaturänderung/-erhöhung (Grad Celsius/Minute).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -16384 / Max: 16383.5.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

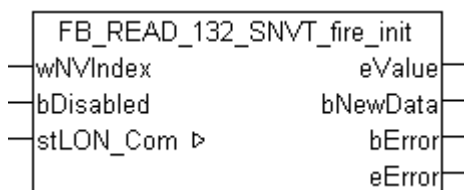
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.128 FB\_READ\_132\_SNVT\_fire\_init



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_fire\_init.

**SNVT Nummer:** 132.

**Beschreibung:** Brandmeldertyp (Brandmeldertypen Namen).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue       : E_LON_fire_initiator_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe E\_LON\_fire\_initiator t [▶ 510]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

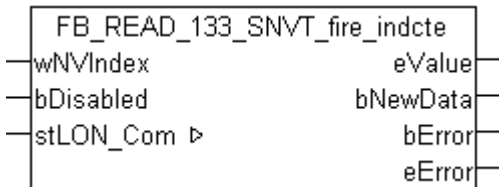
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.129 FB\_READ\_133\_SNVT\_fire\_indcte**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_fire\_indcte.

**SNVT Nummer:** 133.

**Beschreibung:** Brandmeldeanzeige (Brandmeldeanzeigen Namen).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue      : E_LON_fire_indicator_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_fire\\_indicator\\_t](#) [▶ 509]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

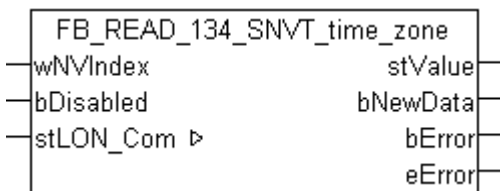
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.130 FB\_READ\_134\_SNVT\_time\_zone**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_zone.

**SNVT Nummer:** 134.

**Beschreibung:** Zeitzonenbeschreibung (Offset, Typ, Start Sommerzeit, Ende Sommerzeit).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_time_zone;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_time\\_zone](#) [▶ 588]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

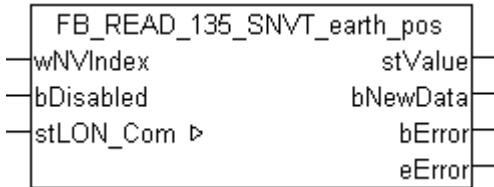
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.131 FB\_READ\_135\_SNVT\_earth\_pos**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_earth\_pos.

**SNVT Nummer:** 135.

**Beschreibung:** Position auf der Erde (Breiten & Längen Ausrichtung, Breitengrad & Min, Längengrad & Min, Höhe).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_earth_pos;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_earth\\_pos](#) [▶ 571]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

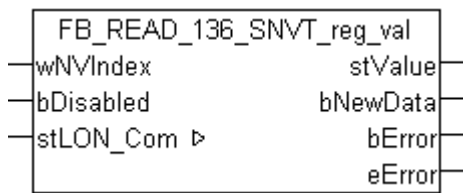
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.132 FB\_READ\_136\_SNVT\_reg\_val



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_reg\_val.

**SNVT Nummer:** 136.

**Beschreibung:** Registerwert (Rohwert, Bit-Code, Anzahl der Dezimalstellen).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_reg_val;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_reg\\_val \[▶ 585\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

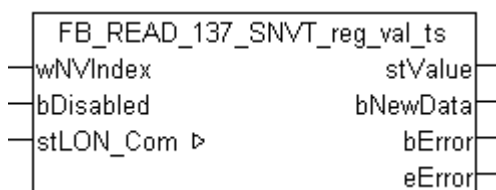
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.133 FB\_READ\_137\_SNVT\_reg\_val\_ts



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_reg\_val\_ts.

**SNVT Nummer:** 137.

**Beschreibung:** Registerwert (Rohwert, bit-Code, Anzahl der Dezimalstellen, Status, Zustand, Zeitstempel).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_reg_val_ts;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_reg\\_val\\_ts \[▶ 585\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

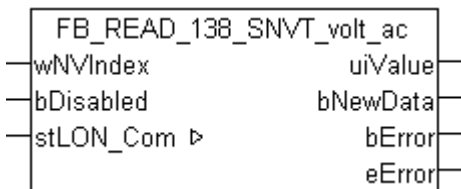
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.134 FB\_READ\_138\_SNVT\_volt\_ac**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_volt\_ac.

**SNVT Nummer:** 138.

**Beschreibung:** Wechselspannung (Volt AC).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue      : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

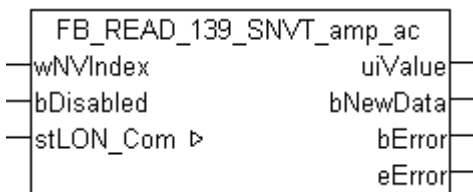
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.135 FB\_READ\_139\_SNVT\_amp\_ac**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_amp\_ac.

**SNVT Nummer:** 139.

**Beschreibung:** Wechselstrom (Ampere AC).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue      : UINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

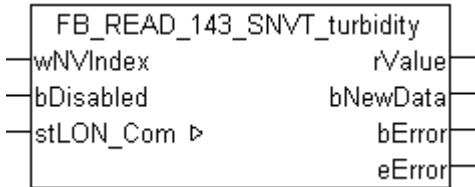
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.136 FB\_READ\_143\_SNVT\_turbidity**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_turbidity.

**SNVT Nummer:** 143.

**Beschreibung:** Trübung (Trübeeinheit).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 65.535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

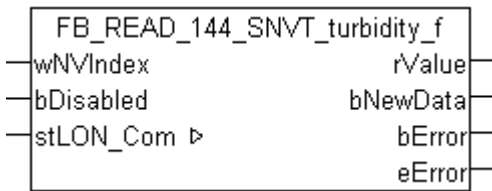
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).



### 7.2.137 FB\_READ\_144\_SNVT\_turbidity\_f



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_turbidity\_f.

**SNVT Nummer:** 144.

**Beschreibung:** Trübung (Trübeeinheit).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 3.40E+38.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

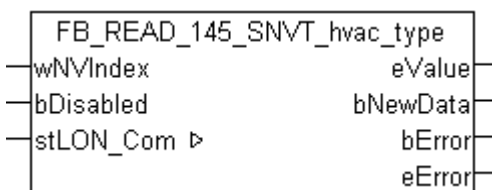
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.138 FB\_READ\_145\_SNVT\_hvac\_type



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_type.

**SNVT Nummer:** 145.

**Beschreibung:** HLK Anlagentyp (HLK Anlagentypbeschreibung).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue        : E_LON_hvac_hvt_t;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_hvac\\_hvt\\_t](#) [▶ 511]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

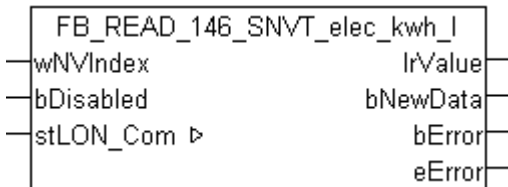
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.139 FB\_READ\_146\_SNVT\_elec\_kwh\_I**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_elec\_kwh\_I.

**SNVT Nummer:** 146.

**Beschreibung:** Elektrische Energie (Kilowatt / Stunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
lrValue      : LREAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**lrValue:** Min: -214748364.8 / Max: 214748364.7.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

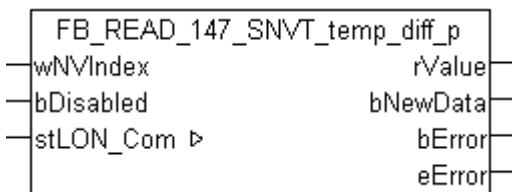
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.140 FB\_READ\_147\_SNVT\_temp\_diff\_p**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_temp\_diff\_p.

**SNVT Nummer:** 147.

**Beschreibung:** Temperaturdifferenz (Grad Celsius).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.67.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

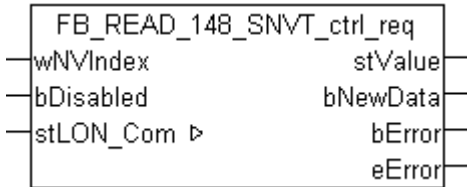
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.141 FB\_READ\_148\_SNVT\_ctrl\_req**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ctrl\_req.

**SNVT Nummer:** 148.

**Beschreibung:** Steuerungsanfrage (Empfänger ID, Sender ID, Sender Priorität). Anfrage einer Steuerung für die Steuererlaubnis einer anderen Steuerung.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_ctrl_req;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_ctrl\\_req](#) [[▶ 569](#)]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

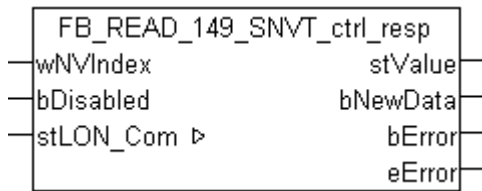
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

## 7.2.142 FB\_READ\_149\_SNVT\_ctrl\_resp



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ctrl\_resp.

**SNVT Nummer:** 149.

**Beschreibung:** Steuerungsantwort (Status, Sender, Controller ID). Antwort auf eine Steuerungsanfrage.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_ctrl_resp;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_ctrl\\_resp \[▶ 570\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

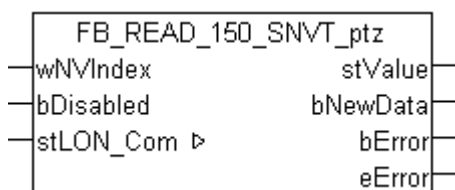
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.143 FB\_READ\_150\_SNVT\_ptz



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ptz.

**SNVT Nummer:** 150.

**Beschreibung:** Kamera PTZ (SNZ) (Schwenkung, Schwenkgeschwindigkeit, Neigung, Neigungsgeschwindigkeit, Zoom, Zoomgeschwindigkeit).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_ptz;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_ptz](#) [▶ 582]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

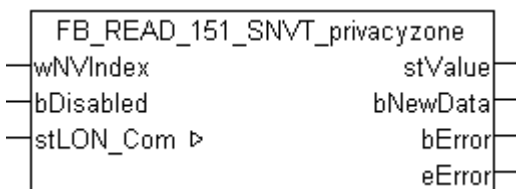
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.144 FB\_READ\_151\_SNVT\_privacyzone**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_privacyzone.

**SNVT Nummer:** 151.

**Beschreibung:** Privatzone (Aktionsbereich, Zonenummer, Kamera ID). Gewisse Bereiche können von der Kamera ausgenommen werden.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_privacyzone;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_privacyzone \[▶ 581\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

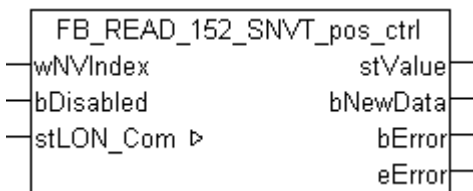
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.145 FB\_READ\_152\_SNVT\_pos\_ctrl**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pos\_ctrl.

**SNVT Nummer:** 152.

**Beschreibung:** Positionseinstellung für Kameras in Netzwerken (Empfänger, Controller ID, Controller Priorität, Funktion, Aktion, Wert).

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_pos_ctrl;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_pos\\_ctrl \[▶ 580\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

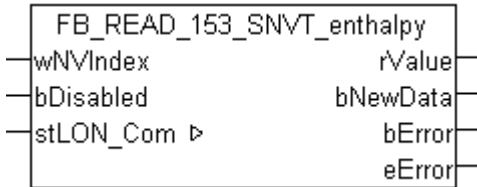
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.146 FB\_READ\_153\_SNVT\_enthalpy**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_enthalpy.

**SNVT Nummer:** 153.

**Beschreibung:** Enthalpie (Kilojoules/kg).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue : REAL;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.67.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

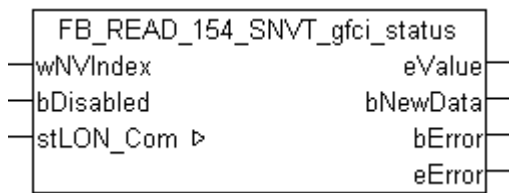
**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).



## 7.2.147 FB\_READ\_154\_SNVT\_gfci\_status



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_gfci\_status.

**SNVT Nummer:** 154.

**Beschreibung:** Fehlerstromschutzschalter Status (Fehlerstromschutzschalter Status Text).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
eValue       : E_LON_gfci_status_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_gfci\\_status\\_t \[▶ 511\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

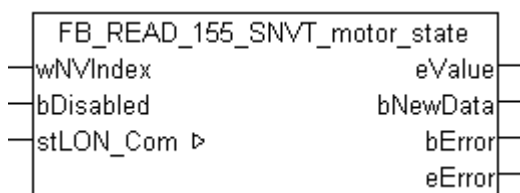
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.148 FB\_READ\_155\_SNVT\_motor\_state



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_motor\_state.

**SNVT Nummer:** 155.

**Beschreibung:** Motorstatus (Motorstatus Benennung).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue       : E_LON_motor_state_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_motor\\_state\\_t](#) [▶ 516]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

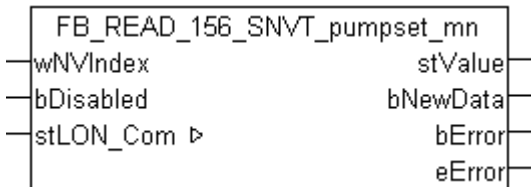
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.149 FB\_READ\_156\_SNVT\_pumpset\_mn**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pumpset\_mn.

**SNVT Nummer:** 156.

**Beschreibung:** Pumpengruppe (Hauptpumpe, Hilfspumpe, Priorität, Bereit, Not-Aus, Hauptpumpe freigegeben, Hilfspumpe freigegeben, Wartungsanforderung). Aufeinander synchronisierte Vakuumpumpen.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_pumpset_mn;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_pumpset\\_mn \[► 583\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

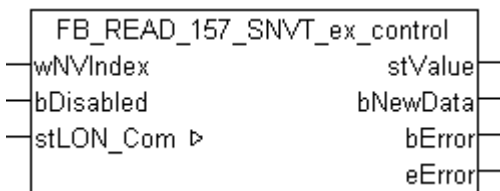
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

**7.2.150 FB\_READ\_157\_SNVT\_ex\_control**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ex\_control.

**SNVT Nummer:** 157.

**Beschreibung:** Exklusive Kontrolle (Status, Adresse). Ein Gerät hat die exklusive Kontrolle über ein anderes Gerät.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_ex_control;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_ex\\_control \[► 574\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

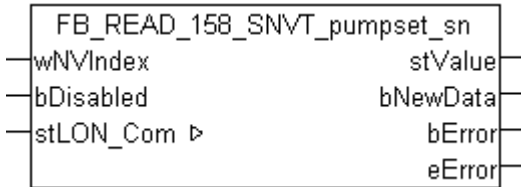
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.151 FB\_READ\_158\_SNVT\_pumpset\_sn**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pumpset\_sn.

**SNVT Nummer:** 158.

**Beschreibung:** Pumpengruppensensor (Verdünnung, Ausstoß, Druck, Vakuum, ...).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_pumpset_sn;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_pumpset\\_sn](#) [▶ 583]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

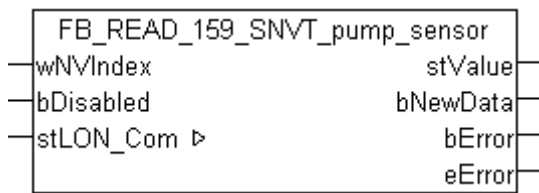
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.152 FB\_READ\_159\_SNVT\_pump\_sensor



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_pump\_sensor.

**SNVT Nummer:** 159.

**Beschreibung:** Pumpensensor (Geschwindigkeit, Temperatur, Status).

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_pump_sensor;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_pump\\_sensor](#) [▶ 582]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

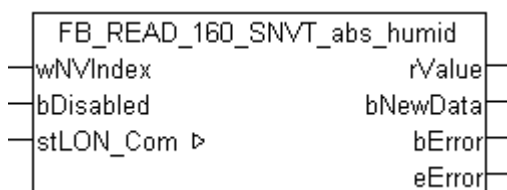
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.153 FB\_READ\_160\_SNVT\_abs\_humid



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_abs\_humid.

**SNVT Nummer:** 160.

**Beschreibung:** Absolute Feuchtigkeit (Gramm/Kilogramm).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 655.35.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

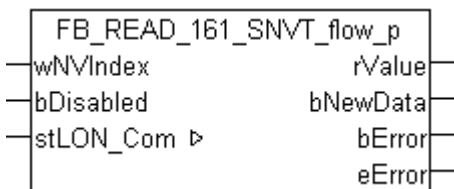
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.154 FB\_READ\_161\_SNVT\_flow\_p**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_flow\_p.

**SNVT Nummer:** 161.

**Beschreibung:** Volumenstrom (Kubikmeter / Stunde).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 655.35.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

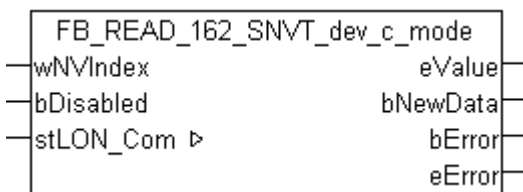
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.155 FB\_READ\_162\_SNVT\_dev\_c\_mode**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_dev\_c\_mode.

**SNVT Nummer:** 162.

**Beschreibung:** Gerätebetriebsart (Gerätebetriebsart Zustände).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue : E_LON_device_c_mode_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_device\\_c\\_mode\\_t \[▶ 504\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

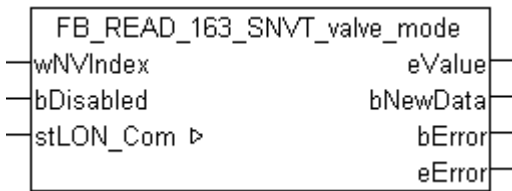
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.156 FB\_READ\_163\_SNVT\_valve\_mode



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_valve\_mode.

**SNVT Nummer:** 163.

**Beschreibung:** Ventilzustand.

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
eValue       : E_LON_valve_mode_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_valve\\_mode\\_t](#) [▶ 535]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

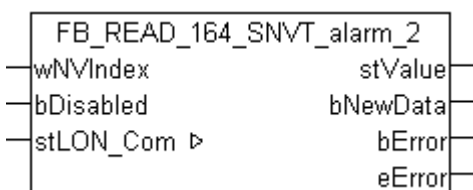
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.157 FB\_READ\_164\_SNVT\_alarm\_2



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_alarm\_2.

**SNVT Nummer:** 164.



**Beschreibung:** Alarmstatus 2. Meldet den Alarmstatus eines Funktionsbausteins oder Geräts. Ersetzt SNVT\_alarm.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_alarm_2;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_alarm\\_2](#) [▶ 566]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

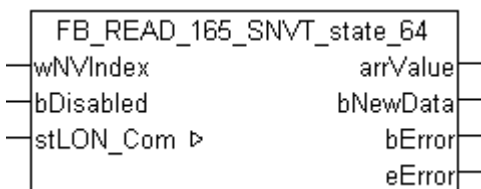
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Send-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.158 FB\_READ\_165\_SNVT\_state\_64**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_state\_64.

**SNVT Nummer:** 165.

**Beschreibung:** Statusinformation (64 individuelle Bitwerte). Jeder Status ist ein boolescher Wert.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
arrValue      : ARRAY [0..63] OF BOOL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**arrValue:** 0-63 Bit.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

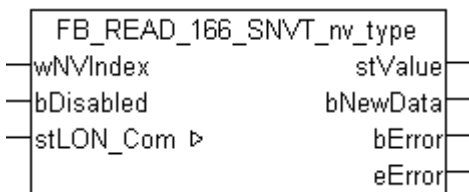
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.159 FB\_READ\_166\_SNVT\_nv\_type**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_nv\_type.

**SNVT Nummer:** 166.

**Beschreibung:** Netzwerkvariablentyp. Typbeschreibung für veränderbare Netzwerkvariablen.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_nv_type;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_nv\\_type \[▶ 578\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

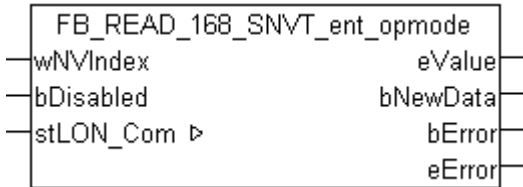
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.160 FB\_READ\_168\_SNVT\_ent\_opmode**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ent\_opmode.

**SNVT Nummer:** 168.

**Beschreibung:** Betriebsart für Zutrittsobjekte (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue : E_LON_ent_opmode_cmd_t;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_ent\\_opmode\\_cmd\\_t](#) [▶ 506]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

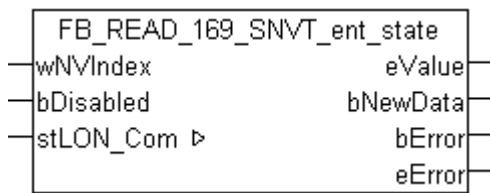
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.161 FB\_READ\_169\_SNVT\_ent\_state



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ent\_state.

**SNVT Nummer:** 169.

**Beschreibung:** Zustand für Zutrittsobjekte (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
eValue       : E_LON_ent_cmd_t;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_ent\\_cmd\\_t](#) [▶ 506]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

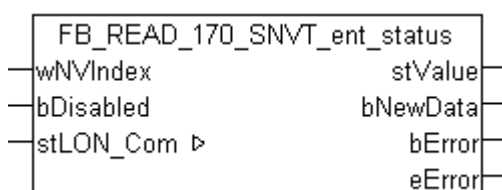
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.162 FB\_READ\_170\_SNVT\_ent\_status



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_ent\_status.

**SNVT Nummer:** 170.

**Beschreibung:** Status von Zutrittsobjekten (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_ent_status;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_ent\\_status \[▶ 572\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

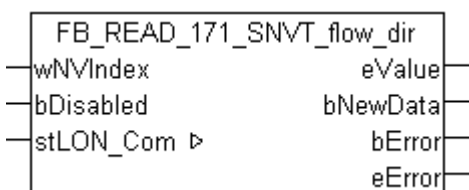
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.163 FB\_READ\_171\_SNVT\_flow\_dir**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_flow\_dir.

**SNVT Nummer:** 171.

**Beschreibung:** Gangrichtung. Richtung des zu erlaubenden Gangflusses oder Richtung des aktuellen Gangflusses.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
eValue      : E_LON_flow_direction_t;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**eValue:** Enum, der empfangen werden soll (siehe [E\\_LON\\_flow\\_direction\\_t](#) [▶ 511]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

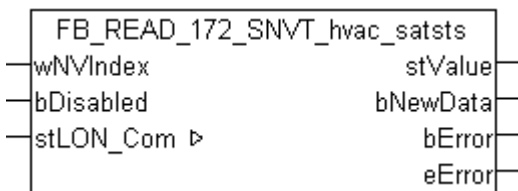
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com   : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.164 FB\_READ\_172\_SNVT\_hvac\_satsts**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_hvac\_satsts.

**SNVT Nummer:** 172.

**Beschreibung:** HLK Sättigungsstatus.

Eine 0 in einem Feld bedeutet, dass die mit dem Feld verbundene Anlage (Gerät) nicht gesättigt ist oder die Endabschaltung nicht erreicht bevor der gewünschte Sollwert erreicht ist.

Eine 1 in einem Feld bedeutet, dass die mit dem Feld verbundene Anlage (Gerät) gesättigt ist oder die Endabschaltung erreicht bevor der gewünschte Sollwert nicht erreicht ist.

**VAR\_INPUT**

```
wNVIndex    : WORD;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue     : ST_LON_SNVT_hvac_satsts;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_hvac\\_satsts](#) [▶ 575]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

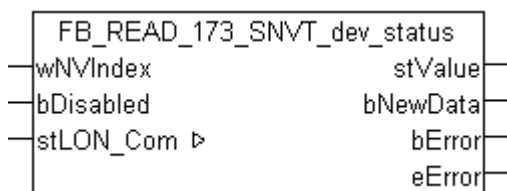
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.165 FB\_READ\_173\_SNVT\_dev\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_dev\_status.

**SNVT Nummer:** 173.

**Beschreibung:** Gerätestatus.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_dev_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_dev\\_status](#) [▶ 571]).

Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0), dann ist die Struktur *stValue.stDev\_type.stPump\_ctrl* gültig.

Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1), dann ist die Struktur *stValue.stDev\_type.stValvePos* gültig.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

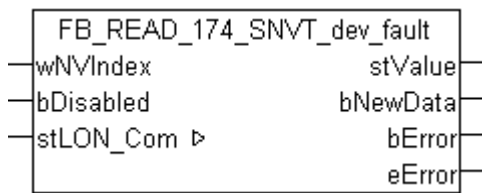
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.166 FB\_READ\_174\_SNVT\_dev\_fault



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_dev\_fault.

**SNVT Nummer:** 174.

**Beschreibung:** Fehlerstatus. Fehlerinformation für ein Gerät.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_dev_fault;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_dev\\_fault \[► 570\]](#)).

Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0), dann ist die Struktur *stValue.stDev\_type.stPump\_ctrl* gültig.

Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1), dann ist die Struktur *stValue.stDev\_type.stValvePos* gültig.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

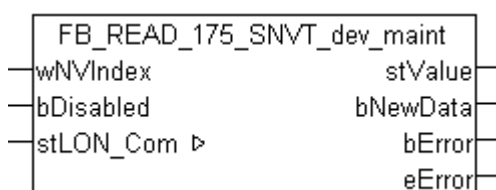
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

## 7.2.167 FB\_READ\_175\_SNVT\_dev\_maint



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):



**SNVT Name:** SNVT\_dev\_maint.

**SNVT Nummer:** 175.

**Beschreibung:** Geräte Wartungsstatus.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_dev_maint;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_dev\\_maint \[► 571\]](#)).

Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0) ist die Struktur

*stValue.stDev\_type.stPump\_ctrl* gültig.

Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1) ist die Struktur *stValue.stDev\_type.stValvePos* gültig.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

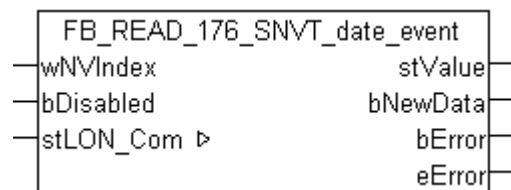
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

**7.2.168 FB\_READ\_176\_SNVT\_date\_event**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_date\_event.

**SNVT Nummer:** 176.

**Beschreibung:** Status eines Ereignisses.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_date_event;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_date\\_event \[▶ 570\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

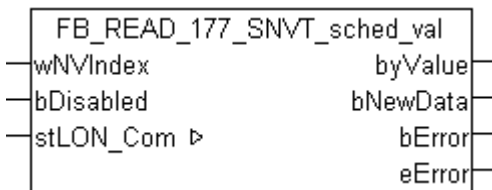
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.169 FB\_READ\_177\_SNVT\_sched\_val**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_sched\_val.

**SNVT Nummer:** 177.

**Beschreibung:** Kalenderwert. Index eines Kalenderwerts, dass einen Eintrag in einem SCPT Wert Definitionsfeld Array wählt oder ein direkter Wertausgang ist.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
byValue      : BYTE;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**byValue:** Min: 0 / Max: 255.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

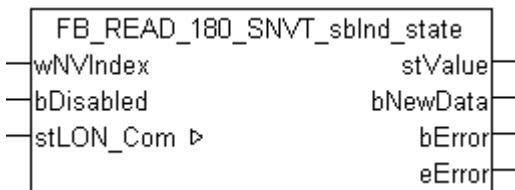
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.170 FB\_READ\_180\_SNVT\_sblnd\_state**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_sblnd\_state.

**SNVT Nummer:** 180.

**Beschreibung:** Jalousie Status.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_sblnd_state;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_sblnd\\_state \[▶ 585\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

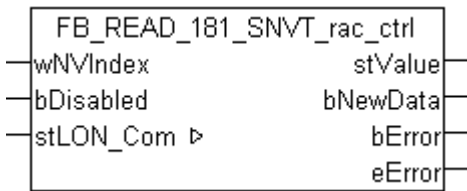
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.171 FB\_READ\_181\_SNVT\_rac\_ctrl



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_rac\_ctrl.

**SNVT Nummer:** 181.

**Beschreibung:** Steuerung Soundfunktion. Aktiviert die Soundfunktion einer bestimmten Quelle.

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_rac_ctrl;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_rac\\_ctrl \[▶ 584\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

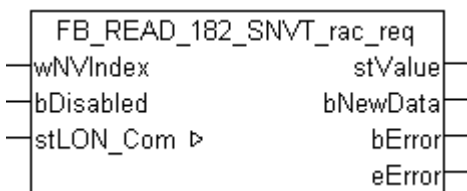
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.172 FB\_READ\_182\_SNVT\_rac\_req



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_rac\_req.

**SNVT Nummer:** 182.

**Beschreibung:** Anforderung Soundfunktion. Fordert die Soundfunktion einer bestimmten Quelle an.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_rac_req;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_rac\\_req \[▶ 584\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

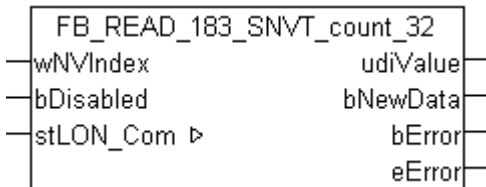
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.173 FB\_READ\_183\_SNVT\_count\_32**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_count\_32.

**SNVT Nummer:** 183.

**Beschreibung:** Absoluter Zähler. 32-bit Zähler.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
udiValue      : UDINT;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**udiValue:** Min: 0 / Max: 4294967294.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

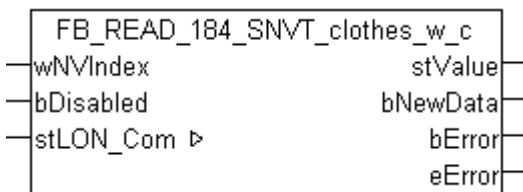
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.174 FB\_READ\_184\_SNVT\_clothes\_w\_c**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_clothes\_w\_c.

**SNVT Nummer:** 184.

**Beschreibung:** Waschmaschine / Kommandos. Zum Programmieren und Starten einer Waschmaschine.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_clothes_w_c;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_c \[▶ 568\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

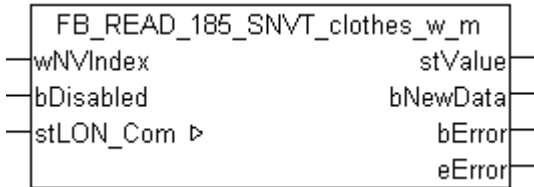
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.175 FB\_READ\_185\_SNVT\_clothes\_w\_m**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_clothes\_w\_m.

**SNVT Nummer:** 185.

**Beschreibung:** Waschmaschine / Management-Status. Aktueller Status von Tür / Deckel und Abfluss.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_clothes_w_m;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_m](#) [▶ 568]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

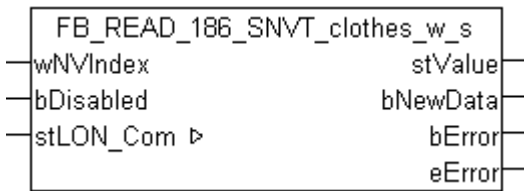
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.176 FB\_READ\_186\_SNVT\_clothes\_w\_s



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_clothes\_w\_s.

**SNVT Nummer:** 186.

**Beschreibung:** Waschmaschine / Status. Aktueller Zustand einer Waschmaschine, einschließlich Befehls- und Alarm-Informationen.

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_clothes_w_s;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_s \[▶ 569\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

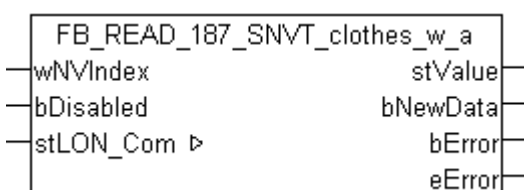
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

### 7.2.177 FB\_READ\_187\_SNVT\_clothes\_w\_a



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_clothes\_w\_a.



**SNVT Nummer:** 187.

**Beschreibung:** Waschmaschine / Alarmmeldungen.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_clothes_w_a;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_a \[▶ 566\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

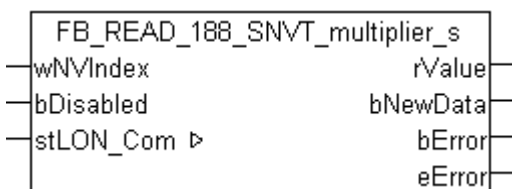
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.178 FB\_READ\_188\_SNVT\_multiplier\_s**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_multiplier\_s.

**SNVT Nummer:** 188.

**Beschreibung:** Multiplikator.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
rValue      : REAL;
bNewData    : BOOL;
bError      : BOOL;
eError      : E_LON_ERROR;
```

**rValue:** Min: 0 / Max: 2.54.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

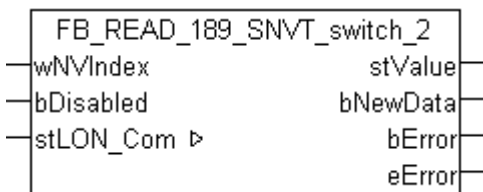
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.179 FB\_READ\_189\_SNVT\_switch\_2**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_switch\_2.

**SNVT Nummer:** 189.

**Beschreibung:** Schalter zum Einstellen von Szenen und Einstellungen. Eine erweiterte Version des SNVT\_switch zur Einstellung von Szenen und Einstellungen ähnlich den SNVT\_scene und SNVT\_setting.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_switch_2;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_switch\\_2 \[▶ 587\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

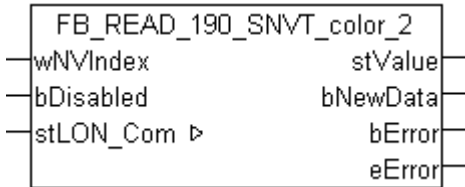
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.180 FB\_READ\_190\_SNVT\_color\_2**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_color\_2.

**SNVT Nummer:** 190.

**Beschreibung:** Farbe.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_color_2;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_color\\_2](#) [▶ 569]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

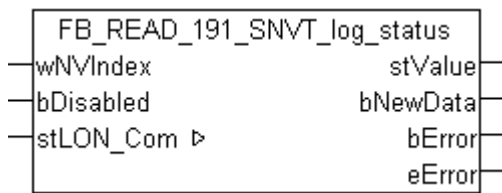
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.181 FB\_READ\_191\_SNVT\_log\_status



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_log\_status.

**SNVT Nummer:** 191.

**Beschreibung:** Log Status (hundertstel Sekunde). Stellt den aktuellen Status eines Datenloggings dar. Update basiert auf dem cpLogNotificationThreshold Wert. Zeigt nur den Status an; Alarme werden über das Node Object nvoAlarm2 ausgegeben. Benötigt, wenn das Node Object keinen nvoLogStat Ausgang enthält.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_log_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_log\\_status](#) [▶ 578]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

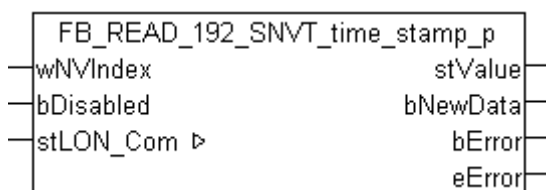
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.182 FB\_READ\_192\_SNVT\_time\_stamp\_p



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_stamp\_p.

**SNVT Nummer:** 192.

**Beschreibung:** Präziser Zeitstempel (Sekunden). Zeitstempel mit einer Auflösung einer hundertstel Sekunde.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : Timestruct;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [Timestruct](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

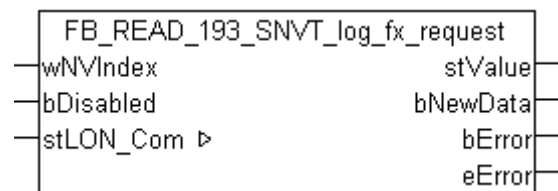
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.183 FB\_READ\_193\_SNVT\_log\_fx\_request**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_log\_fx\_request.

**SNVT Nummer:** 193.

**Beschreibung:** Log file transfer Anforderung. Fordert einen Datenlog per FTP Transfer an. Muss auf eine Standard FTP Anfrage folgen, um ein Datenlogfile zu erhalten. Erforderlich auf Geräten, die das Datenlogger Funktionsprofil nutzen, das den Datenlog Transfer über FTP erlaubt.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_log_fx_request;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_log\\_fx\\_request \[▶ 577\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

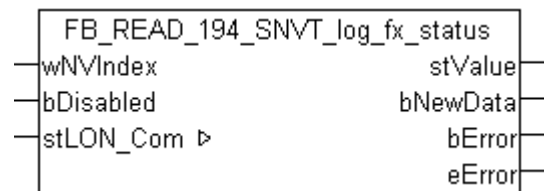
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

**7.2.184 FB\_READ\_194\_SNVT\_log\_fx\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_log\_fx\_status.

**SNVT Nummer:** 194.

**Beschreibung:** Log file transfer Status. Zeigt den Status eines Datenlogs per FTP Transfer an. Erforderlich auf Geräten, die das Datenlogger Funktionsprofil nutzen, das den Datenlog Transfer über FTP erlaubt.

**VAR\_INPUT**

```
wNVIndex     : WORD;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue      : ST_LON_SNVT_log_fx_status;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_log\\_fx\\_status \[▶ 577\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

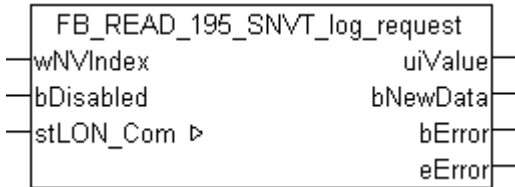
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

**7.2.185 FB\_READ\_195\_SNVT\_log\_request**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_log\_request.

**SNVT Nummer:** 195.

**Beschreibung:** Log Status Anforderung. Fordert den aktuellen Status eines Datenlogs an. Der Status wird über das SNVT\_log\_status output ausgegeben.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue : UINT;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

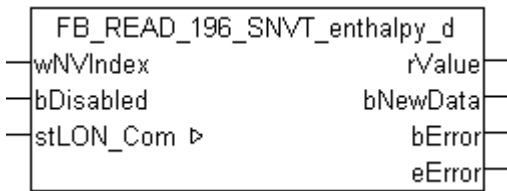
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

### 7.2.186 FB\_READ\_196\_SNVT\_enthalpy\_d



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_enthalpy\_d.

**SNVT Nummer:** 196.

**Beschreibung:** Enthalpie Unterschied(kJ/kg).

#### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
rValue       : REAL;
bNewData     : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
```

**rValue:** Min: -327.68 / Max: 327.66.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

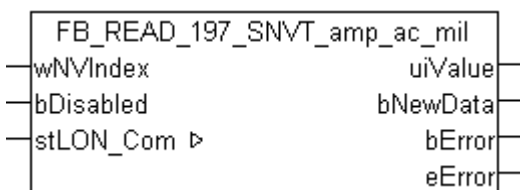
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

### 7.2.187 FB\_READ\_197\_SNVT\_amp\_ac\_mil



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_amp\_ac\_mil.

**SNVT Nummer:** 197.



**Beschreibung:** Elektrischer Strom (Milliampere).

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
uiValue       : UINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**uiValue:** Min: 0 / Max: 65535.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

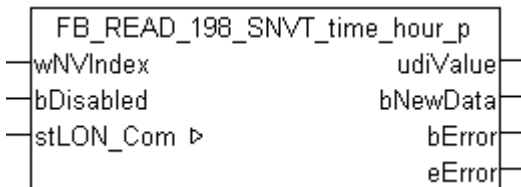
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.188 FB\_READ\_198\_SNVT\_time\_hour\_p**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_time\_hour\_p.

**SNVT Nummer:** 198.

**Beschreibung:** Zeit in Stunden.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
udiValue       : UDINT;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**udiValue:** Min: 0 / Max: 4294967294.

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

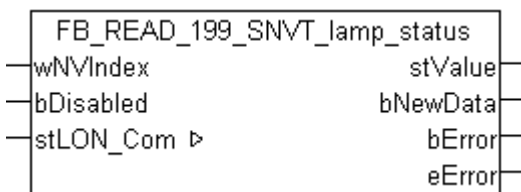
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.189 FB\_READ\_199\_SNVT\_lamp\_status**



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_lamp\_status.

**SNVT Nummer:** 199.

**Beschreibung:** Lampen Status.

**VAR\_INPUT**

```
wNVIndex : WORD;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue : ST_LON_SNVT_lamp_status;
bNewData : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_lamp\\_status](#) [▶ 577]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

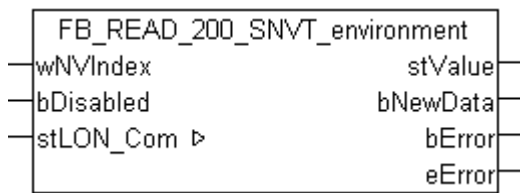
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.190 FB\_READ\_200\_SNVT\_environment



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_environment.

**SNVT Nummer:** 200.

**Beschreibung:** Umgebung.

### VAR\_INPUT

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
stValue       : ST_LON_SNVT_environment;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_environment \[▶ 573\]](#)).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

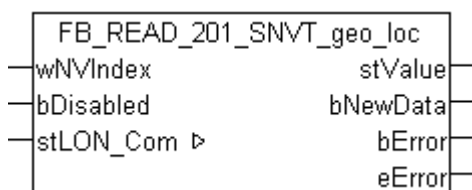
**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.191 FB\_READ\_201\_SNVT\_geo\_loc



Dieser Baustein empfängt folgende LON Eingangsvariable (nvi):

**SNVT Name:** SNVT\_geo\_loc.

**SNVT Nummer:** 201.

**Beschreibung:** Geographische Lage.

**VAR\_INPUT**

```
wNVIndex      : WORD;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
stValue       : ST_LON_SNVT_geo_loc;
bNewData      : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
```

**stValue:** Struktur, die empfangen werden soll (siehe [ST\\_LON\\_SNVT\\_geo\\_loc](#) [▶ 575]).

**bNewData:** Wird für 1 Zyklus TRUE, wenn der Baustein Daten empfangen hat.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.192 FB\_SEND\_001\_SNVT\_amp**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_amp.

**SNVT Nummer:** 001.

**Beschreibung:** Elektrischer Strom (Ampere).

**VAR\_INPUT**

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

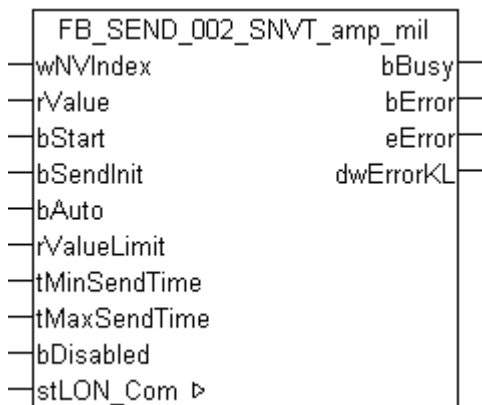
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.193 FB\_SEND\_002\_SNVT\_amp\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_amp\_mil.

**SNVT Nummer:** 002.

**Beschreibung:** Elektrischer Strom (Milliampere).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

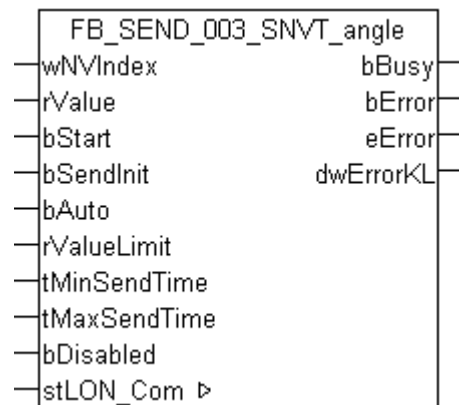
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.194 FB\_SEND\_003\_SNVT\_angle



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_angle.

**SNVT Nummer:** 003.

**Beschreibung:** Winkelabstand (Bogenmaß).

### VAR\_INPUT

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 65.535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

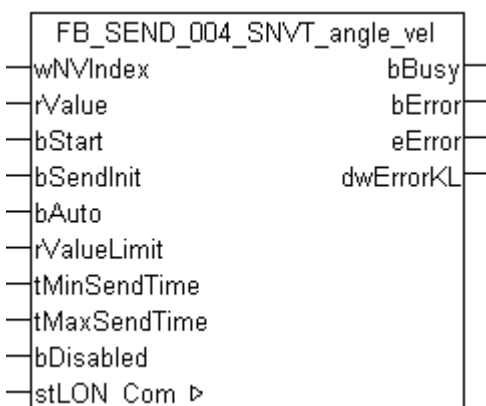
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.195 FB\_SEND\_004\_SNVT\_angle\_vel**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_angle\_vel.

**SNVT Nummer:** 004.

**Beschreibung:** Winkelgeschwindigkeit (Bogenmaß / Sekunde).



## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.196 FB\_SEND\_005\_SNVT\_btu\_kilo



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_btu\_kilo.

**SNVT Nummer:** 005.

**Beschreibung:** Thermische Energie (Kilo BTU).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

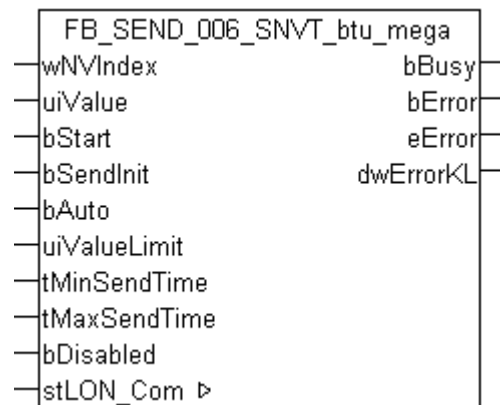
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.197 FB\_SEND\_006\_SNVT\_btu\_mega



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_btu\_mega.

**SNVT Nummer:** 006.

**Beschreibung:** Thermische Energie (Mega BTU).

### VAR\_INPUT

```
wNVIndex : WORD;
uiValue : UINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.198 FB\_SEND\_007\_SNVT\_char\_ascii**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_char\_ascii.

**SNVT Nummer:** 007.

**Beschreibung:** ASCII Zeichen (8-bit ASCII character).

## VAR\_INPUT

```
wNVIndex      : WORD;
sValue        : STRING(1);
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**sValue:** STRING(1).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

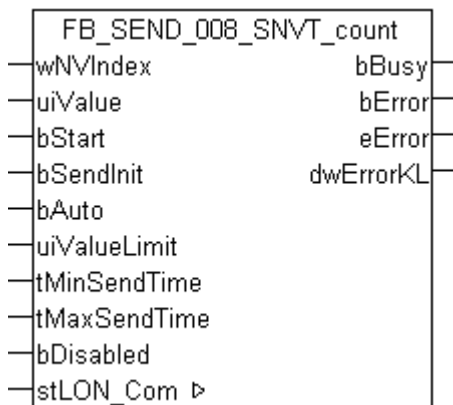
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.199 FB\_SEND\_008\_SNVT\_count



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_count.

**SNVT Nummer:** 008.

**Beschreibung:** Zähler (nur positive Werte).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

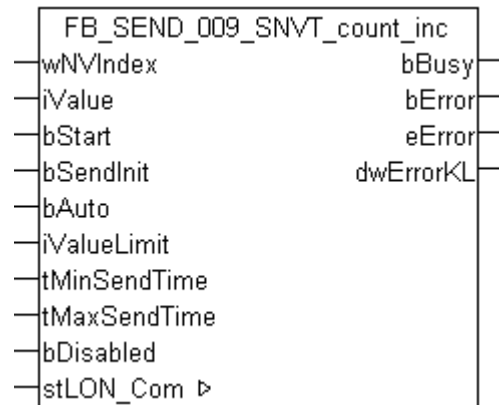
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.200 FB\_SEND\_009\_SNVT\_count\_inc**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_count\_inc.

**SNVT Nummer:** 009.

**Beschreibung:** Zähler (negative und positive Werte).

**VAR\_INPUT**

```
wNVIndex : WORD;
iValue : INT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
iValueLimit : INT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**iValue:** Min: -32768 / Max: 32767.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**iValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*iValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.201 FB\_SEND\_011\_SNVT\_date\_day**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_date\_day.

**SNVT Nummer:** 011.

**Beschreibung:** Wochentag.



## VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_days_of_week_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_days\\_of\\_week\\_t](#) [▶ 502]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

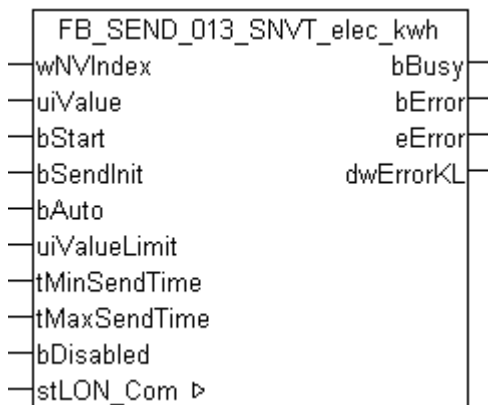
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.202 FB\_SEND\_013\_SNVT\_elec\_kwh



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_elec\_kwh.

**SNVT Nummer:** 013.

**Beschreibung:** Elektrische Energie (kW/h).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

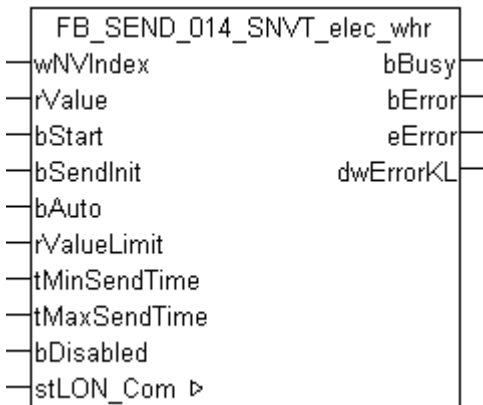
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.203 FB\_SEND\_014\_SNVT\_elec\_whr**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_elec\_whr.

**SNVT Nummer:** 014.

**Beschreibung:** Elektrische Energie (W/h).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

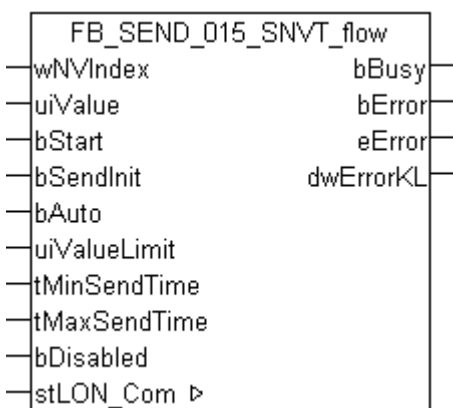
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.204 FB\_SEND\_015\_SNVT\_flow**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_flow.

**SNVT Nummer:** 015.

**Beschreibung:** Volumenstrom (Liter / Sekunde).

## VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.205 FB\_SEND\_016\_SNVT\_flow\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_flow\_mil.

**SNVT Nummer:** 016.

**Beschreibung:** Volumenstrom (Milliliter / Sekunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

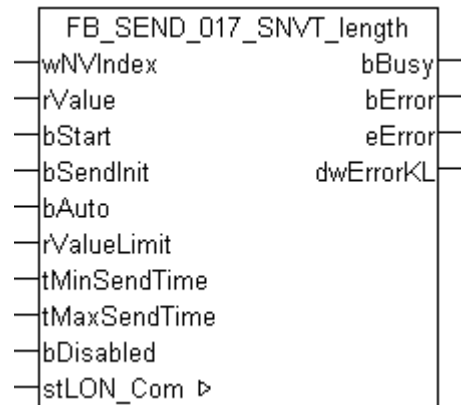
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.206 FB\_SEND\_017\_SNVT\_length**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_length.

**SNVT Nummer:** 017.

**Beschreibung:** Länge (Meter).

**VAR\_INPUT**

```

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

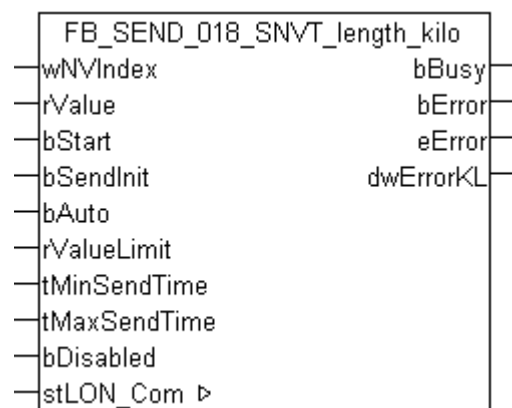
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.207 FB\_SEND\_018\_SNVT\_length\_kilo**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_length\_kilo.

**SNVT Nummer:** 018.

**Beschreibung:** Länge (Kilometer).



## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

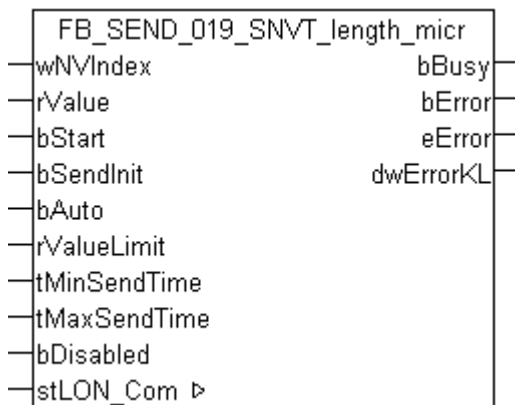
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.208 FB\_SEND\_019\_SNVT\_length\_micr



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_length\_micr.

**SNVT Nummer:** 019.

**Beschreibung:** Länge (Mikrometer).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der bBusy Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable "eError" beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird "bError" TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401](#) [▶ 68]. Die Variable "eError" hat in diesem Fall den Wert "eKL6401\_Error" (17). Gleichzeitig wird "bError" TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.209 FB\_SEND\_020\_SNVT\_length\_mil**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_length\_mil.

**SNVT Nummer:** 020.

**Beschreibung:** Länge (Millimeter).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.210 FB\_SEND\_021\_SNVT\_lev\_cont**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_lev\_cont.

**SNVT Nummer:** 021.

**Beschreibung:** Stufenlose Wertänderung (% vom obersten Niveau).

## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 100.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

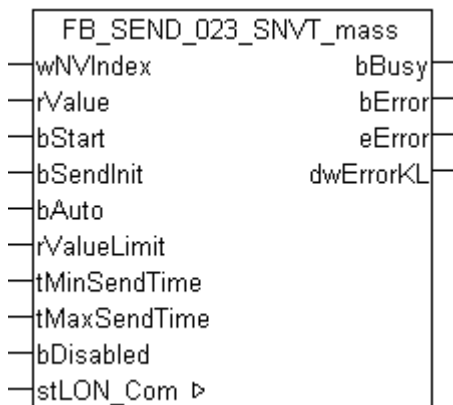
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.211 FB\_SEND\_023\_SNVT\_mass



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_mass.

**SNVT Nummer:** 023.

**Beschreibung:** Masse (Gramm).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

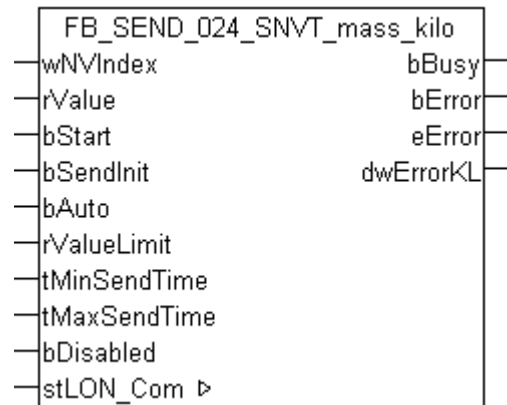
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.212 FB\_SEND\_024\_SNVT\_mass\_kilo**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_mass\_kilo.

**SNVT Nummer:** 024.

**Beschreibung:** Masse (Kilogramm).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

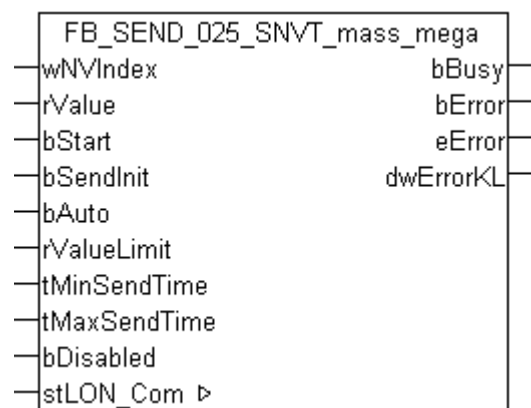
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.213 FB\_SEND\_025\_SNVT\_mass\_mega**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_mass\_mega.

**SNVT Nummer:** 025.

**Beschreibung:** Masse (Metrische Tonne).



## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.214 FB\_SEND\_026\_SNVT\_mass\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_mass\_mil.

**SNVT Nummer:** 026.

**Beschreibung:** Masse (Milligramm).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.215 FB\_SEND\_027\_SNVT\_power**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_power.

**SNVT Nummer:** 027.

**Beschreibung:** Leistung (Watt).

**VAR\_INPUT**

```

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.216 FB\_SEND\_028\_SNVT\_power\_kilo**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_power\_kilo.

**SNVT Nummer:** 028.

**Beschreibung:** Leistung (Kilowatt).

## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

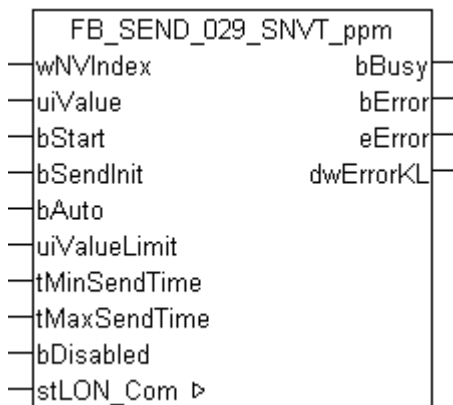
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.217 FB\_SEND\_029\_SNVT\_ppm



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ppm.

**SNVT Nummer:** 029.

**Beschreibung:** Konzentration (ppm).

### VAR\_INPUT

```

wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

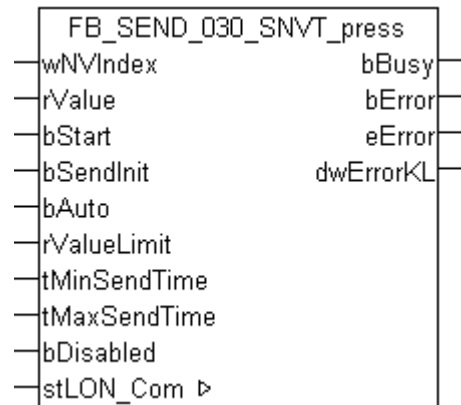
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.218 FB\_SEND\_030\_SNVT\_press**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_press.

**SNVT Nummer:** 030.

**Beschreibung:** Druck (Überdruck) (Kilopascal).

**VAR\_INPUT**

```

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

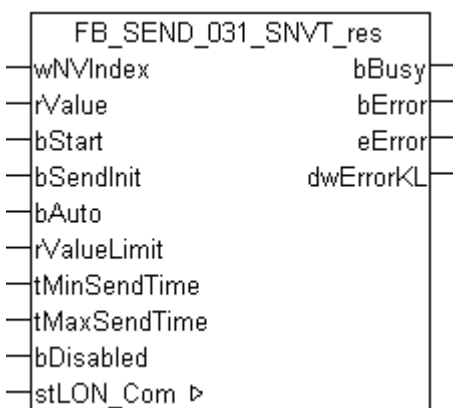
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.219 FB\_SEND\_031\_SNVT\_res**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_res.

**SNVT Nummer:** 031.

**Beschreibung:** Elektrischer Widerstand (Ohm).



## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

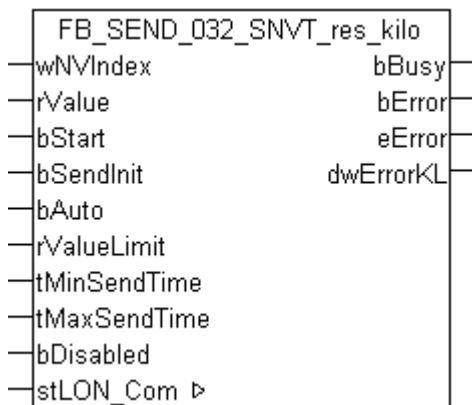
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.220 FB\_SEND\_032\_SNVT\_res\_kilo



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_res\_kilo.

**SNVT Nummer:** 032.

**Beschreibung:** Elektrischer Widerstand (Kilohm).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

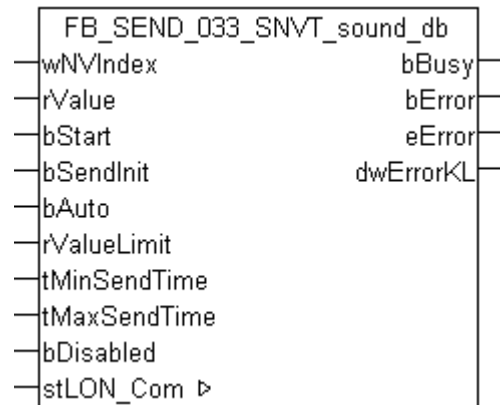
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.221 FB\_SEND\_033\_SNVT\_sound\_db**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_sound\_db.

**SNVT Nummer:** 033.

**Beschreibung:** Schalldruckpegel (dB).

**VAR\_INPUT**

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

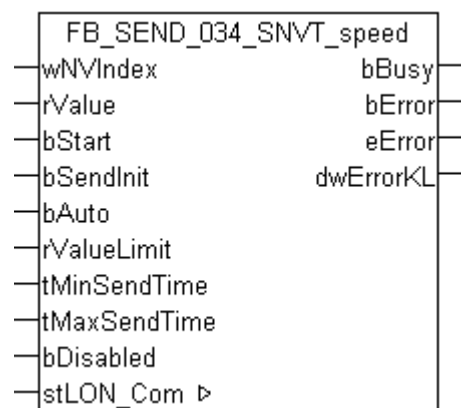
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.222 FB\_SEND\_034\_SNVT\_speed**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_speed.

**SNVT Nummer:** 034.

**Beschreibung:** Lineare Geschwindigkeit (Meter / Sekunde).

## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.223 FB\_SEND\_035\_SNVT\_speed\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_speed\_mil.

**SNVT Nummer:** 035.

**Beschreibung:** Lineare Geschwindigkeit (Millimeter / Sekunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 65.535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

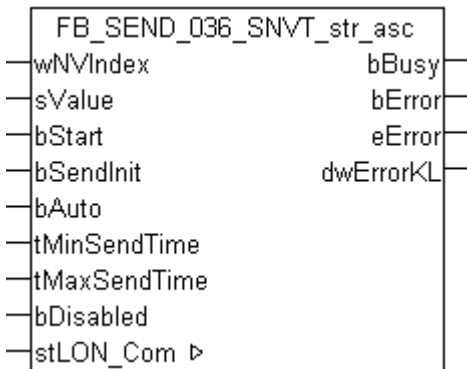
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.224 FB\_SEND\_036\_SNVT\_str\_asc**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_str\_asc.

**SNVT Nummer:** 036.

**Beschreibung:** Zeichenkette (30 Zeichen max) (ASCII-Zeichenkette).

**VAR\_INPUT**

```
wNVIndex : WORD;
sValue : STRING(31);
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**sValue:** STRING(31).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.225 FB\_SEND\_037\_SNVT\_str\_int**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_str\_int.

**SNVT Nummer:** 037.

**Beschreibung:** Wide Character String mit eigenem Code (Max. 15 Zeichen) (Wide character string).

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_str_int;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.



**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_str\\_int](#) [▶ 587]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

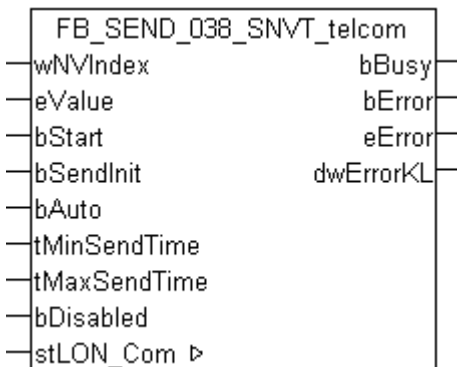
**dwErrorKL:** [Fehlerkennung](#) [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.226 FB\_SEND\_038\_SNVT\_telcom**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_telcom.

**SNVT Nummer:** 038.

**Beschreibung:** Telefonstatus (Telefonstatus Namen).

**VAR\_INPUT**

```

wNVIndex      : WORD;
eValue        : E_LON_telcom_states_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_telcom\\_states\\_t](#) [▶ 533]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

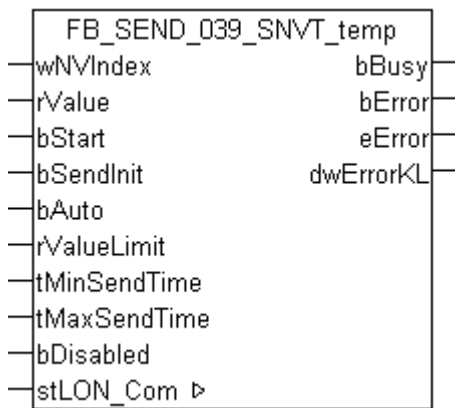
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.227 FB\_SEND\_039\_SNVT\_temp



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp.

**SNVT Nummer:** 039.

**Beschreibung:** Temperatur (Grad Celsius).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -274 / Max: 6279.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

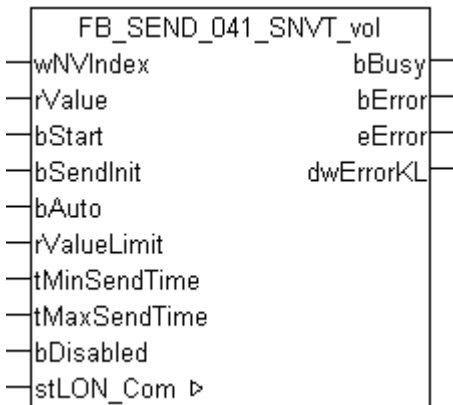
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.228 FB\_SEND\_041\_SNVT\_vol**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_vol.

**SNVT Nummer:** 041.

**Beschreibung:** Volumen (Liter).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

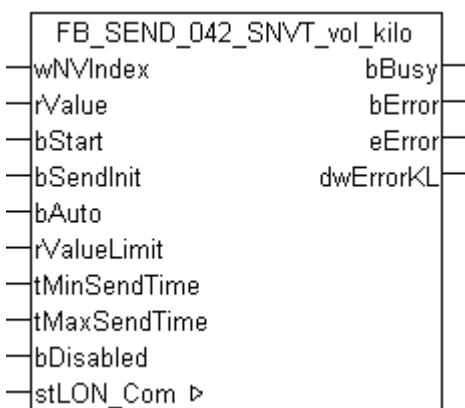
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.229 FB\_SEND\_042\_SNVT\_vol\_kilo**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_vol\_kilo.

**SNVT Nummer:** 042.

**Beschreibung:** Volumen (Kiloliter).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

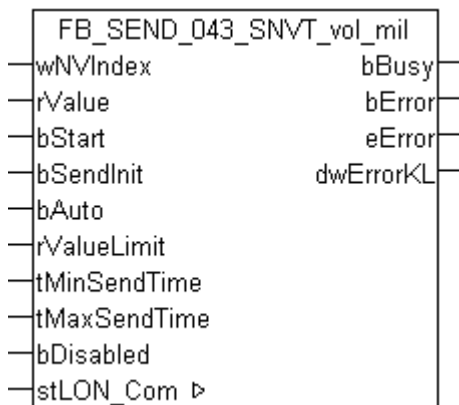
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.230 FB\_SEND\_043\_SNVT\_vol\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_vol\_mil.

**SNVT Nummer:** 043.

**Beschreibung:** Volumen (Milliliter).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.231 FB\_SEND\_044\_SNVT\_volt**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt.

**SNVT Nummer:** 044.

**Beschreibung:** Elektrische Spannung (Volt).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.232 FB\_SEND\_045\_SNVT\_volt\_dbmv**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt\_dbmv.

**SNVT Nummer:** 045.

**Beschreibung:** Elektrische Spannung (dB Mikrovolt).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

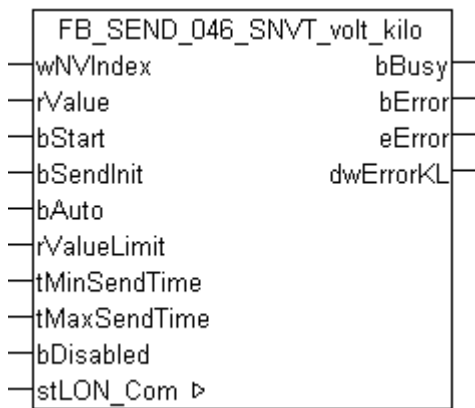
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.233 FB\_SEND\_046\_SNVT\_volt\_kilo



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt\_kilo.

**SNVT Nummer:** 046.

**Beschreibung:** Elektrische Spannung (Kilovolt).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.234 FB\_SEND\_047\_SNVT\_volt\_mil**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt\_mil.

**SNVT Nummer:** 047.

**Beschreibung:** Elektrische Spannung (Millivolt).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3276.8 / Max: 3276.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.235 FB\_SEND\_048\_SNVT\_amp\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_amp\_f.

**SNVT Nummer:** 048.

**Beschreibung:** Elektrischer Strom (Ampere).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.236 FB\_SEND\_049\_SNVT\_angle\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_angle\_f.

**SNVT Nummer:** 049.

**Beschreibung:** Winkelabstand (Bogenmaß).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

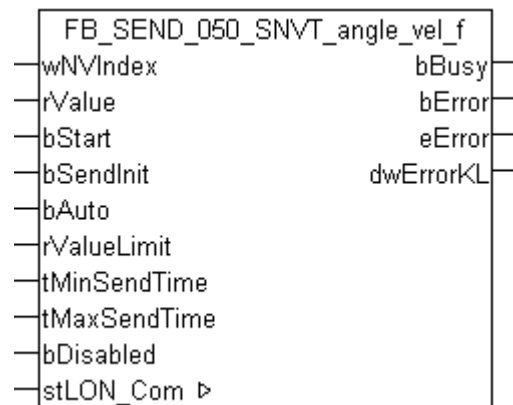
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.237 FB\_SEND\_050\_SNVT\_angle\_vel\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_angle\_vel\_f.

**SNVT Nummer:** 050.

**Beschreibung:** Winkelgeschwindigkeit (Radiant / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

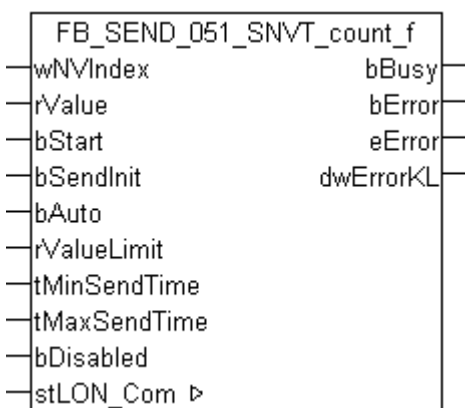
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.238 FB\_SEND\_051\_SNVT\_count\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_count\_f.

**SNVT Nummer:** 051.

**Beschreibung:** Absolute Anzahl (Stück).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

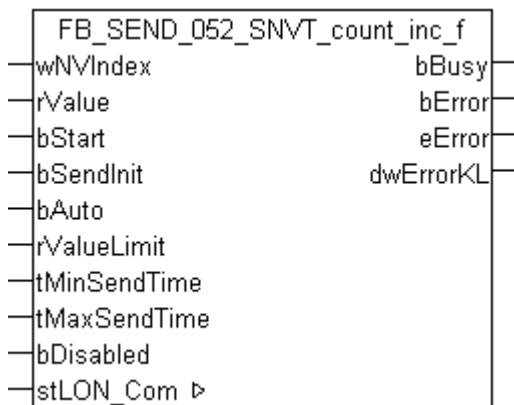
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.239 FB\_SEND\_052\_SNVT\_count\_inc\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_count\_inc\_f.

**SNVT Nummer:** 052.

**Beschreibung:** Inkrement Zähler (Stück (delta)).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

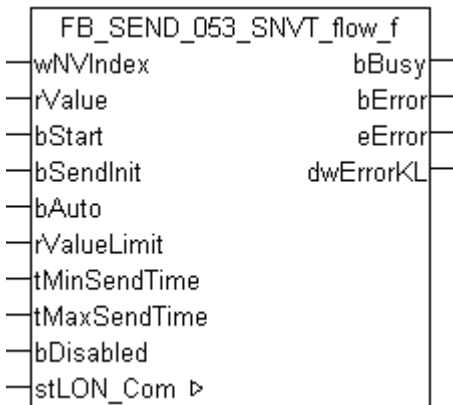
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.240 FB\_SEND\_053\_SNVT\_flow\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_flow\_f.

**SNVT Nummer:** 053.

**Beschreibung:** Volumenstrom (Liter / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

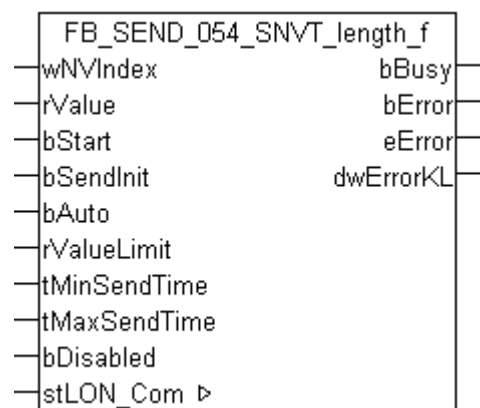
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.241 FB\_SEND\_054\_SNVT\_length\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_length\_f.

**SNVT Nummer:** 054.

**Beschreibung:** Länge (Meter).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

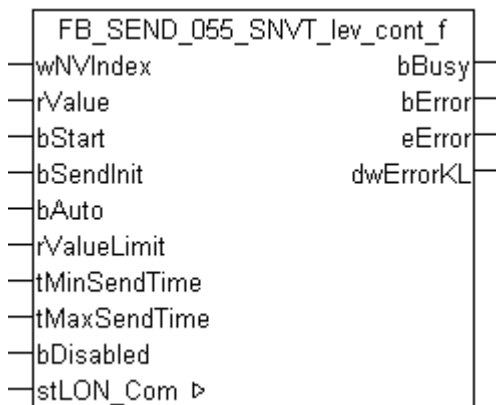
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.242 FB\_SEND\_055\_SNVT\_lev\_cont\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_lev\_cont\_f.

**SNVT Nummer:** 055.

**Beschreibung:** Stufenlose Wertänderung (% vom obersten Niveau).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 100.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.243 FB\_SEND\_056\_SNVT\_mass\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_mass\_f.

**SNVT Nummer:** 056.

**Beschreibung:** Masse (Gramm).

**VAR\_INPUT**

```

wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

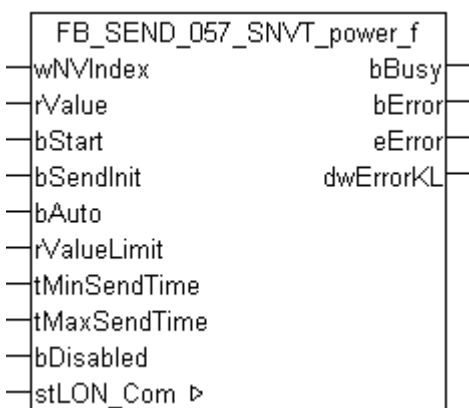
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.244 FB\_SEND\_057\_SNVT\_power\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_power\_f.

**SNVT Nummer:** 057.

**Beschreibung:** Leistung (Watt).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.245 FB\_SEND\_058\_SNVT\_ppm\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ppm\_f.

**SNVT Nummer:** 058.

**Beschreibung:** Konzentration (ppm).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

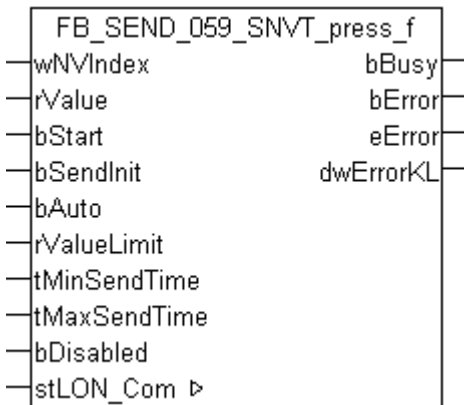
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.246 FB\_SEND\_059\_SNVT\_press\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_press\_f.

**SNVT Nummer:** 059.

**Beschreibung:** Druck (Überdruck) (Pascal).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.247 FB\_SEND\_060\_SNVT\_res\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_res\_f.

**SNVT Nummer:** 060.

**Beschreibung:** Elektrischer Widerstand (Ohm).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

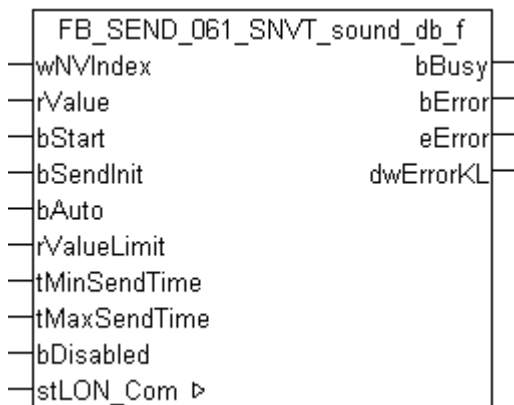
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.248 FB\_SEND\_061\_SNVT\_sound\_db\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_sound\_db\_f.

**SNVT Nummer:** 061.

**Beschreibung:** Schalldruckpegel (dBspl).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.249 FB\_SEND\_062\_SNVT\_speed\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_speed\_f.

**SNVT Nummer:** 062.

**Beschreibung:** Lineare Geschwindigkeit (Meter / Sekunde).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

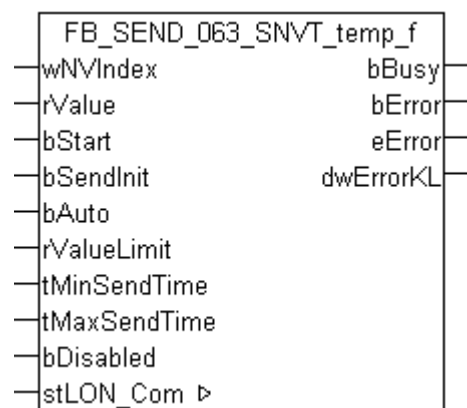
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.250 FB\_SEND\_063\_SNVT\_temp\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp\_f.

**SNVT Nummer:** 063.

**Beschreibung:** Temperatur (Grad Celsius).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -273,17 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

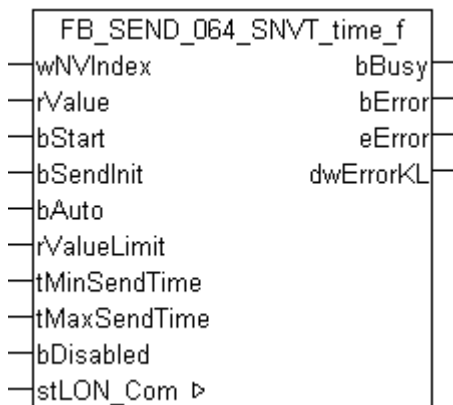
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.251 FB\_SEND\_064\_SNVT\_time\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_f.

**SNVT Nummer:** 064.

**Beschreibung:** Abgelaufene Zeit (Sekunden).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

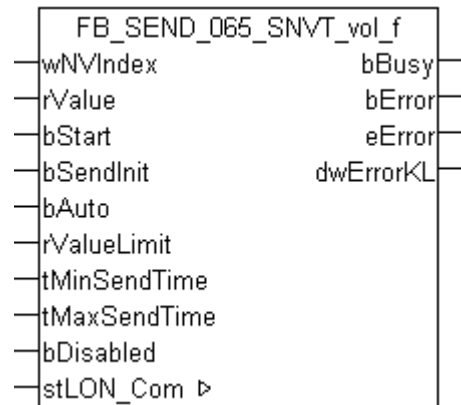
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.252 FB\_SEND\_065\_SNVT\_vol\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_vol\_f.

**SNVT Nummer:** 065.

**Beschreibung:** Volumen (Liter).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.253 FB\_SEND\_066\_SNVT\_volt\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt\_f.

**SNVT Nummer:** 066.

**Beschreibung:** Elektrische Spannung (Volt).

**VAR\_INPUT**

```

wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.254 FB\_SEND\_067\_SNVT\_btu\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_btu\_f.

**SNVT Nummer:** 067.

**Beschreibung:** Thermische Energie (BTU).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

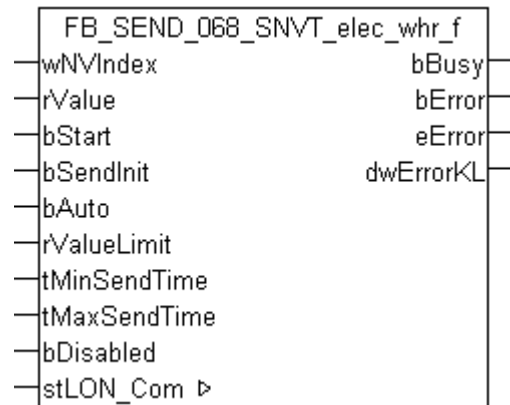
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.255 FB\_SEND\_068\_SNVT\_elec\_whr\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_elec\_whr\_f.

**SNVT Nummer:** 068.

**Beschreibung:** Elektrische Energie (Watt / Stunden).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.256 FB\_SEND\_069\_SNVT\_config\_src**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_config\_src.

**SNVT Nummer:** 069.

**Beschreibung:** Konfigurationseigenschaften (Name der Konfigurationsquelle (0=eigen, 1=fremd)).

**VAR\_INPUT**

```

wNVIndex      : WORD;
eValue        : E_LON_config_source_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_config\\_source\\_t](#) [▶ 499]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

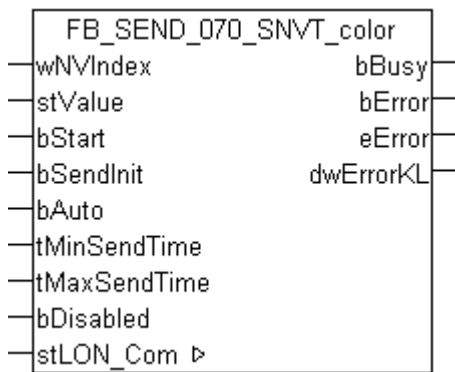
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.257 FB\_SEND\_070\_SNVT\_color



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_color.

**SNVT Nummer:** 070.

**Beschreibung:** Farbe nach CIE Norm (L\*,a\*,b).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_color;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_color](#) [► 569]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

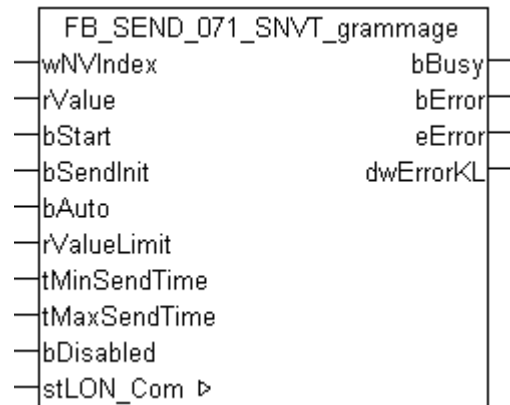
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.258 FB\_SEND\_071\_SNVT\_grammage**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_grammage.

**SNVT Nummer:** 071.

**Beschreibung:** amerikanisches Mass für Papiergewichte und Dichte (Gramm/m<sup>2</sup>).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

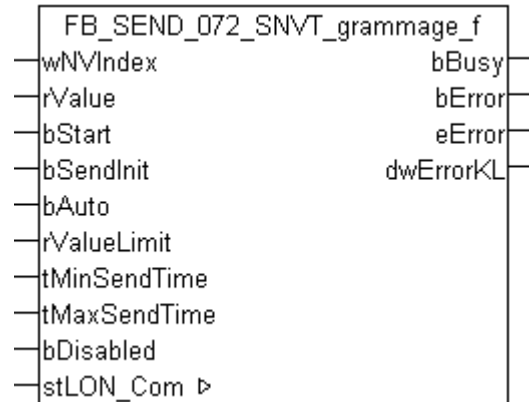
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.259 FB\_SEND\_072\_SNVT\_grammage\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_grammage\_f.

**SNVT Nummer:** 072.

**Beschreibung:** amerikanisches Mass für Papiergewichte und Dichte (Gramm/m2).

**VAR\_INPUT**

```
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

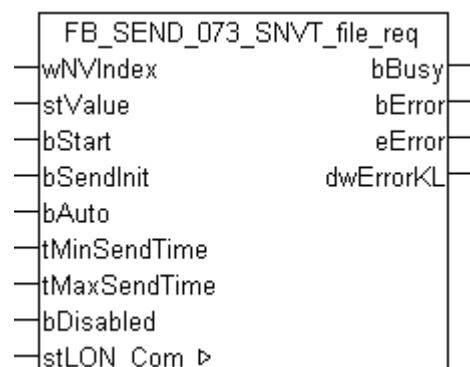
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.260 FB\_SEND\_073\_SNVT\_file\_req



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_file\_req.

**SNVT Nummer:** 073.

**Beschreibung:** Dateianforderung.

#### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_file_req;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_file\\_req \[▶ 574\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[▶ 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[▶ 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[▶ 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

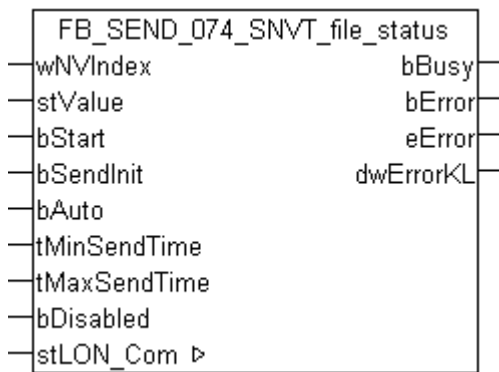
**dwErrorKL:** Fehlerkennung [\[▶ 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.261 FB\_SEND\_074\_SNVT\_file\_status



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_file\_status.

**SNVT Nummer:** 074.

**Beschreibung:** Dateiinformation (Teil des LONWORKS File-Transfer Protokoll (LW-FTP)).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_file_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_file\\_status \[► 575\]](#)). Wenn "stValue.eStatus" = eLON\_FS\_LOOKUP\_OK (1) dann wird die Struktur "stValue.stAdr.stDescriptor" gesendet, ansonsten wird "stValue.stAdr.stAddress" gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.



**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.262 FB\_SEND\_075\_SNVF\_freq\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_freq\_f.

**SNVT Nummer:** 075.

**Beschreibung:** Frequenz (Hertz).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.263 FB\_SEND\_076\_SNVT\_freq\_hz**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_freq\_hz.

**SNVT Nummer:** 076.

**Beschreibung:** Frequenz (Hertz).

**VAR\_INPUT**

```
wNVIndex   : WORD;
rValue     : REAL;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

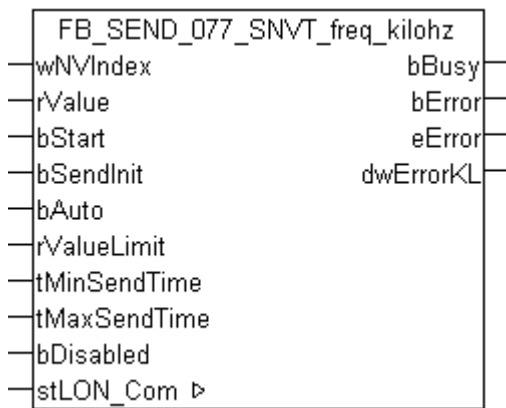
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.264 FB\_SEND\_077\_SNVT\_freq\_kilohz



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_freq\_kilohz.

**SNVT Nummer:** 077.

**Beschreibung:** Frequenz (Kilohertz).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

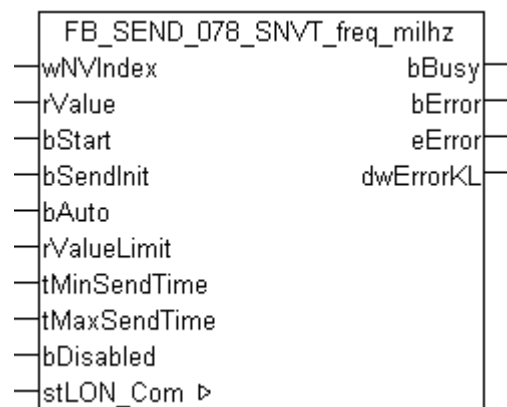
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.265 FB\_SEND\_078\_SNVT\_freq\_milhz**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_freq\_milhz.

**SNVT Nummer:** 078.

**Beschreibung:** Frequenz (Millihertz).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6.5535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

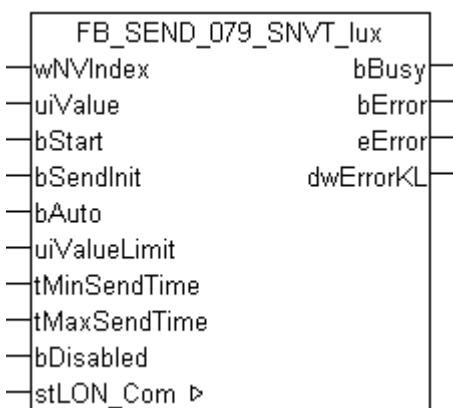
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.266 FB\_SEND\_079\_SNVT\_lux**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_lux.

**SNVT Nummer:** 079.

**Beschreibung:** Beleuchtungsstärke (Lux) 1 lux = 1 lumen/m².

## VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

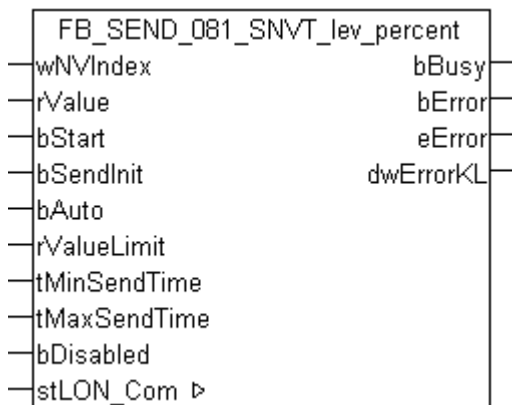
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.267 FB\_SEND\_081\_SNVT\_lev\_percent



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_lev\_percent.

**SNVT Nummer:** 081.

**Beschreibung:** Prozentangabe in 0,005-%-Schritten mit Vorzeichen. Anstelle von SNVT\_switch sollte SNVT\_lev\_percent benutzt werden. Ausnahmen bilden Netzwerkvariablen die einen Prozentwert übertragen und die von SNVT\_lev\_percent unterstützte zusätzlich Auflösung erfordern oder Netzwerkvariablen von Funktionsprofilen, die hauptsächlich für das Verknüpfen mit SNVT\_lev\_percent Variablen anderer Funktionsprofile gedacht sind.

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -163.84 / Max: 163.835.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.



**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

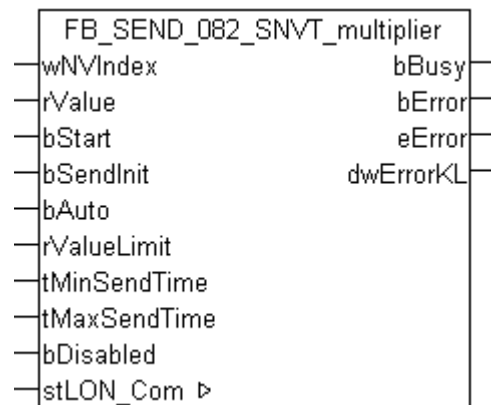
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.268 FB\_SEND\_082\_SNVT\_multiplier**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_multiplier.

**SNVT Nummer:** 082.

**Beschreibung:** Multiplikator in 0,0005- Schritten (16-bit Vorzeichenloser Wert).

**VAR\_INPUT**

```
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 32.7675.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

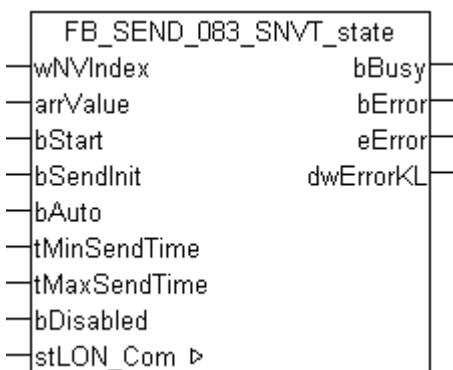
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.269 FB\_SEND\_083\_SNVT\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_state.

**SNVT Nummer:** 083.

**Beschreibung:** Statusinformation (16 individuelle Bitwerte). Jeder Status ist ein boolescher Wert. SNVT\_state\_64 bevorzugt.

### VAR\_INPUT

```
wNVIndex      : WORD;
arrValue      : ARRAY [0..15] OF BOOL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**arrValue:** 0-15 Bit.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

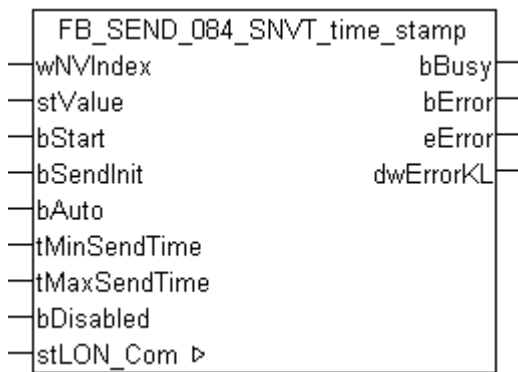
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.270 FB\_SEND\_084\_SNVT\_time\_stamp



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_stamp.

**SNVT Nummer:** 084.

**Beschreibung:** Zeitstempel (Jahr, Monat, Tag, Stunde, Minute, Sekunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : TIMESTRUCT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON-Knoten benötigt. Pro LON-Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [TIMESTRUCT](#)). Die Strukturvariablen *wDayOfWeek* und *wMilliseconds* sind hier nicht gültig und werden nicht übertragen.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.271 FB\_SEND\_085\_SNVT\_zerospan**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_zerospan.

**SNVT Nummer:** 085.

**Beschreibung:** Nullpunkt und Proportionalitätsfaktor. Lineare Transformationsparameter: Multiplizieren mit Proportionalitätsfaktor, dann addieren mit Nullpunkt Offset.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_zerospan;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_zerospan](#) [▶ 589]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

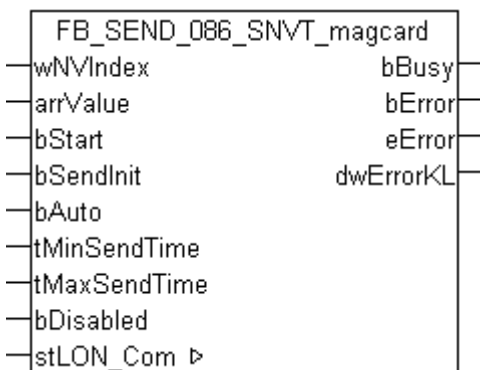
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.272 FB\_SEND\_086\_SNVT\_magcard**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_magcard.

**SNVT Nummer:** 086.

**Beschreibung:** ISO 7811 (40 Hexadezimal Zahlen). Daten entsprechend des ISO 7811 Standards für Magnetkartenleser.

**VAR\_INPUT**

```
wNVIndex    : WORD;
arrValue    : ARRAY [0..40] OF BYTE;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**arrValue:** 1-40 Byte.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.273 FB\_SEND\_087\_SNVT\_elapsed\_tm**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_elapsed\_tm.

**SNVT Nummer:** 087.

**Beschreibung:** Abgelaufene Zeit (Tag, Stunde, Minute, Sekunde, Millisekunde).

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_elapsed_tm;
bStart      : BOOL;
```

```
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_elapsed\\_tm](#) [▶ 572]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy : BOOL;
bError : BOOL;
eError : E_LON_ERROR;
dwErrorKL : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

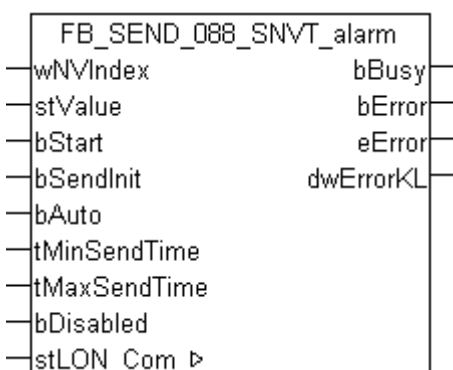
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.274 FB\_SEND\_088\_SNVT\_alarm**





Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_alarm

**SNVT Nummer:** 088

**Beschreibung:** Alarm Status

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_alarm;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_alarm](#) [▶ 565]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.275 FB\_SEND\_089\_SNVT\_currency



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_currency.

**SNVT Nummer:** 089.

**Beschreibung:** Währung (Einheit, Multiplikator, Wert).

### VAR\_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_currency;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_currency](#) [► 570]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
  
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

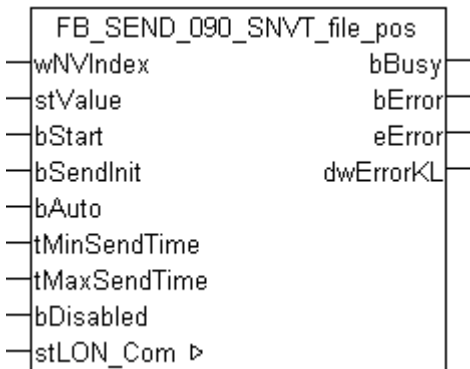
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.276 FB\_SEND\_090\_SNVT\_file\_pos**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_file\_pos.

**SNVT Nummer:** 090.

**Beschreibung:** Dateiposition (Pointer, Länge).

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_file_pos;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_file_pos` [▶ 574]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

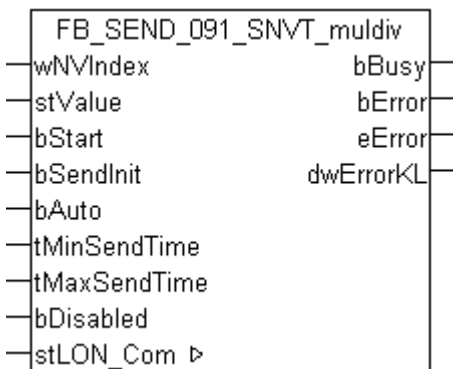
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Diese Struktur verbindet FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangs-Bausteinen (see ST\_LON\_Communication [▶ 562]).

**7.2.277 FB\_SEND\_091\_SNVT\_muldiv**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_muldiv.

**SNVT Nummer:** 091.

**Beschreibung:** Verstärkungsfaktor/Dämpfungsfaktor (Multiplikator, Divisor).

**VAR\_INPUT**

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_muldiv;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_muldiv [▶ 578]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.278 FB\_SEND\_092\_SNVT\_obj\_request**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_obj\_request.

**SNVT Nummer:** 092.

**Beschreibung:** Funktionsauswahl (ID, Request).

**VAR\_INPUT**

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_obj_request;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_obj\\_request](#) [▶ 579]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.279 FB\_SEND\_093\_SNVT\_obj\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_obj\_status.

**SNVT Nummer:** 093.

**Beschreibung:** Objektstatus (ID, Status (4 Byte)).

#### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_obj_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_obj\\_status \[► 579\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** [Fehlerkennung \[► 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[► 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

## 7.2.280 FB\_SEND\_094\_SNVT\_preset



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_preset.

**SNVT Nummer:** 094.

**Beschreibung:** Voreinstellung (Lernmodus, Wert, Zeit).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_preset;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_preset \[► 581\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.



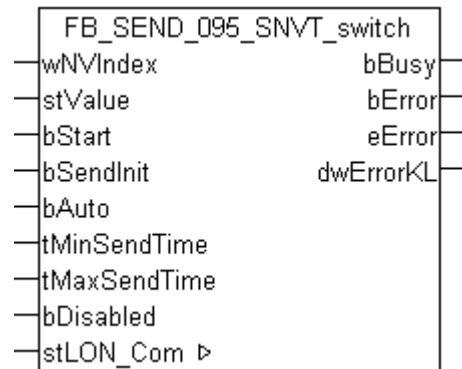
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.281 FB\_SEND\_095\_SNVT\_switch**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_switch.

**SNVT Nummer:** 095.

**Beschreibung:** Schalter (Wert, Status).

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_switch;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_switch` [▶ 587]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

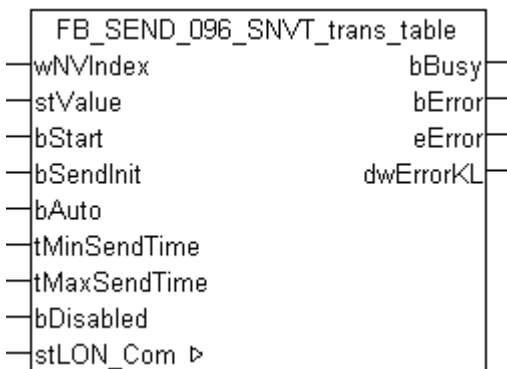
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.282 FB\_SEND\_096\_SNVT\_trans\_table**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_trans\_table.

**SNVT Nummer:** 096.

**Beschreibung:** Übersetzungstabelle (Anzahl Werte, Interpolation).

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_trans_table;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_trans\_table [▶ 589]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

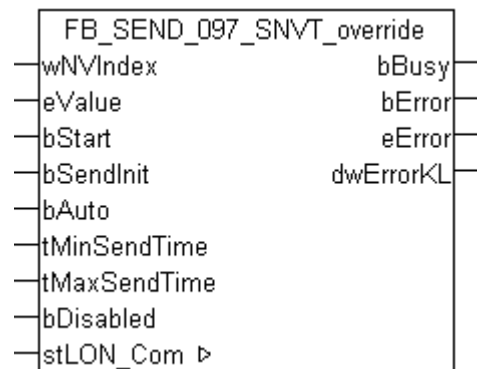
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.283 FB\_SEND\_097\_SNVT\_override**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_override.

**SNVT Nummer:** 097.

**Beschreibung:** Übersteuerungsmodus (Enumeration: 0= aktuellen Wert halten, 1= spezifischen Wert setzen, 2= default Wert setzen).

**VAR\_INPUT**

```
wNVIndex   : WORD;
eValue     : E_LON_override_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
```

```
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_override\\_t](#) [▶ 518]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.284 FB\_SEND\_098\_SNVT\_pwr\_fact



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pwr\_fact.

**SNVT Nummer:** 098.

**Beschreibung:** Leistungsfaktor (Multiplikator).

#### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 0.1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -1 / Max: 1.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.285 FB\_SEND\_099\_SNVT\_pwr\_fact\_f



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pwr\_fact\_f.

**SNVT Nummer:** 099.

**Beschreibung:** Leistungsfaktor (Multiplikator).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 0.1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -1 / Max: 1.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

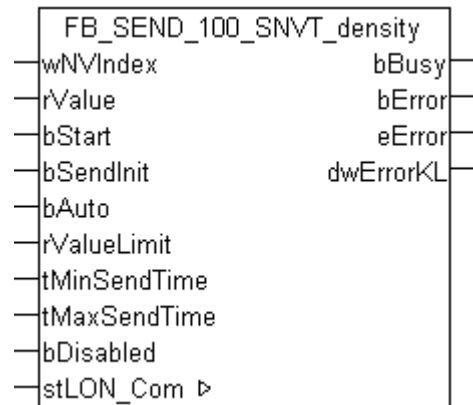
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.286 FB\_SEND\_100\_SNVT\_density**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_density.

**SNVT Nummer:** 100.

**Beschreibung:** Dichte (kg/m³).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 32767.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

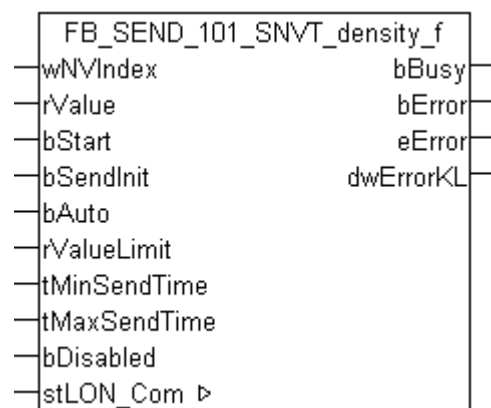
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.287 FB\_SEND\_101\_SNVT\_density\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_density\_f.

**SNVT Nummer:** 101.

**Beschreibung:** Dichte (kg/m³).



## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

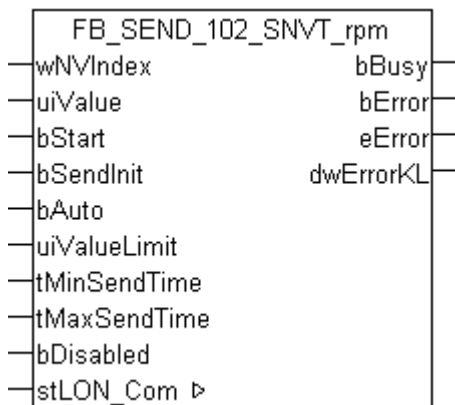
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.288 FB\_SEND\_102\_SNVT\_rpm



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_rpm.

**SNVT Nummer:** 102.

**Beschreibung:** Drehzahl (Umdrehungen/Minute (RPM)).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

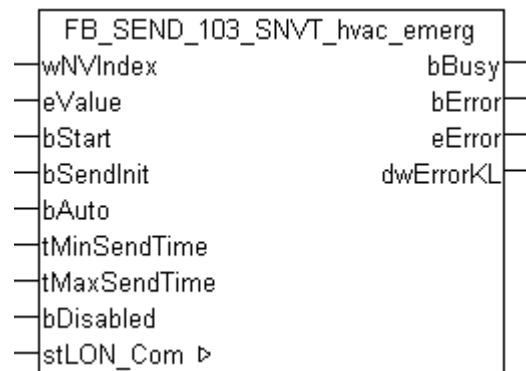
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.289 FB\_SEND\_103\_SNVT\_hvac\_emerg



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_emerg.

**SNVT Nummer:** 103.

**Beschreibung:** HLK Notbetrieb (Betriebsarten).

### VAR\_INPUT

```
wNVIndex : WORD;
eValue : E_LON_emerg_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe E\_LON\_emerg\_t [▶ 505]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

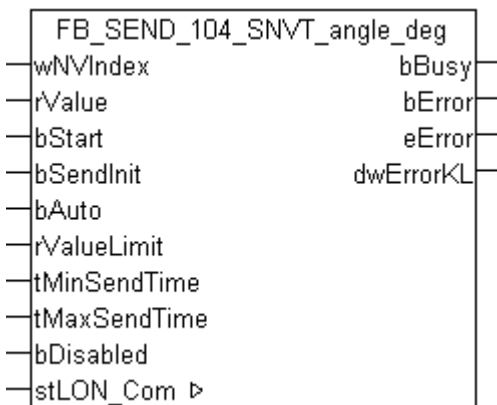
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.290 FB\_SEND\_104\_SNVT\_angle\_deg**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_angle\_deg.

**SNVT Nummer:** 104.

**Beschreibung:** Winkelangabe in 1/50-Grad-Schritten.

**VAR\_INPUT**

```
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -359.98 / Max: 360.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

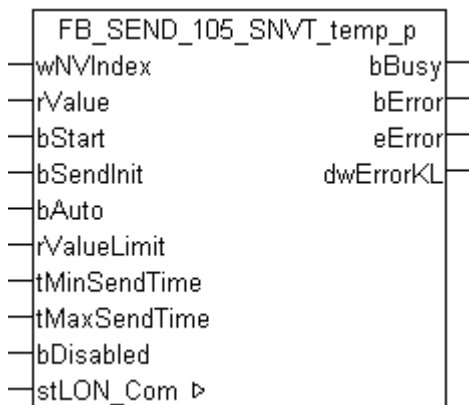
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.291 FB\_SEND\_105\_SNVT\_temp\_p



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp\_p.

**SNVT Nummer:** 105.

**Beschreibung:** Temperatur (Grad Celsius).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -273.17 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

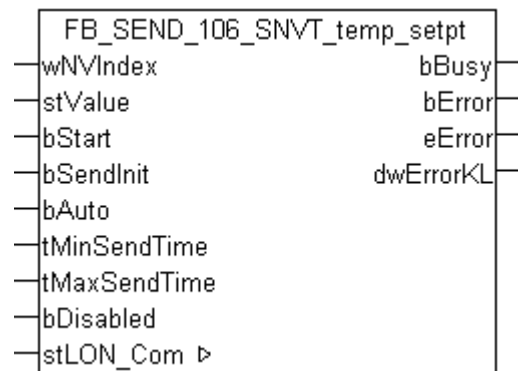
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.292 FB\_SEND\_106\_SNVT\_temp\_setpt



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp\_setpt.

**SNVT Nummer:** 106.

**Beschreibung:** Temperatur (6 Temperaturwerte).

### VAR\_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_temp_setpt;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_temp\_setpt [▶ 587]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

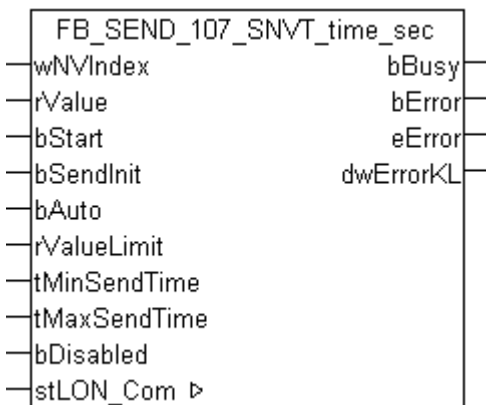
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.293 FB\_SEND\_107\_SNVT\_time\_sec**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_sec.

**SNVT Nummer:** 107.

**Beschreibung:** Abgelaufene Zeit (Sekunde).

**VAR\_INPUT**

```
wNVIndex    : WORD;
rValue      : REAL;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```



**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 6553.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

### 7.2.294 FB\_SEND\_108\_SNVT\_hvac\_mode



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_mode.

**SNVT Nummer:** 108.

**Beschreibung:** HLK Betriebsmodus (Betriebsarten).

#### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_hvac_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_hvac\\_t](#) [► 514]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

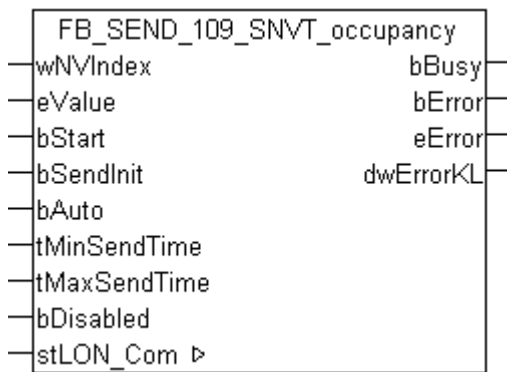
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.295 FB\_SEND\_109\_SNVT\_occupancy



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_occupancy.

**SNVT Nummer:** 109.

**Beschreibung:** Präsenzmeldung (Status).

### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_occup_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_occup\\_t \[► 518\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

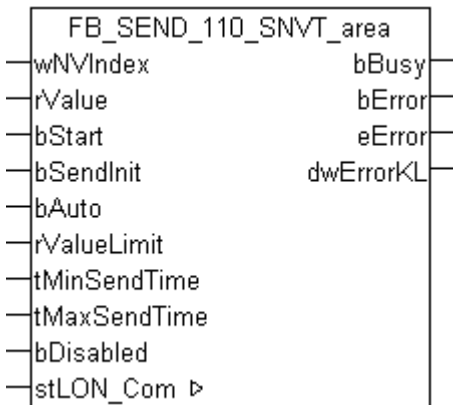
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.296 FB\_SEND\_110\_SNVT\_area**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_area.

**SNVT Nummer:** 110.

**Beschreibung:** Fläche (Quadratmeter).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 13.107.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

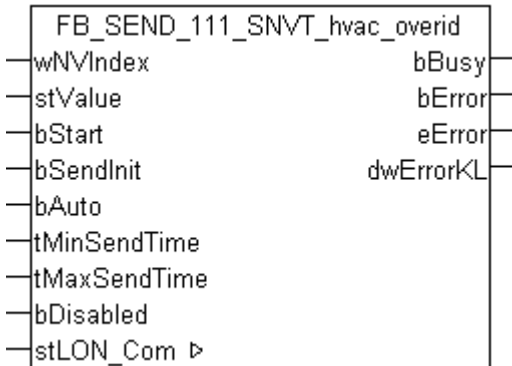
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.297 FB\_SEND\_111\_SNVT\_hvac\_overid**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_overid.

**SNVT Nummer:** 111.

**Beschreibung:** HLK Übersteuerungsmodus (Modus, Position/Durchfluss, Min/Max Durchflusswert).

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_hvac_overid;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_hvac\\_overid](#) [▶ 575]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Senden](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

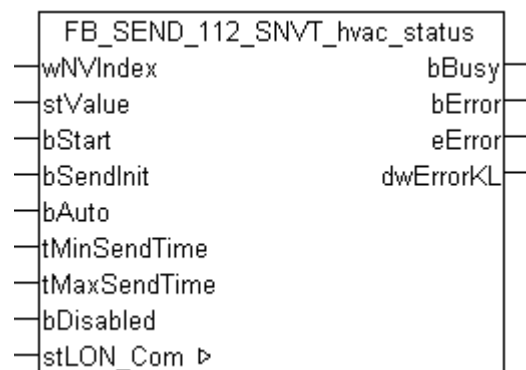
**dwErrorKL:** [Fehlerkennung](#) [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.298 FB\_SEND\_112\_SNVT\_hvac\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_status.

**SNVT Nummer:** 112.

**Beschreibung:** HLK Status (Modus, 5 Leistungsangaben, Alarmflag).

## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_hvac_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_hvac\\_status](#) [► 576]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

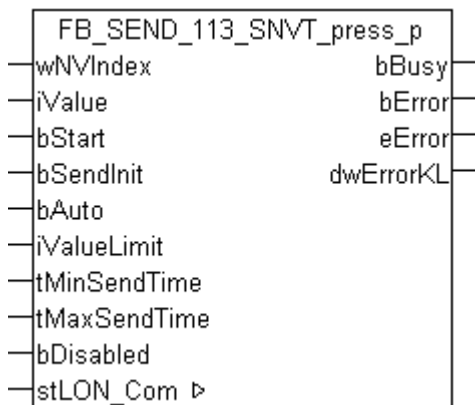
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.299 FB\_SEND\_113\_SNVT\_press\_p



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_press\_p.

**SNVT Nummer:** 113.

**Beschreibung:** Druck (Überdruck) (Pascal).

**VAR\_INPUT**

```

wNVIndex      : WORD;
iValue        : INT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
iValueLimit   : INT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**iValue:** Min: -32768 / Max: 32767.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**iValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*iValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```



**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlererkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com      : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.300 FB\_SEND\_114\_SNVT\_address**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_address.

**SNVT Nummer:** 114.

**Beschreibung:** Neuron Adresse (16-bit Adress Wert).

**VAR\_INPUT**

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 16384 / Max: 64767.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

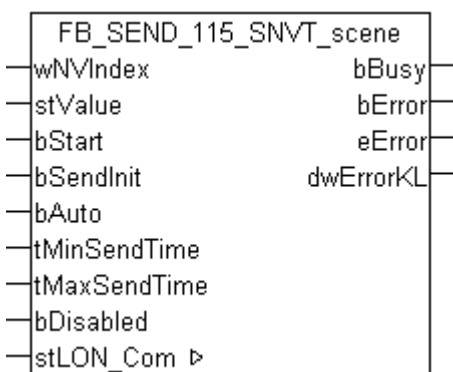
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.301 FB\_SEND\_115\_SNVT\_scene**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_scene.

**SNVT Nummer:** 115.

**Beschreibung:** Szenen (Funktion (abrufen/lernen), Szenennummer).

## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_scene;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_scene](#) [▶ 586]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

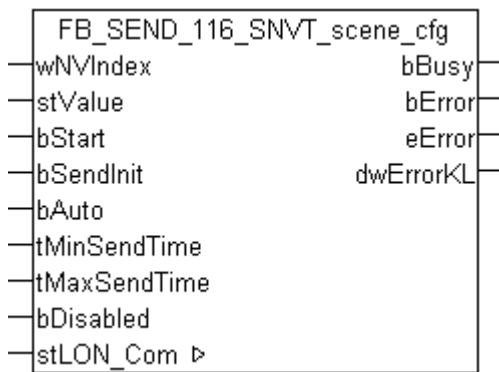
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.302 FB\_SEND\_116\_SNVT\_scene\_cfg



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_scene\_cfg.

**SNVT Nummer:** 116.

**Beschreibung:** Szenen Einstellung (Funktion , Szenennummer, Einstellung, Übergangsnummer, Übergangszeit, Verzögerungszeit, Priorität).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_scene_cfg;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_scene\\_cfg](#) [▶ 586]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

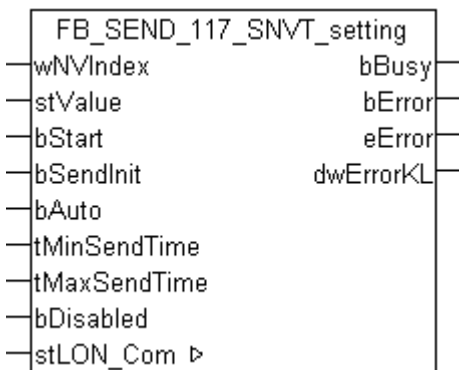
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.303 FB\_SEND\_117\_SNVT\_setting**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_setting.

**SNVT Nummer:** 117.

**Beschreibung:** Einstellungen (Funktion, Wert, Übergangsnummer).

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_setting;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_setting](#) [▶ 586]).

**bStart:** Der Baustein startet das Senden mit positiver Flanke dieser Variablen (unabhängig von "bAuto").

**bSendInit:** Der Baustein sendet beim Neustart der SPS automatisch 1 mal seine Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit "tMaxSendTime" abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird erst nach dem Ablauf einer minimale Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird nach nach Ablauf dieser Zeit gesendet, auch wenn die minimale Wertänderung (ValueLimit) nicht erreicht ist (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

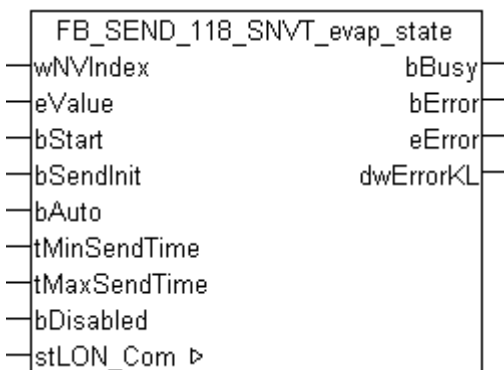
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.304 FB\_SEND\_118\_SNVT\_evap\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_evap\_state.

**SNVT Nummer:** 118.

**Beschreibung:** Verdampferstatus (Enumeration).

**VAR\_INPUT**

```
wNVIndex    : WORD;
eValue      : E_LON_evap_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe E\_LON\_evap\_t [▶ 507]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

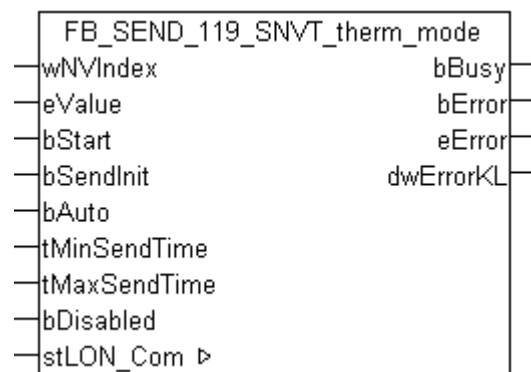
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.305 FB\_SEND\_119\_SNVT\_therm\_mode**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_therm\_mode.

**SNVT Nummer:** 119.

**Beschreibung:** Thermostatmodus (Enumeration(Freigabe, Modulation)).

**VAR\_INPUT**

```
wNVIndex    : WORD;
eValue      : E_LON_therm_mode_t;
bStart      : BOOL;
```

```
bSendInit : BOOL := bSendInitDefault;
bAuto     : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled  : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_therm\\_mode t \[► 534\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Senden \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

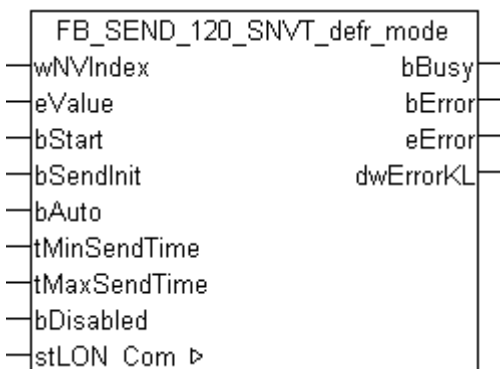
**dwErrorKL:** Fehlerkennung [\[► 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[► 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

**7.2.306 FB\_SEND\_120\_SNVT\_defr\_mode**





Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_defr\_mode.

**SNVT Nummer:** 120.

**Beschreibung:** Abtaumodus (Enumeration).

#### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_defrost_mode_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_defrost\\_mode\\_t](#) [▶ 502]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

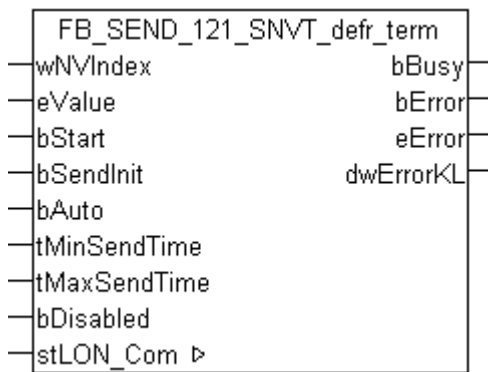
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.307 FB\_SEND\_121\_SNVT\_defr\_term



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_defr\_term.

**SNVT Nummer:** 121.

**Beschreibung:** Abschluss des Abtauvorgangs (Enumeration).

### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_defrost_term_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_defrost\\_term t \[► 503\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

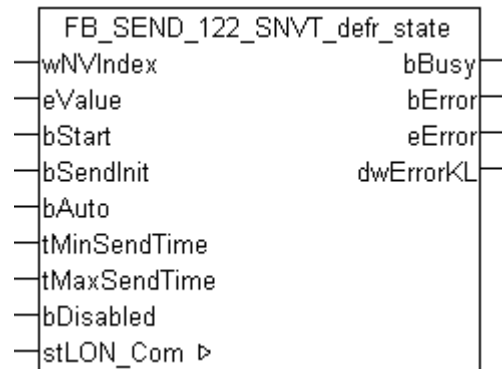
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.308 FB\_SEND\_122\_SNVT\_defr\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_defr\_state.

**SNVT Nummer:** 122.

**Beschreibung:** Abtaustatus (Enumeration).

**VAR\_INPUT**

```
wNVIndex : WORD;
eValue : E_LON_defrost_state_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe `E_LON_defrost_state_t` [▶ 503]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.309 FB\_SEND\_123\_SNVT\_time\_min**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_min.

**SNVT Nummer:** 123.

**Beschreibung:** Abgelaufen Zeit (Minuten).

**VAR\_INPUT**

```
wNVIndex    : WORD;
uiValue     : UINT;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.310 FB\_SEND\_124\_SNVT\_time\_hour**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_hour.

**SNVT Nummer:** 124.

**Beschreibung:** Abgelaufene Zeit (Stunde).

## VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

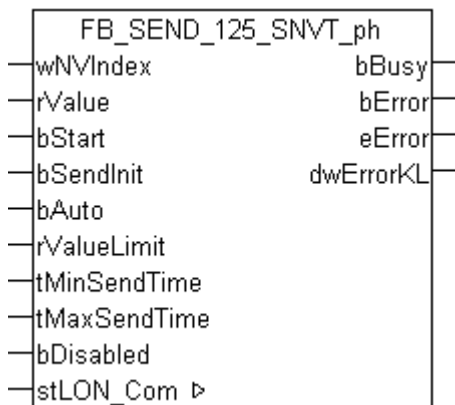
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.311 FB\_SEND\_125\_SNVT\_ph



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ph.

**SNVT Nummer:** 125.

**Beschreibung:** Säuregehalt (pH). Verhältnis der Ionenkonzentration.

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -32.768 / Max: 32.767.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.312 FB\_SEND\_126\_SNVT\_ph\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ph\_f.

**SNVT Nummer:** 126.

**Beschreibung:** Säuregehalt (pH). Verhältnis der Ionenkonzentration.

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -3.40E+38 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).



**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

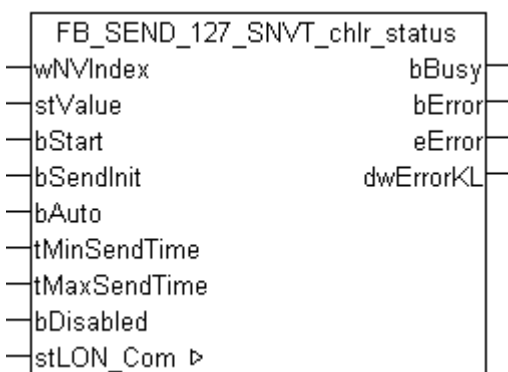
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.313 FB\_SEND\_127\_SNVT\_chlr\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_chlr\_status.

**SNVT Nummer:** 127.

**Beschreibung:** Kälteerzeugerstatus (run mode, op mode, Statusbits).

**VAR\_INPUT**

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_chlr_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_chlr\\_status](#) [► 566]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.314 FB\_SEND\_128\_SNVT\_tod\_event



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_tod\_event.

**SNVT Nummer:** 128.

**Beschreibung:** Präsenzzeit (Besetztstatus, nächste Besetztzeit).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_tod_event;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_tod\\_event \[► 588\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

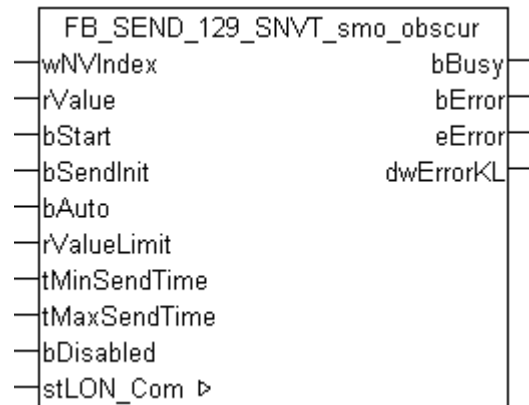
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.315 FB\_SEND\_129\_SNVT\_smo\_obscur**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_smo\_obscur.

**SNVT Nummer:** 129.

**Beschreibung:** Verdunkelung durch Rauch (Verdunkelung %).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

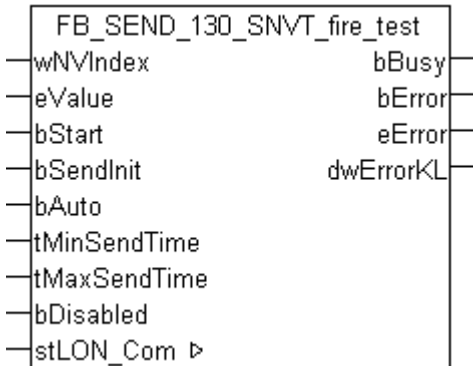
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.316 FB\_SEND\_130\_SNVT\_fire\_test**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_fire\_test.

**SNVT Nummer:** 130.

**Beschreibung:** Anforderung Test Brandmeldeanlage (Brandmeldetest Benennungen).

**VAR\_INPUT**

```
wNVIndex      : WORD;
eValue        : E_LON_fire_test_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_fire\\_test\\_t](#) [▶ 510]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

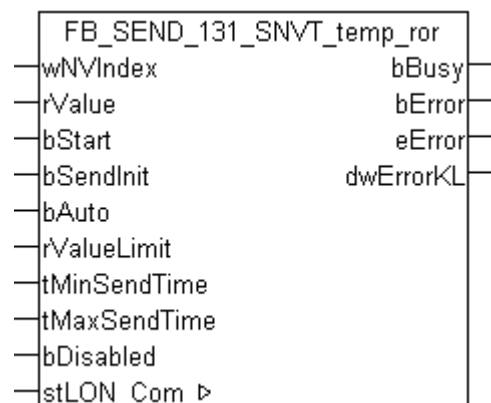
**dwErrorKL:** [Fehlerkennung](#) [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.317 FB\_SEND\_131\_SNVT\_temp\_ror**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp\_ror.

**SNVT Nummer:** 131.

**Beschreibung:** Wert der Temperaturänderung/-erhöhung (Grad Celsius/Minute).

## VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -16384 / Max: 16383.5.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

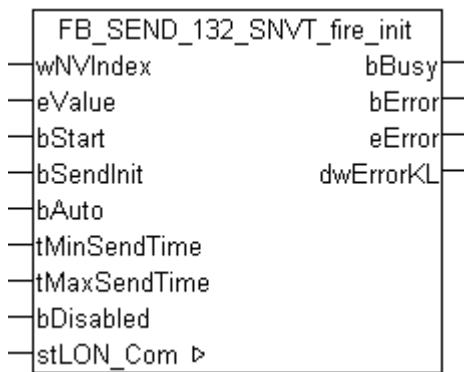
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.318 FB\_SEND\_132\_SNVT\_fire\_init



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_fire\_init.

**SNVT Nummer:** 132.

**Beschreibung:** Brandmeldertyp (Brandmeldertypen Namen).

### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_fire_initiator_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_fire\\_initiator\\_t](#) [► 510]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.



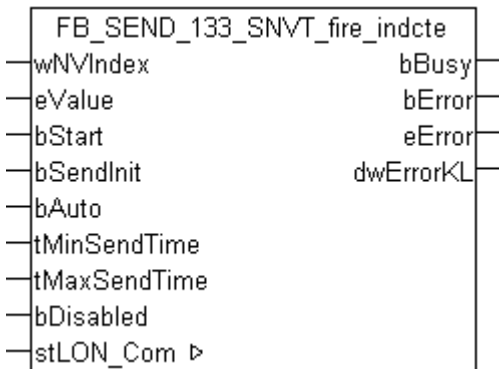
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.319 FB\_SEND\_133\_SNVT\_fire\_indcte**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_fire\_indcte.

**SNVT Nummer:** 133.

**Beschreibung:** Brandmeldeanzeige (Brandmeldeanzeigen Namen).

**VAR\_INPUT**

```
wNVIndex : WORD;
eValue : E_LON_fire_indicator_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe `E_LON_fire_indicator_t` [▶ 509]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.320 FB\_SEND\_134\_SNVT\_time\_zone**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_zone.

**SNVT Nummer:** 134.

**Beschreibung:** Zeitzonenbeschreibung (Offset, Typ, Start Sommerzeit, Ende Sommerzeit).

**VAR\_INPUT**

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_time_zone;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_time\_zone [▶ 588]). Wenn *stValue.eType\_of\_description* = eLON\_CAL\_GREG (0) werden die Variablen *stValue.stStart\_DST.uiG\_day\_of\_start\_DST* und *stValue.stEnd\_DST.uiG\_day\_of\_end\_DST* gesendet. Wenn *stValue.eType\_of\_description* = eLON\_CAL\_JUL (1) werden die Variablen

*stValue.stStart\_DST.uiJ\_day\_of\_start\_DST* und *stValue.stEnd\_DST.uiJ\_day\_of\_end\_DST* gesendet.  
 Wenn *stValue.eType\_of\_description = eLON\_CAL\_MEU(2)* werden die Strukturen  
*stValue.stStart\_DST.stM\_start\_DST* und *stValue.stEnd\_DST.stM\_end\_DST* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.321 FB\_SEND\_135\_SNVT\_earth\_pos**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_earth\_pos.

**SNVT Nummer:** 135.

**Beschreibung:** Position auf der Erde (Breiten & Längen Ausrichtung, Breitengrad & Min, Längengrad & Min, Höhe).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_earth_pos;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_earth\\_pos](#) [► 571]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

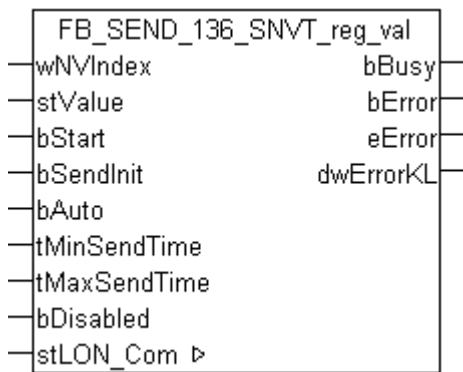
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.322 FB\_SEND\_136\_SNVT\_reg\_val



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_reg\_val.

**SNVT Nummer:** 136.

**Beschreibung:** Registerwert (Rohwert, bit-Code, Anzahl der Dezimalstellen).

### VAR\_INPUT

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_reg_val;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
  
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_reg\\_val \[► 585\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
  
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.323 FB\_SEND\_137\_SNVT\_reg\_val\_ts**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_reg\_val\_ts.

**SNVT Nummer:** 137.

**Beschreibung:** Registerwert (Rohwert, bit-Code, Anzahl der Dezimalstellen, Status, Zustand, Zeitstempel).

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_reg_val_ts;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_reg_val_ts` [▶ 585]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

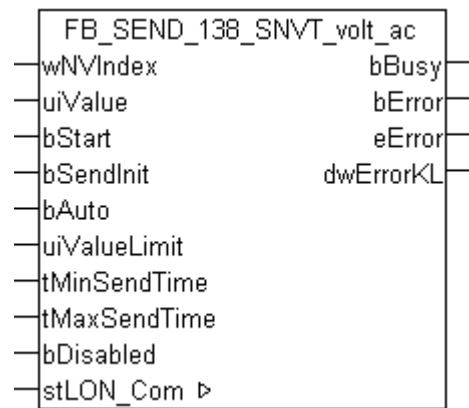
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.324 FB\_SEND\_138\_SNVT\_volt\_ac**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_volt\_ac.

**SNVT Nummer:** 138.

**Beschreibung:** Wechselspannung (Volt AC).

**VAR\_INPUT**

```
wNVIndex    : WORD;
uiValue     : UINT;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

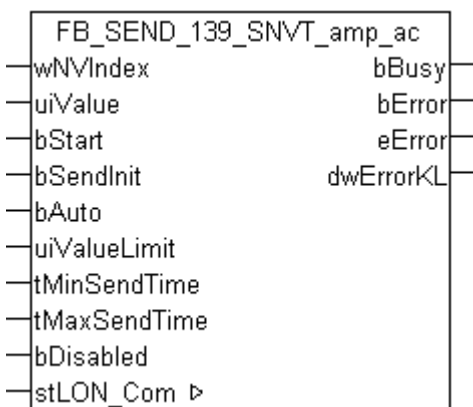
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.325 FB\_SEND\_139\_SNVT\_amp\_ac**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_amp\_ac.

**SNVT Nummer:** 139.



**Beschreibung:** Wechselstrom (Ampere AC).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

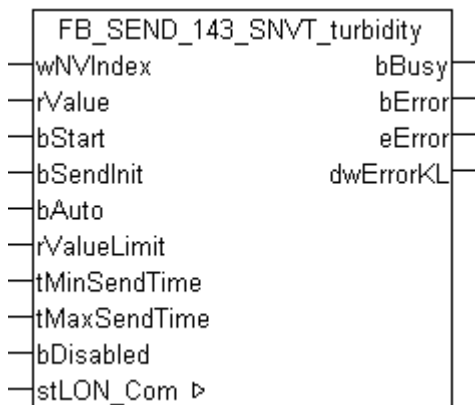
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.326 FB\_SEND\_143\_SNVT\_turbidity



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_turbidity.

**SNVT Nummer:** 143.

**Beschreibung:** Trübung (Trübeeinheit).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 65.535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

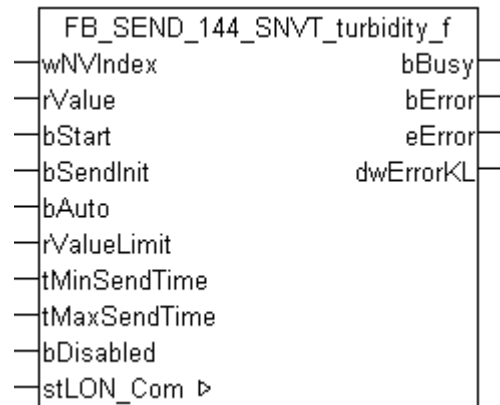
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.327 FB\_SEND\_144\_SNVT\_turbidity\_f**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_turbidity\_f.

**SNVT Nummer:** 144.

**Beschreibung:** Trübung (Trübeeinheit).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 3.40E+38.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.328 FB\_SEND\_145\_SNVT\_hvac\_type**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_type.

**SNVT Nummer:** 145.

**Beschreibung:** HLK Anlagentyp (HLK Anlagentypbeschreibung).

## VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_hvac_hvt_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_hvac\\_hvt\\_t \[▶ 511\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

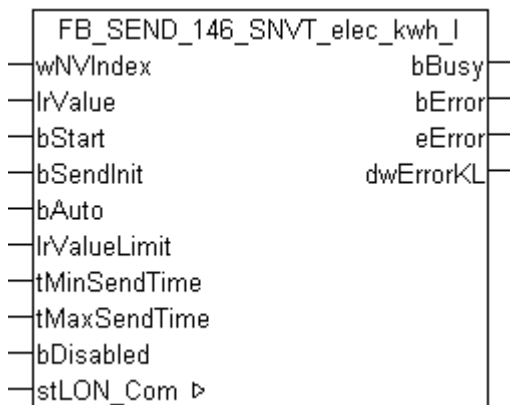
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.2.329 FB\_SEND\_146\_SNVT\_elec\_kwh\_I



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_elec\_kwh\_I.

**SNVT Nummer:** 146.

**Beschreibung:** Elektrische Energie (Kilowatt / Stunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
lrValue       : LREAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
lrValueLimit  : LREAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**lrValue:** Min: -214748364.8 / Max: 214748364.7.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**lrValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*lrValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

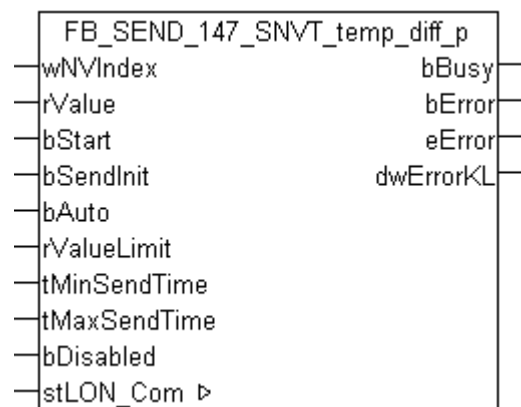
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.330 FB\_SEND\_147\_SNVT\_temp\_diff\_p**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_temp\_diff\_p.

**SNVT Nummer:** 147.

**Beschreibung:** Temperaturdifferenz (Grad Celsius).

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue : REAL;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

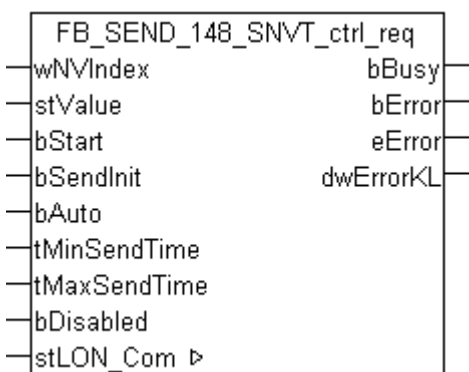
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.331 FB\_SEND\_148\_SNVT\_ctrl\_req**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ctrl\_req.

**SNVT Nummer:** 148.

**Beschreibung:** Steuerungsanfrage (Empfänger ID, Sender ID, Sender Priorität). Anfrage einer Steuerung für die Steuererlaubnis einer anderen Steuerung.



## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ctrl_req;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_ctrl\\_req](#) [► 569]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

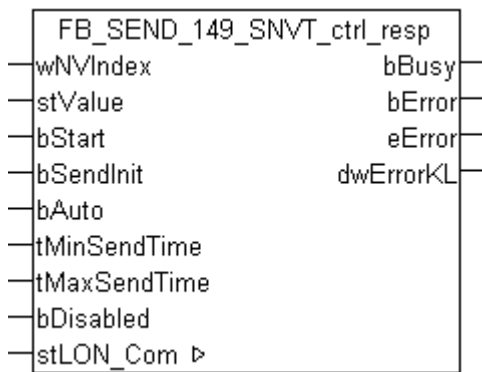
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com    : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.332 FB\_SEND\_149\_SNVT\_ctrl\_resp



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ctrl\_resp.

**SNVT Nummer:** 149.

**Beschreibung:** Steuerungsantwort (Status, Sender, Controller ID). Antwort auf eine Steuerungsanfrage.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ctrl_resp;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto        : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ).  
Wenn *stValue.eStatus* = eLON\_CTRLR\_RES (4) wird die Struktur *stValue.stSender.stRange* gesendet.  
Wenn *stValue.eStatus* <> eLON\_CTRLR\_RES (4) wird die Variable *stValue.stSender.uilD* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

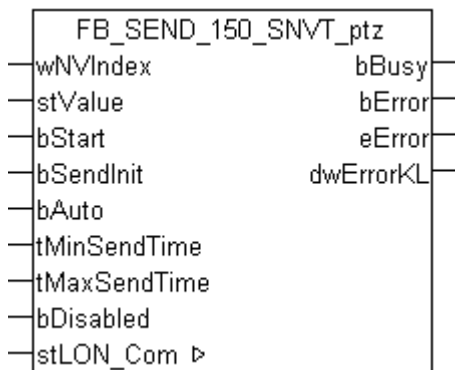
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.333 FB\_SEND\_150\_SNVT\_ptz



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ptz.

**SNVT Nummer:** 150.

**Beschreibung:** Kamera PTZ (SNZ) (Schwenkung, Schwenkgeschwindigkeit, Neigung, Neigungsgeschwindigkeit, Zoom, Zoomgeschwindigkeit).

### VAR\_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_ptz;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_ptz](#) [▶ 582]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.334 FB\_SEND\_151\_SNVT\_privacyzone**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_privacyzone.

**SNVT Nummer:** 151.

**Beschreibung:** Privatzone (Aktionsbereich, Zonennummer, Kamera ID). Gewisse Bereiche können von der Kamera ausgenommen werden.

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_privacyzone;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_privacyzone [▶ 581]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

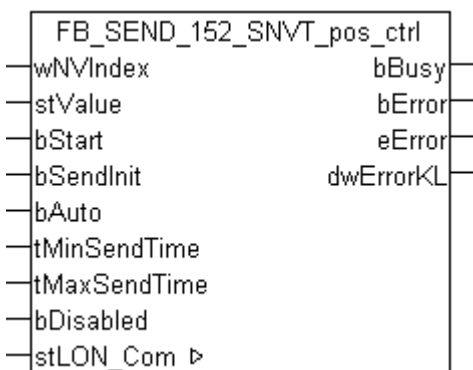
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.335 FB\_SEND\_152\_SNVT\_pos\_ctrl**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pos\_ctrl.

**SNVT Nummer:** 152.

**Beschreibung:** Positionseinstellung für Kameras in Netzwerken (Empfänger, Controller ID, Controller Priorität, Funktion, Aktion, Wert).

**VAR\_INPUT**

```

wNVIndex      : WORD;
stValue       : ST_LON_SNVT_pos_ctrl;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_pos\\_ctrl](#) [► 580]). Wenn *stValue.eFunction* = eLON\_CMF\_ABS wird die Struktur *stValue.stAbspos* gesendet. Wenn *stValue.eFunction* <> eLON\_CMF\_ABS wird die Variable *stValue.byNumber* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```

bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

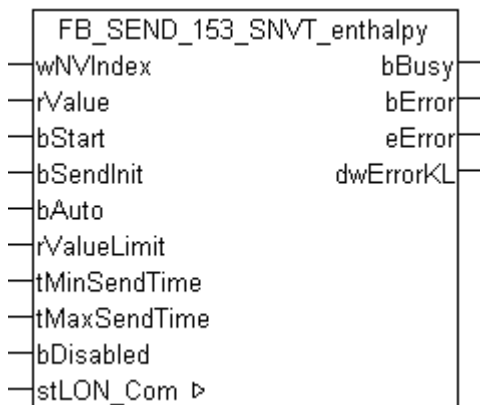
```

stLON_Com     : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.336 FB\_SEND\_153\_SNVT\_enthalpy



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_enthalpy.

**SNVT Nummer:** 153.

**Beschreibung:** Enthalpie (Kilojoules/kg).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.337 FB\_SEND\_154\_SNVT\_gfci\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_gfci\_status.

**SNVT Nummer:** 154.

**Beschreibung:** Fehlerstromschutzschalter Status (Fehlerstromschutzschalter Status Text).

**VAR\_INPUT**

```
wNVIndex : WORD;
eValue : E_LON_gfci_status_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe E\_LON\_gfci\_status\_t [▶ 511]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.



**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

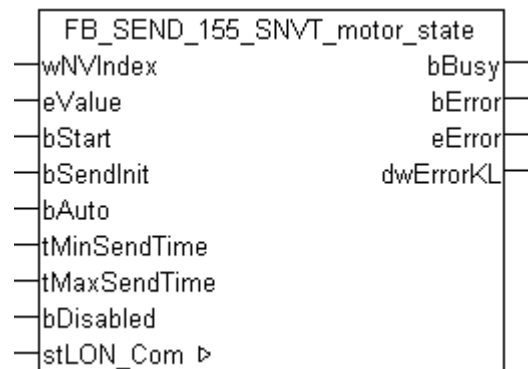
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.338 FB\_SEND\_155\_SNVT\_motor\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_motor\_state.

**SNVT Nummer:** 155.

**Beschreibung:** Motorstatus (Motorstatus Benennung).

**VAR\_INPUT**

```
wNVIndex    : WORD;
eValue      : E_LON_motor_state_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_motor\\_state t](#) [▶ 516]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

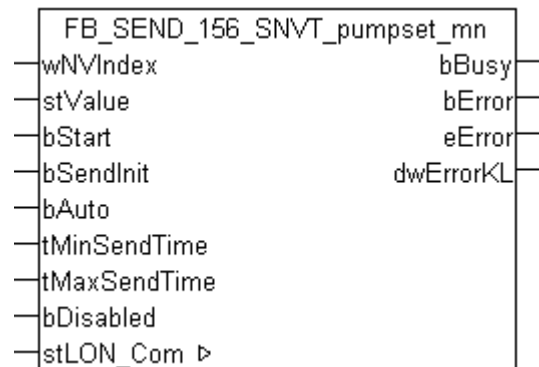
**dwErrorKL:** [Fehlerkennung](#) [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.339 FB\_SEND\_156\_SNVT\_pumpset\_mn**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pumpset\_mn.

**SNVT Nummer:** 156.

**Beschreibung:** Pumpengruppe (Hauptpumpe, Hilfspumpe, Priorität, Bereit, Not-Aus, Hauptpumpe freigegeben, Hilfspumpe freigegeben, Wartungsanforderung). Aufeinander synchronisierte Vakuumpumpen.

## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_pumpset_mn;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_pumpset\\_mn](#) [► 583]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

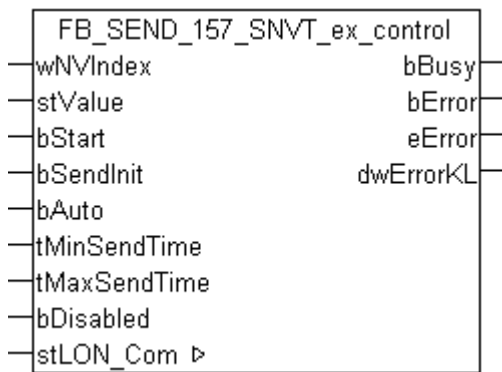
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.340 FB\_SEND\_157\_SNVT\_ex\_control



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ex\_control.

**SNVT Nummer:** 157.

**Beschreibung:** Exklusive Kontrolle (Status, Adresse). Ein Gerät hat die exklusive Kontrolle über ein anderes Gerät.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ex_control;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_ex\\_control](#) [► 574]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

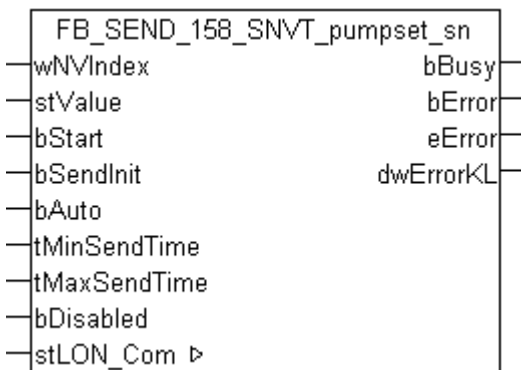
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.341 FB\_SEND\_158\_SNVT\_pumpset\_sn



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pumpset\_sn.

**SNVT Nummer:** 158.

**Beschreibung:** Pumpengruppensensor (Verdünnung, Ausstoß, Druck, Vakuum, ...).

### VAR\_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_pumpset_sn;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_pumpset\\_sn](#) [▶ 583]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

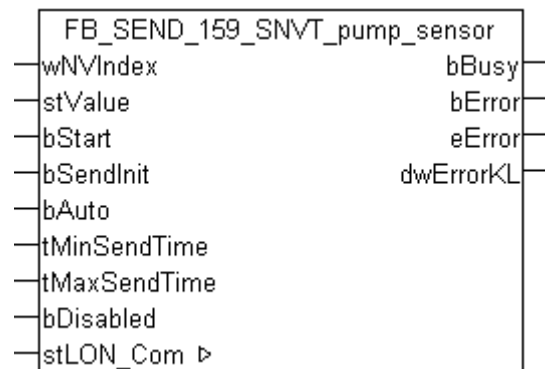
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.342 FB\_SEND\_159\_SNVT\_pump\_sensor**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_pump\_sensor.

**SNVT Nummer:** 159.

**Beschreibung:** Pumpensensor (Geschwindigkeit, Temperatur, Status).

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_pump_sensor;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_pump\_sensor [▶ 582]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

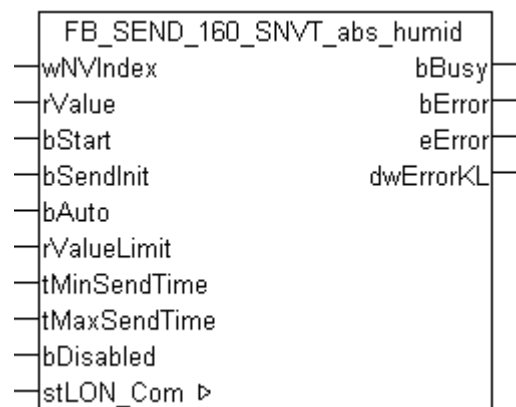
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.343 FB\_SEND\_160\_SNVT\_abs\_humid**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_abs\_humid.

**SNVT Nummer:** 160.

**Beschreibung:** Absolute Feuchtigkeit (Gramm/Kilogramm).

**VAR\_INPUT**

```
wNVIndex   : WORD;
rValue     : REAL;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
```

```

bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;

```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 655.35.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```

bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;

```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```

stLON_Com : ST_LON_Communication;

```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).



## 7.2.344 FB\_SEND\_161\_SNVT\_flow\_p



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_flow\_p.

**SNVT Nummer:** 161.

**Beschreibung:** Volumenstrom (Kubikmeter / Stunde).

### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 655.35.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.345 FB\_SEND\_162\_SNVT\_dev\_c\_mode**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_dev\_c\_mode.

**SNVT Nummer:** 162.

**Beschreibung:** Gerätebetriebsart (Gerätebetriebsart Zustände).

**VAR\_INPUT**

```
wNVIndex : WORD;
eValue : E_LON_device_c_mode_t;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe E\_LON\_device\_c\_mode\_t [▶ 504]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.346 FB\_SEND\_163\_SNVT\_valve\_mode**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_valve\_mode.

**SNVT Nummer:** 163.

**Beschreibung:** Ventilzustand.

**VAR\_INPUT**

```
wNVIndex    : WORD;
eValue      : E_LON_valve_mode_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_valve\\_mode\\_t](#) [▶ 535]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

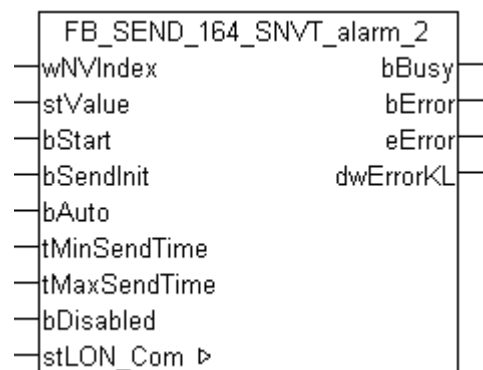
**dwErrorKL:** [Fehlerkennung](#) [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.347 FB\_SEND\_164\_SNVT\_alarm\_2**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_alarm\_2.

**SNVT Nummer:** 164.

**Beschreibung:** Alarmstatus 2. Meldet den Alarmstatus eines Funktionsbausteins oder Geräts. Ersetzt SNVT\_alarm.

## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_alarm_2;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_alarm\\_2](#) [▶ 566]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

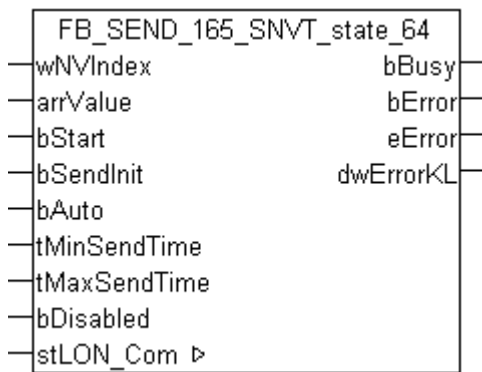
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.348 FB\_SEND\_165\_SNVT\_state\_64



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_state\_64.

**SNVT Nummer:** 165.

**Beschreibung:** Statusinformation (64 individuelle Bitwerte). Jeder Status ist ein boolescher Wert.

### VAR\_INPUT

```
wNVIndex      : WORD;
arrValue      : ARRAY [0..63] OF BOOL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**arrValue:** 0-63 Bit.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
eError       : E_LON_ERROR;
dwErrorKL    : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

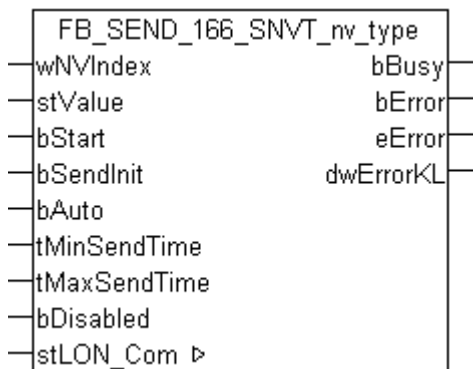
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.349 FB\_SEND\_166\_SNVT\_nv\_type**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_nv\_type.

**SNVT Nummer:** 166.

**Beschreibung:** Netzwerkvariablentyp. Typbeschreibung für veränderbare Netzwerkvariablen.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_nv_type;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_nv_type` [▶ 578]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

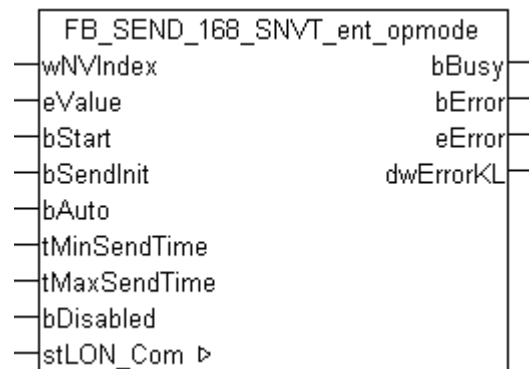
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.350 FB\_SEND\_168\_SNVT\_ent\_opmode**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ent\_opmode.

**SNVT Nummer:** 168.

**Beschreibung:** Betriebsart für Zutrittsobjekte (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

**VAR\_INPUT**

```
wNVIndex    : WORD;
eValue      : E_LON_ent_opmode_cmd_t;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe E\_LON\_ent\_opmode\_cmd\_t [▶ 506]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.



**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

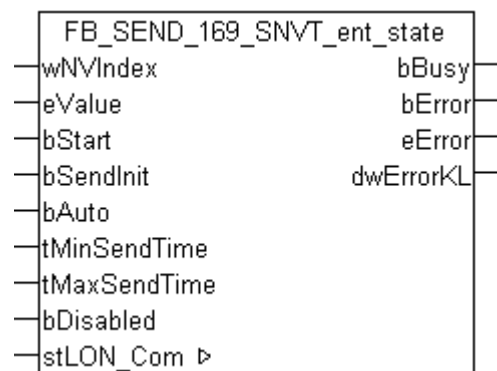
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.351 FB\_SEND\_169\_SNVT\_ent\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ent\_state.

**SNVT Nummer:** 169.

**Beschreibung:** Zustand für Zutrittsobjekte (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

**VAR\_INPUT**

```
wNVIndex   : WORD;
eValue     : E_LON_ent_cmd_t;
bStart     : BOOL;
bSendInit  : BOOL := bSendInitDefault;
```

```
bAuto      : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_ent\\_cmd\\_t](#) [▶ 506]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

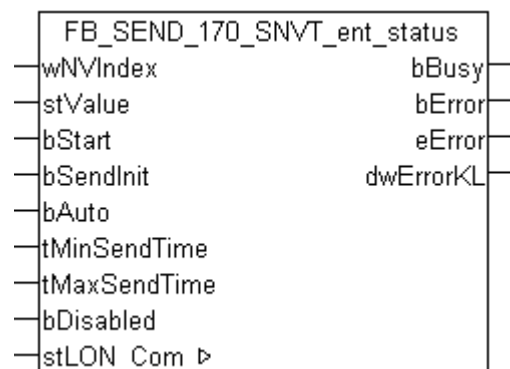
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.352 FB\_SEND\_170\_SNVT\_ent\_status



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_ent\_status.

**SNVT Nummer:** 170.

**Beschreibung:** Status von Zutrittsobjekten (Türen, Schlösser, oder Objekte, die den Zutritt zu Bereichen erlauben oder verbieten).

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_ent_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_ent\\_status \[► 572\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

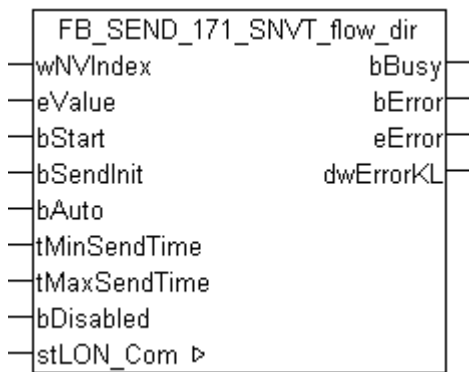
**dwErrorKL:** [Fehlerkennung \[► 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[► 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).

## 7.2.353 FB\_SEND\_171\_SNVT\_flow\_dir



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_flow\_dir.

**SNVT Nummer:** 171.

**Beschreibung:** Gangrichtung. Richtung des zu erlaubenden Gangflusses oder Richtung des aktuellen Gangflusses.

### VAR\_INPUT

```
wNVIndex      : WORD;
eValue        : E_LON_flow_direction_t;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**eValue:** Enum, der gesendet werden soll (siehe [E\\_LON\\_flow\\_direction\\_t](#) [► 511]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.354 FB\_SEND\_172\_SNVT\_hvac\_satsts**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_hvac\_satsts.

**SNVT Nummer:** 172.

**Beschreibung:** HLK Sättigungsstatus.

Eine 0 in einem Feld bedeutet, dass die mit dem Feld verbundene Anlage (Gerät) nicht gesättigt ist oder die Endabschaltung nicht erreicht bevor der gewünschte Sollwert erreicht ist.

Eine 1 in einem Feld bedeutet, dass die mit dem Feld verbundene Anlage (Gerät) gesättigt ist oder die Endabschaltung erreicht bevor der gewünschte Sollwert nicht erreicht ist.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_hvac_satsts;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser Index wird für das Binding der LON Knoten benötigt. Pro Klemme (KL6401) sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_hvac\\_satsts](#) [▶ 575]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.355 FB\_SEND\_173\_SNVT\_dev\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_dev\_status.

**SNVT Nummer:** 173.

**Beschreibung:** Gerätestatus.

**VAR\_INPUT**

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_dev_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_dev\\_status](#) [▶ 571]).  
 Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0) wird die Struktur *stValue.stDev\_type.stPump\_ctrl* gesendet.  
 Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1) wird die Struktur *stValue.stDev\_type.stValvePos* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.356 FB\_SEND\_174\_SNVT\_dev\_fault**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_dev\_fault.

**SNVT Nummer:** 174.

**Beschreibung:** Fehlerstatus. Fehlerinformation für ein Gerät.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_dev_fault;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_dev\\_fault \[► 570\]](#)).

Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0) wird die Struktur *stValue.stDev\_type.stPump\_ctrl* gesendet.

Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1) wird die Struktur *stValue.stDev\_type.stValvePos* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** [Fehlerkennung \[► 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[► 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[► 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[► 562\]](#)).



## 7.2.357 FB\_SEND\_175\_SNVT\_dev\_maint



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_dev\_maint.

**SNVT Nummer:** 175.

**Beschreibung:** Geräte Wartungsstatus.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_dev_maint;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_dev\\_maint \[► 571\]](#)).

Wenn *stValue.eDevice\_select* = eLON\_DV\_PUMP\_CTRL (0) wird die Struktur *stValue.stDev\_type.stPump\_ctrl* gesendet.

Wenn *stValue.eDevice\_select* = eLON\_DV\_VALVE\_POS (1) wird die Struktur *stValue.stDev\_type.stValvePos* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.358 FB\_SEND\_176\_SNVT\_date\_event**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_date\_event.

**SNVT Nummer:** 176.

**Beschreibung:** Status eines Ereignisses.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_date_event;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_date\\_event](#) [▶ 570]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

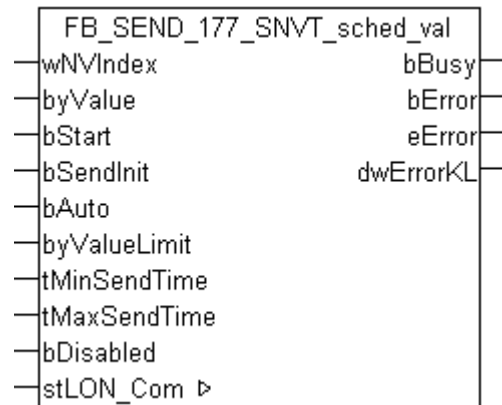
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.359 FB\_SEND\_177\_SNVT\_sched\_val**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_sched\_val.

**SNVT Nummer:** 177.

**Beschreibung:** Kalenderwert. Index eines Kalenderwerts, dass einen Eintrag in einem SCPT Wert Definitionsfeld array wählt, oder ein direkter Wertausgang ist.

**VAR\_INPUT**

```
wNVIndex    : WORD;
byValue     : BYTE;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
byValueLimit : BYTE := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**byValue:** Min: 0 / Max: 255.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**byValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*byValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

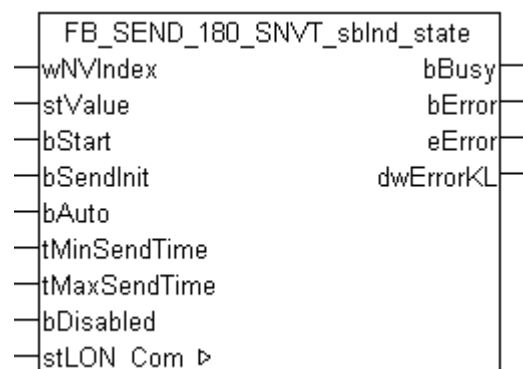
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.360 FB\_SEND\_180\_SNVT\_sblnd\_state**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_sblnd\_state.

**SNVT Nummer:** 180.

**Beschreibung:** Jalousie Status.

#### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_sblnd_state;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_sblnd\\_state](#) [[▶ 585](#)]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [[▶ 612](#)], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [[▶ 612](#)]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [[▶ 612](#)]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

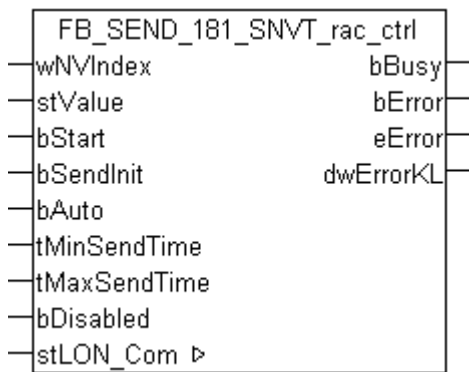
**dwErrorKL:** Fehlerkennung [[▶ 613](#)] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

## 7.2.361 FB\_SEND\_181\_SNVT\_rac\_ctrl



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_rac\_ctrl.

**SNVT Nummer:** 181.

**Beschreibung:** Steuerung Soundfunktion. Aktiviert die Soundfunktion einer bestimmten Quelle.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_rac_ctrl;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_rac\\_ctrl \[► 584\]](#)).  
Wenn *stValue.bDest\_p2p* = TRUE wird die Struktur *stValue.stAddr\_dest.stP2p* gesendet.  
Wenn *stValue.bDest\_p2p* = FALSE wird die Struktur *stValue.stAddr\_dest.stP2m* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

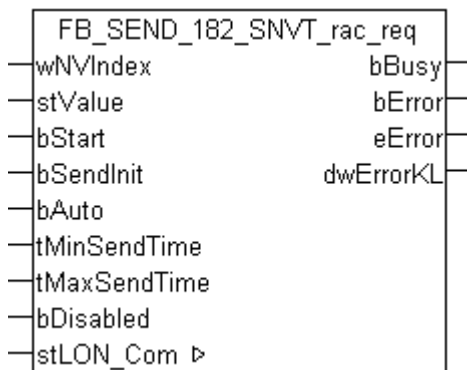
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.362 FB\_SEND\_182\_SNVT\_rac\_req



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_rac\_req.

**SNVT Nummer:** 182.

**Beschreibung:** Anforderung Soundfunktion. Fordert die Soundfunktion einer bestimmten Quelle an.

### VAR\_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_rac_req;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_rac\\_req](#) [► 584]). Wenn *stValue.bDest\_p2p* = TRUE wird die Struktur *stValue.stAddr\_dest.stP2p* gesendet. Wenn *stValue.bDest\_p2p* = FALSE wird die Struktur *stValue.stAddr\_dest.stP2m* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

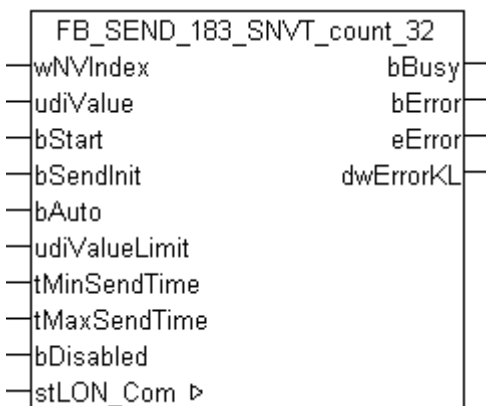
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.363 FB\_SEND\_183\_SNVT\_count\_32**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_count\_32.

**SNVT Nummer:** 183.

**Beschreibung:** Absoluter Zähler. 32-bit Zähler.

**VAR\_INPUT**

```
wNVIndex      : WORD;
udiValue      : UDINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**udiValue:** Min: 0 / Max: 4294967294.



**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**udiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*udiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

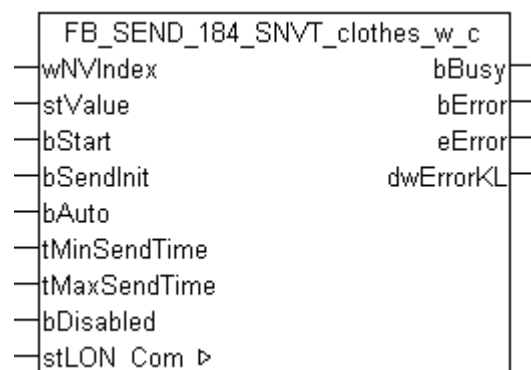
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.364 FB\_SEND\_184\_SNVT\_clothes\_w\_c**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_clothes\_w\_c.

**SNVT Nummer:** 184.

**Beschreibung:** Waschmaschine / Kommandos. Zum Programmieren und Starten einer Waschmaschine.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_clothes_w_c;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_c](#) [► 568]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

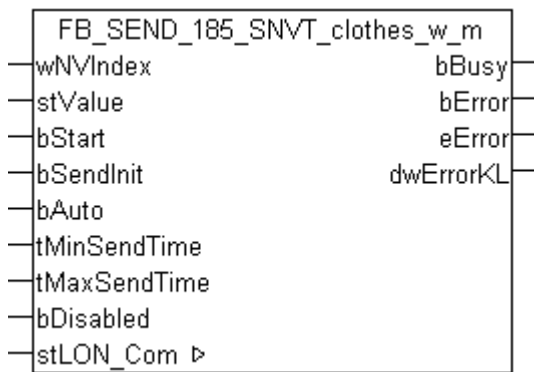
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.365 FB\_SEND\_185\_SNVT\_clothes\_w\_m



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_clothes\_w\_m.

**SNVT Nummer:** 185.

**Beschreibung:** Waschmaschine / Management-Status. Aktueller Status von Tür / Deckel und Abfluss.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_clothes_w_m;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_m \[► 568\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[► 467\]](#)). Gleichzeitig wird *bError* TRUE.

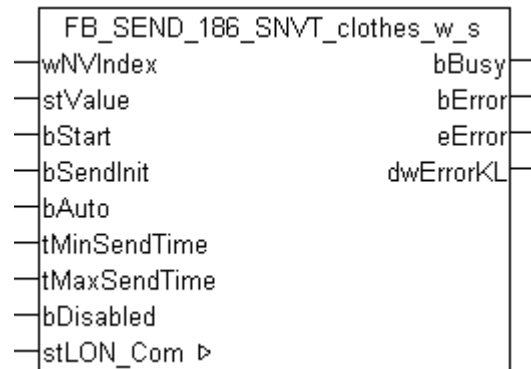
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.366 FB\_SEND\_186\_SNVT\_clothes\_w\_s**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** `SNVT_clothes_w_s`.

**SNVT Nummer:** 186.

**Beschreibung:** Waschmaschine / Status. Aktueller Zustand einer Waschmaschine, einschließlich Befehls- und Alarm-Informationen.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_clothes_w_s;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_clothes_w_s` [▶ 569]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com  : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.367 FB\_SEND\_187\_SNVT\_clothes\_w\_a**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_clothes\_w\_a.

**SNVT Nummer:** 187.

**Beschreibung:** Waschmaschine / Alarmmeldungen.

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_clothes_w_a;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe ST\_LON\_SNVT\_clothes\_w\_a [▶ 566]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

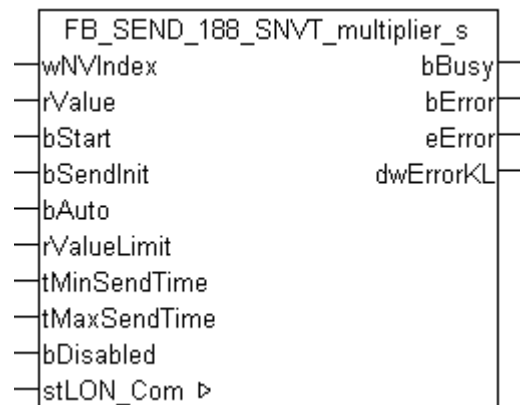
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.368 FB\_SEND\_188\_SNVT\_multiplier\_s**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_multiplier\_s.

**SNVT Nummer:** 188.

**Beschreibung:** Multiplikator.

**VAR\_INPUT**

```
wNVIndex : WORD;
rValue   : REAL;
bStart   : BOOL;
bSendInit : BOOL := bSendInitDefault;
```

```
bAuto      : BOOL := bAutoDefault;
rValueLimit : REAL := 0.1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNvIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: 0 / Max: 2.54.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [► 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

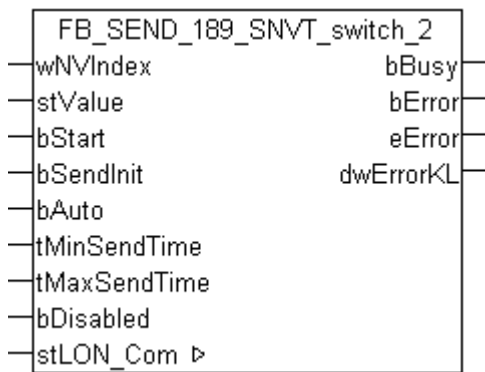
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

## 7.2.369 FB\_SEND\_189\_SNVT\_switch\_2



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_switch\_2.

**SNVT Nummer:** 189.

**Beschreibung:** Schalter zum Einstellen von Szenen und Einstellungen. Eine erweiterte Version des SNVT\_switch zur Einstellung von Szenen und Einstellungen ähnlich den SNVT\_scene und SNVT\_setting.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_switch_2;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_switch\\_2](#) [▶ 587]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.



**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

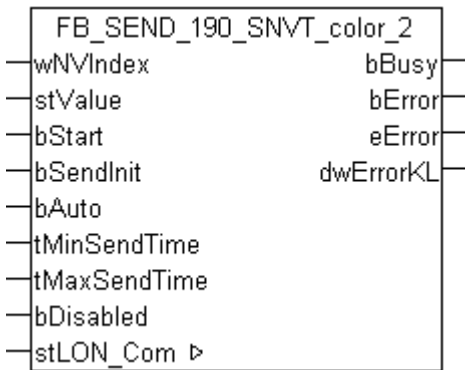
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.370 FB\_SEND\_190\_SNVT\_color\_2**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_color\_2.

**SNVT Nummer:** 190.

**Beschreibung:** Farbe.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_color_2;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_color\\_2](#) [▶ 569]).

Wenn *stValue.eEncoding* = eLON\_COLOR\_CIE31\_LUMEN (0) wird die Struktur *stValue.stColor\_value.stCIE1931\_lumen* gesendet.

Wenn *stValue.eEncoding* = eLON\_COLOR\_CIE31\_PERCENT (1) wird die Struktur *stValue.stColor\_value.stCIE1931\_percent* gesendet.

Wenn *stValue.eEncoding* = eLON\_COLOR\_RGB (2) wird die Struktur *stValue.stColor\_value.stRGB* gesendet.

Wenn *stValue.eEncoding* = eLON\_COLOR\_TEMPERATURE (3) wird die Variable *stValue.stColor\_value.uiColor\_temperature* gesendet.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [► 467]). Gleichzeitig wird *bError* TRUE.

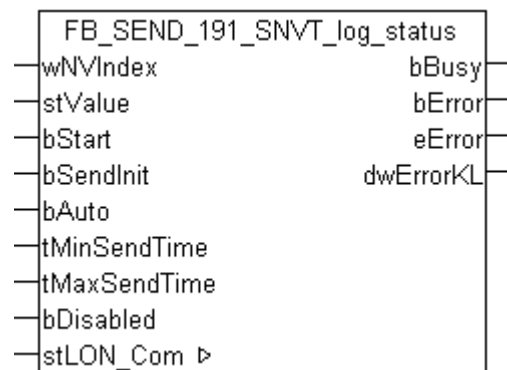
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins FB\_LON\_KL6401() [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [► 562]).

**7.2.371 FB\_SEND\_191\_SNVT\_log\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_log\_status.

**SNVT Nummer:** 191.

**Beschreibung:** Log Status (hundertstel Sekunde). Stellt den aktuellen Status eines Datenloggings dar. Update basiert auf dem cpLogNotificationThreshold Wert. Zeigt nur den Status an. Alarme werden über das Node Object nvoAlarm2 ausgegeben. Benötigt, wenn das Node Object keinen nvoLogStat Ausgang enthält.

**VAR\_INPUT**

```
wNVIndex    : WORD;
stValue     : ST_LON_SNVT_log_status;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
```

```
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled    : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_log\\_status](#) [▶ 578]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

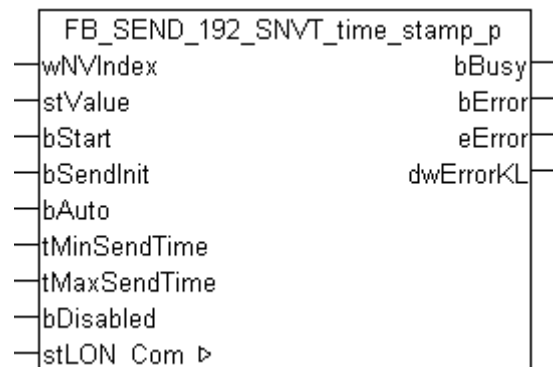
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

## 7.2.372 FB\_SEND\_192\_SNVT\_time\_stamp\_p



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_stamp\_p.

**SNVT Nummer:** 192.

**Beschreibung:** Präziser Zeitstempel (Sekunden). Zeitstempel mit einer Auflösung einer hundertstel Sekunde.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : TIMESTRUCT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON-Knoten benötigt. Pro LON-Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [TIMESTRUCT](#)). Die Strukturvariable *wDayOfWeek* ist hier nicht gültig und wird nicht übertragen.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [[▶ 612](#)], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

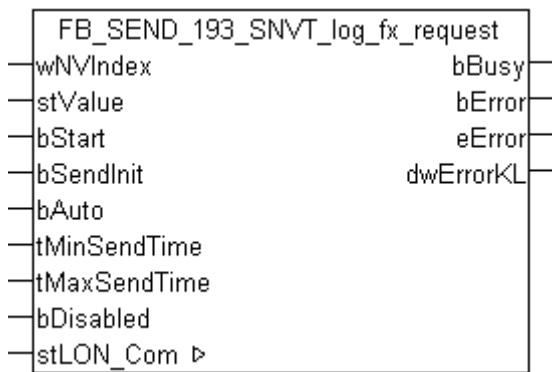
**dwErrorKL:** Fehlerkennung [[▶ 613](#)] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [[▶ 68](#)] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [[▶ 562](#)]).

## 7.2.373 FB\_SEND\_193\_SNVT\_log\_fx\_request



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_log\_fx\_request.

**SNVT Nummer:** 193.

**Beschreibung:** Log Datei Transfer Anforderung. Fordert einen Datenlog per FTP Transfer an. Muss auf eine Standard FTP Anfrage folgen, um eine Datenlog Datei zu erhalten. Erforderlich auf Geräten, die das Datenlogger Funktionsprofil nutzen und den Datenlog Transfer über FTP erlauben.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_log_fx_request;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_log\\_fx\\_request \[► 577\]](#)).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens \[► 612\]](#), wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden \[► 612\]](#). Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

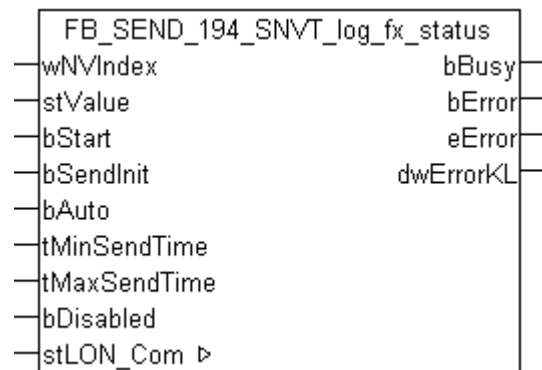
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.374 FB\_SEND\_194\_SNVT\_log\_fx\_status



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_log\_fx\_status.

**SNVT Nummer:** 194.

**Beschreibung:** Log Datei Transfer Status. Zeigt den Status eines Datenlogs per FTP Transfer an. Erforderlich auf Geräten, die das Datenlogger Funktionsprofil nutzen und den Datenlog Transfer über FTP erlauben.

## VAR\_INPUT

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_log_fx_status;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_log\\_fx\\_status](#) [► 577]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [► 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [► 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [► 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.375 FB\_SEND\_195\_SNVT\_log\_request**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_log\_request.

**SNVT Nummer:** 195.

**Beschreibung:** Log Status Anforderung. Fordert den aktuellen Status eines Datenlogs an. Der Status wird über das SNVT\_log\_status output ausgegeben.

**VAR\_INPUT**

```
wNVIndex    : WORD;
uiValue     : UINT;
bStart      : BOOL;
bSendInit   : BOOL := bSendInitDefault;
bAuto       : BOOL := bAutoDefault;
uiValueLimit : UINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled   : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 1 / Max: 65535.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Senden [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

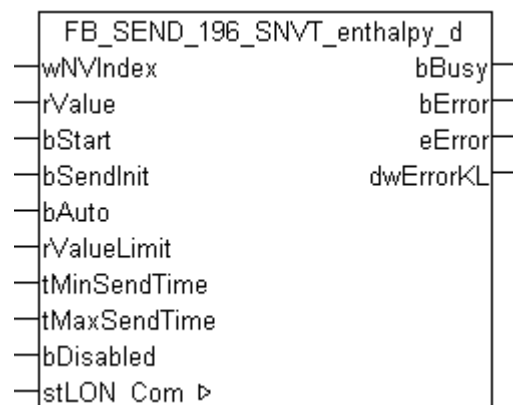
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.376 FB\_SEND\_196\_SNVT\_enthalpy\_d**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):



**SNVT Name:** SNVT\_enthalpy\_d.

**SNVT Nummer:** 196.

**Beschreibung:** Enthalpie Unterschied (kJ/kg).

#### VAR\_INPUT

```
wNVIndex      : WORD;
rValue        : REAL;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
rValueLimit   : REAL := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**rValue:** Min: -327.68 / Max: 327.67.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**rValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*rValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

#### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

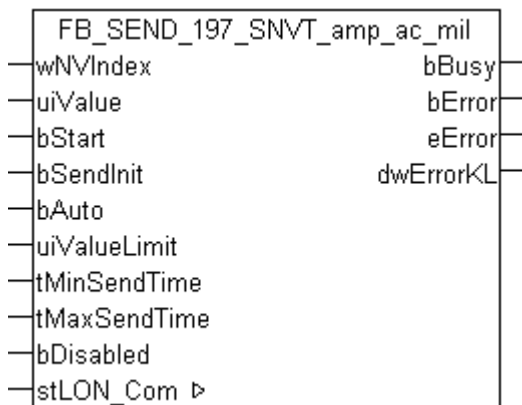
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

#### VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

## 7.2.377 FB\_SEND\_197\_SNVT\_amp\_ac\_mil



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_amp\_ac\_mil.

**SNVT Nummer:** 197.

**Beschreibung:** Elektrischer Strom (Milliampere).

### VAR\_INPUT

```
wNVIndex      : WORD;
uiValue       : UINT;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
uiValueLimit  : UINT := 1;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**uiValue:** Min: 0 / Max: 65534.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**uiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*uiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [▶ 467]). Gleichzeitig wird *bError* TRUE.

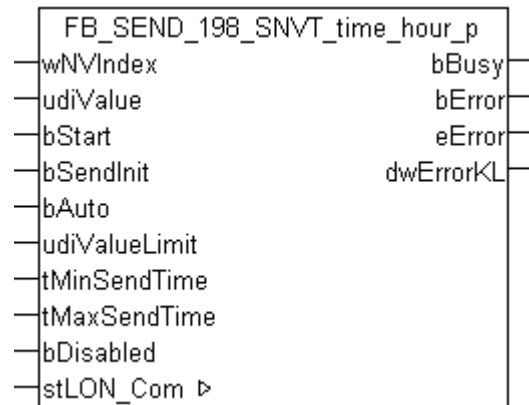
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [▶ 562]).

**7.2.378 FB\_SEND\_198\_SNVT\_time\_hour\_p**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_time\_hour\_p.

**SNVT Nummer:** 198.

**Beschreibung:** Zeit in Stunden.

**VAR\_INPUT**

```
wNVIndex : WORD;
udiValue : UDINT;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
udiValueLimit : UDINT := 1;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**udiValue:** Min: 0 / Max: 4294967294.

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**udiValueLimit:** Parameter für das automatische Senden [▶ 612]. Der Wert wird nur gesendet, wenn die Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung gesendet. Bei Enums und Strukturen ist dieser Wert nicht vorhanden. Dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet, auch wenn die mindest Wertänderung (*udiValueLimit*) nicht erreicht wurde (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe E\_LON\_ERROR [▶ 467]). Gleichzeitig wird *bError* TRUE.

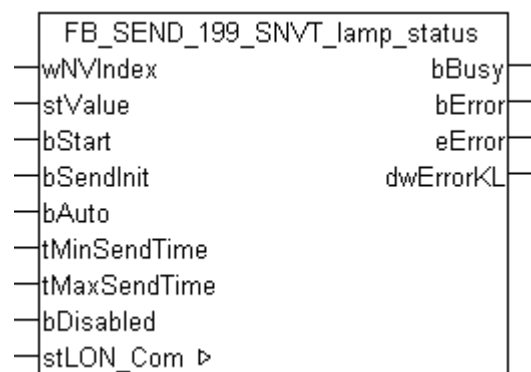
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins FB\_LON\_KL6401() [▶ 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird FB\_LON\_KL6401() [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe ST\_LON\_Communication [▶ 562]).

**7.2.379 FB\_SEND\_199\_SNVT\_lamp\_status**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_lamp\_status.

**SNVT Nummer:** 199.

**Beschreibung:** Lampen Status.

## VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_lamp_status;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_lamp\\_status](#) [► 577]).

**bStart:** Der Baustein startet das Senden mit positiver Flanke dieser Variablen (unabhängig von "bAuto").

**bSendInit:** Der Baustein sendet beim Neustart der SPS automatisch 1 mal seine Werte.

**bAuto:** Anwahl des automatischen Sendens [► 612], wenn sich ein Wert ändert oder wenn die Zeit "tMaxSendTime" abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [► 612]. Ein neuer Wert wird erst nach dem Ablauf einer minimale Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [► 612]. Der Wert wird nach nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

## VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [► 467]). Gleichzeitig wird *bError* TRUE.

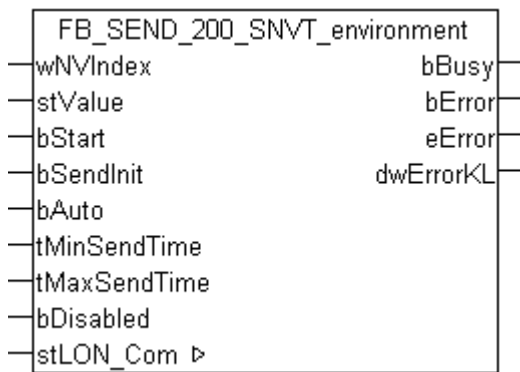
**dwErrorKL:** Fehlerkennung [► 613] des Bausteins [FB\\_LON\\_KL6401\(\)](#) [► 68]. Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

## VAR\_IN\_OUT

```
stLON_Com     : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\)](#) [► 68] mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication](#) [► 562]).

## 7.2.380 FB\_SEND\_200\_SNVT\_environment



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_environment.

**SNVT Nummer:** 200.

**Beschreibung:** Umgebung.

### VAR\_INPUT

```
wNVIndex      : WORD;
stValue       : ST_LON_SNVT_environment;
bStart        : BOOL;
bSendInit     : BOOL := bSendInitDefault;
bAuto         : BOOL := bAutoDefault;
tMinSendTime  : TIME := tMinSendTimeDefault;
tMaxSendTime  : TIME := tMaxSendTimeDefault;
bDisabled     : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe [ST\\_LON\\_SNVT\\_environment](#) [[▶ 573](#)]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von *bAuto*).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des [automatischen Sendens](#) [[▶ 612](#)], wenn sich ein Wert ändert oder wenn die Zeit *tMaxSendTime* abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das [automatische Senden](#) [[▶ 612](#)]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das [automatische Senden](#) [[▶ 612](#)]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

### VAR\_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
eError        : E_LON_ERROR;
dwErrorKL     : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR](#) [[▶ 467](#)]). Gleichzeitig wird *bError* TRUE.

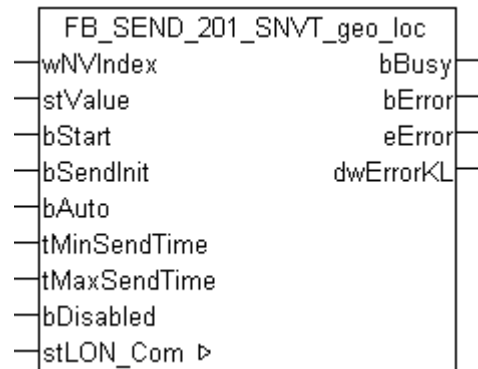
**dwErrorKL:** Fehlerkennung [▶ 613] des Bausteins `FB_LON_KL6401()` [▶ 68]. Die Variable `eError` hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird `bError` TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird `FB_LON_KL6401()` [▶ 68] mit den Sende-/Empfangsbausteinen verbunden (siehe `ST_LON_Communication` [▶ 562]).

**7.2.381 FB\_SEND\_201\_SNVT\_geo\_loc**



Dieser Baustein sendet folgende LON Ausgangsvariable (nvo):

**SNVT Name:** SNVT\_geo\_loc.

**SNVT Nummer:** 201.

**Beschreibung:** Geographische Lage.

**VAR\_INPUT**

```
wNVIndex : WORD;
stValue : ST_LON_SNVT_geo_loc;
bStart : BOOL;
bSendInit : BOOL := bSendInitDefault;
bAuto : BOOL := bAutoDefault;
tMinSendTime : TIME := tMinSendTimeDefault;
tMaxSendTime : TIME := tMaxSendTimeDefault;
bDisabled : BOOL := FALSE;
```

**wNVIndex:** Eindeutiger Index. Dieser wird für das Binding der LON Knoten benötigt. Pro LON Klemme sind maximal 62 SNVT's erlaubt. Werte von 0 bis 61 möglich.

**stValue:** Struktur der Daten, die gesendet werden sollen (siehe `ST_LON_SNVT_geo_loc` [▶ 575]).

**bStart:** Eine positive Flanke startet das Senden (unabhängig von `bAuto`).

**bSendInit:** Sendet beim Neustart der SPS automatisch 1-mal die Werte.

**bAuto:** Anwahl des automatischen Sendens [▶ 612], wenn sich ein Wert ändert oder wenn die Zeit `tMaxSendTime` abgelaufen ist (pollen).

**tMinSendTime:** Parameter für das automatische Senden [▶ 612]. Ein neuer Wert wird frühestens nach Ablauf dieser Zeit gesendet. Somit wird ein ständiges Senden verhindert.

**tMaxSendTime:** Parameter für das automatische Senden [▶ 612]. Der Wert wird spätestens nach Ablauf dieser Zeit gesendet (pollen). Der Wert 0 deaktiviert diese Funktion.

**bDisabled:** TRUE = Abwahl des Bausteins.

**VAR\_OUTPUT**

```
bBusy      : BOOL;
bError     : BOOL;
eError     : E_LON_ERROR;
dwErrorKL  : DWORD;
```

**bBusy:** Der Ausgang ist TRUE solange das Senden der Werte läuft.

**bError:** Der Ausgang wird TRUE sobald ein Fehler auftritt. Dieser Fehler wird über die Variable *eError* beschrieben.

**eError:** Der Ausgang gibt im Fehlerfall einen Fehlercode aus (siehe [E\\_LON\\_ERROR \[▶ 467\]](#)). Gleichzeitig wird *bError* TRUE.

**dwErrorKL:** Fehlerkennung [\[▶ 613\]](#) des Bausteins [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#). Die Variable *eError* hat in diesem Fall den Wert "eKL6401\_Error". Gleichzeitig wird *bError* TRUE.

**VAR\_IN\_OUT**

```
stLON_Com : ST_LON_Communication;
```

**stLON\_Com:** Über diese Struktur wird [FB\\_LON\\_KL6401\(\) \[▶ 68\]](#) mit den Sende-/Empfangsbausteinen verbunden (siehe [ST\\_LON\\_Communication \[▶ 562\]](#)).

## 7.3 Datentypen

**Enums**

| Datentypen                                     | Beschreibung   |
|--|--|
| <a href="#">E_LON_alarm_type t [▶ 493]</a>     | Genutzt durch: SNVT_alarm / SNVT_alarm_2   |
| <a href="#">E_LON_appl_cwc t [▶ 495]</a>       | Genutzt durch: SNVT_clothes_w_c / SNVT_clothes_w_s   |
| <a href="#">E_LON_appl_cwp t [▶ 495]</a>       | Genutzt durch: SNVT_clothes_w_c  |
| <a href="#">E_LON_appl_cws t [▶ 496]</a>       | Genutzt durch: SNVT_clothes_w_c / SNVT_clothes_w_s   |
| <a href="#">E_LON_appl_rin t [▶ 497]</a>       | Genutzt durch: SNVT_clothes_w_c  |
| <a href="#">E_LON_boolean t [▶ 497]</a>        | Genutzt durch: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPThighLimit1Enable / SCPThighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT_clothes_w_c / SNVT_pump_sensor / SNVT_pumpset_mn / SNVT_pumpset_sn |
| <a href="#">E_LON_calendar_type t [▶ 497]</a>  | Genutzt durch: SNVT_time_zone  |
| <a href="#">E_LON_cam_act t [▶ 497]</a>        | Genutzt durch: SNVT_pos_ctrl   |
| <a href="#">E_LON_cam_func t [▶ 498]</a>       | Genutzt durch: SNVT_pos_ctrl   |
| <a href="#">E_LON_chiller t [▶ 498]</a>        | Genutzt durch: SNVT_chlr_status  |
| <a href="#">E_LON_color_encoding t [▶ 498]</a> | Genutzt durch: SNVT_color_2  |
| <a href="#">E_LON_config_source t [▶ 499]</a>  | Genutzt durch: SNVT_config_src   |
| <a href="#">E_LON_control_resp t [▶ 499]</a>   | Genutzt durch: SNVT_ctrl_resp  |
| <a href="#">E_LON_currency t [▶ 499]</a>       | Genutzt durch: SNVT_currency   |
| <a href="#">E_LON_days_of_week t [▶ 502]</a>   | Genutzt durch: SCPTtimePeriod / SNVT_date_day / SNVT_time_zone   |
| <a href="#">E_LON_defrost_mode t [▶ 502]</a>   | Genutzt durch: SNVT_defr_mode  |
| <a href="#">E_LON_defrost_state t [▶ 503]</a>  | Genutzt durch: SNVT_defr_state   |



| <b>Datentypen</b>  | <b>Beschreibung</b>   |
|--|---|
| <a href="#">E_LON_defrost_term_t</a> [ <a href="#">▶ 503</a> ]           | Genutzt durch: SNVT_defr_term                                       |
| <a href="#">E_LON_device_c_mode_t</a> [ <a href="#">▶ 504</a> ]          | Genutzt durch: SNVT_dev_c_mode                                      |
| <a href="#">E_LON_device_select_t</a> [ <a href="#">▶ 505</a> ]          | Genutzt durch: SNVT_dev_fault / SNVT_dev_maint / SNVT_dev_status    |
| <a href="#">E_LON_discrete_levels_t</a> [ <a href="#">▶ 505</a> ]        | Genutzt durch: SNVT_clothes_w_c / SNVT_lev_disc                     |
| <a href="#">E_LON_emerg_t</a> [ <a href="#">▶ 505</a> ]                  | Genutzt durch: SNVT_hvac_emerg                                      |
| <a href="#">E_LON_ent_cmd_t</a> [ <a href="#">▶ 506</a> ]                | Genutzt durch: SNVT_ent_state                                       |
| <a href="#">E_LON_ent_opmode_cmd_t</a> [ <a href="#">▶ 506</a> ]         | Genutzt durch: SNVT_ent_opmode / SNVT_ent_status                    |
| <a href="#">E_LON_evap_t</a> [ <a href="#">▶ 507</a> ]                   | Genutzt durch: SNVT_evap_state                                      |
| <a href="#">E_LON_ex_control_t</a> [ <a href="#">▶ 508</a> ]             | Genutzt durch: SNVT_ex_control                                      |
| <a href="#">E_LON_file_request_t</a> [ <a href="#">▶ 508</a> ]           | Genutzt durch: SNVT_file_req  |
| <a href="#">E_LON_file_status_t</a> [ <a href="#">▶ 508</a> ]            | Genutzt durch: SNVT_file_status                                     |
| <a href="#">E_LON_fire_indicator_t</a> [ <a href="#">▶ 509</a> ]         | Genutzt durch: SNVT_fire_indcte                                     |
| <a href="#">E_LON_fire_initiator_t</a> [ <a href="#">▶ 510</a> ]         | Genutzt durch: SNVT_fire_init                                       |
| <a href="#">E_LON_fire_test_t</a> [ <a href="#">▶ 510</a> ]              | Genutzt durch: SNVT_fire_test                                       |
| <a href="#">E_LON_flow_direction_t</a> [ <a href="#">▶ 511</a> ]         | Genutzt durch: SNVT_flow_dir  |
| <a href="#">E_LON_gfci_status_t</a> [ <a href="#">▶ 511</a> ]            | Genutzt durch: SNVT_gfci_status                                     |
| <a href="#">E_LON_hvac_hvt_t</a> [ <a href="#">▶ 511</a> ]               | Genutzt durch: SNVT_hvac_type                                       |
| <a href="#">E_LON_hvac_overid_t</a> [ <a href="#">▶ 512</a> ]            | Genutzt durch: SNVT_hvac_overid                                     |
| <a href="#">E_LON_hvac_t</a> [ <a href="#">▶ 514</a> ]                   | Genutzt durch: SNVT_chlr_status / SNVT_hvac_mode / SNVT_hvac_status |
| <a href="#">E_LON_learn_mode_t</a> [ <a href="#">▶ 515</a> ]             | Genutzt durch: SNVT_preset  |
| <a href="#">E_LON_log_status_t</a> [ <a href="#">▶ 516</a> ]             | Genutzt durch: SCPTlogRecord / SNVT_log_status                      |
| <a href="#">E_LON_motor_state_t</a> [ <a href="#">▶ 516</a> ]            | Genutzt durch: SNVT_motor_state / SNVT_pumpset_mn                   |
| <a href="#">E_LON_nv_type_category_t</a> [ <a href="#">▶ 516</a> ]       | Genutzt durch: SNVT_nv_type   |
| <a href="#">E_LON_object_request_t</a> [ <a href="#">▶ 517</a> ]         | Genutzt durch: SNVT_obj_request                                     |
| <a href="#">E_LON_occup_t</a> [ <a href="#">▶ 518</a> ]                  | Genutzt durch: SNVT_occupancy / SNVT_tod_event                      |
| <a href="#">E_LON_override_t</a> [ <a href="#">▶ 518</a> ]               | Genutzt durch: SNVT_override  |
| <a href="#">E_LON_pan_dir_t</a> [ <a href="#">▶ 519</a> ]                | Genutzt durch: SNVT_ptz   |
| <a href="#">E_LON_priority_level_t</a> [ <a href="#">▶ 519</a> ]         | Genutzt durch: SNVT_alarm / SNVT_alarm_2 / SNVT_pumpset_mn          |
| <a href="#">E_LON_privacyzone_t</a> [ <a href="#">▶ 520</a> ]            | Genutzt durch: SNVT_privacyzone                                     |
| <a href="#">E_LON_rail_audio_sensor_type_t</a> [ <a href="#">▶ 520</a> ] | Genutzt durch: SNVT_rac_ctrl / SNVT_rac_req                         |
| <a href="#">E_LON_rail_audio_type_t</a> [ <a href="#">▶ 521</a> ]        | Genutzt durch: SNVT_rac_ctrl / SNVT_rac_req                         |
| <a href="#">E_LON_reg_val_unit_t</a> [ <a href="#">▶ 522</a> ]           | Genutzt durch: SNVT_reg_val / SNVT_reg_val_ts                       |
| <a href="#">E_LON_sbld cmd_source_t</a> [ <a href="#">▶ 524</a> ]        | Genutzt durch: SNVT_sbld_state                                      |
| <a href="#">E_LON_sbld_error_t</a> [ <a href="#">▶ 526</a> ]             | Genutzt durch: SNVT_sbld_state                                      |
| <a href="#">E_LON_scene_config_t</a> [ <a href="#">▶ 526</a> ]           | Genutzt durch: SNVT_scene_cfg                                       |
| <a href="#">E_LON_scene_t</a> [ <a href="#">▶ 527</a> ]                  | Genutzt durch: SNVT_scene   |
| <a href="#">E_LON_sec_state_t</a> [ <a href="#">▶ 528</a> ]              | Genutzt durch: SNVT_sec_state                                       |
| <a href="#">E_LON_sec_status_t</a> [ <a href="#">▶ 529</a> ]             | Genutzt durch: SNVT_sec_status                                      |
| <a href="#">E_LON_setting_t</a> [ <a href="#">▶ 530</a> ]                | Genutzt durch: SNVT_setting   |
| <a href="#">E_LON_switch_state_t</a> [ <a href="#">▶ 531</a> ]           | Genutzt durch: SNVT_switch_2  |

| Datentypen  | Beschreibung                    |
|---|---------------------------------|
| <a href="#">E_LON_telcom_states_t</a> [ <a href="#">▶ 533</a> ] | Genutzt durch: SNVT_telcom      |
| <a href="#">E_LON_therm_mode_t</a> [ <a href="#">▶ 534</a> ]    | Genutzt durch: SNVT_therm_mode  |
| <a href="#">E_LON_tilt_dir_t</a> [ <a href="#">▶ 534</a> ]      | Genutzt durch: SNVT_ptz         |
| <a href="#">E_LON_unit_temp_t</a> [ <a href="#">▶ 534</a> ]     | Genutzt durch: SNVT_pump_sensor |
| <a href="#">E_LON_valve_mode_t</a> [ <a href="#">▶ 535</a> ]    | Genutzt durch: SNVT_valve_mode  |
| <a href="#">E_LON_zoom_t</a> [ <a href="#">▶ 535</a> ]          | Genutzt durch: SNVT_ptz         |

**Hardware Types**

| Datentypen   | Beschreibung               |
|--|----------------------------|
| <a href="#">ST_LON_Parameter_IN_36B</a> [ <a href="#">▶ 560</a> ]  | Prozessabbild der Eingänge |
| <a href="#">ST_LON_Parameter_OUT_36B</a> [ <a href="#">▶ 561</a> ] | Prozessabbild der Ausgänge |

**LON\_TYPES**

| Datentypen  | Beschreibung     |
|---|------------------|
| <a href="#">E_LON_Parameter_Datatypes</a> [ <a href="#">▶ 485</a> ] | Enums SNVT Typen |

**Structure/AuxiliaryStructure/SNVT\_chlr\_status**

| Datentypen  | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_chlr_state</a> [ <a href="#">▶ 538</a> ] | Genutzt durch: SNVT_chlr_status |

**Structure/AuxiliaryStructure/SNVT\_clothes\_w\_c**

| Datentypen  | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_action</a> [ <a href="#">▶ 538</a> ]   | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_dry</a> [ <a href="#">▶ 539</a> ]      | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_duration</a> [ <a href="#">▶ 539</a> ] | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_function</a> [ <a href="#">▶ 539</a> ] | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_rinse</a> [ <a href="#">▶ 539</a> ]    | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_spin</a> [ <a href="#">▶ 540</a> ]     | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_wash</a> [ <a href="#">▶ 540</a> ]     | Genutzt durch: SNVT_clothes_w_c |

**Structure/AuxiliaryStructure/SNVT\_clothes\_w\_s**

| Datentypen   | Beschreibung                    |
|--|---------------------------------|
| <a href="#">ST_LON_alarm</a> [ <a href="#">▶ 540</a> ] | Genutzt durch: SNVT_clothes_w_s |

**Structure/AuxiliaryStructure/SNVT\_color\_2**

| Datentypen   | Beschreibung                |
|--|-----------------------------|
| <a href="#">ST_LON_CIE1931_lumen</a> [ <a href="#">▶ 542</a> ]   | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_CIE1931_percent</a> [ <a href="#">▶ 542</a> ] | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_color_value</a> [ <a href="#">▶ 543</a> ]     | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_RGB</a> [ <a href="#">▶ 543</a> ]             | Genutzt durch: SNVT_color_2 |

**Structure/AuxiliaryStructure/SNVT\_ctrl\_resp**

| Datentypen            | Beschreibung                  |
|-----------------------|-------------------------------|
| ST_LON_range [▶ 543]  | Genutzt durch: SNVT_ctrl_resp |
| ST_LON_sender [▶ 544] | Genutzt durch: SNVT_ctrl_resp |

**Structure/AuxiliaryStructure/SNVT\_dev\_fault**

| Datentypen                | Beschreibung                  |
|---------------------------|-------------------------------|
| ST_LON_Dev_type1 [▶ 544]  | Genutzt durch: SNVT_dev_fault |
| ST_LON_pump_ctrl1 [▶ 544] | Genutzt durch: SNVT_dev_fault |
| ST_LON_valve_pos1 [▶ 545] | Genutzt durch: SNVT_dev_fault |

**Structure/AuxiliaryStructure/SNVT\_dev\_maint**

| Datentypen                | Beschreibung                  |
|---------------------------|-------------------------------|
| ST_LON_Dev_type2 [▶ 546]  | Genutzt durch: SNVT_dev_maint |
| ST_LON_pump_ctrl2 [▶ 546] | Genutzt durch: SNVT_dev_maint |
| ST_LON_valve_pos2 [▶ 547] | Genutzt durch: SNVT_dev_maint |

**Structure/AuxiliaryStructure/SNVT\_dev\_status**

| Datentypen                | Beschreibung                   |
|---------------------------|--------------------------------|
| ST_LON_Dev_type3 [▶ 547]  | Genutzt durch: SNVT_dev_status |
| ST_LON_pump_ctrl3 [▶ 548] | Genutzt durch: SNVT_dev_status |
| ST_LON_valve_pos3 [▶ 549] | Genutzt durch: SNVT_dev_status |

**Structure/AuxiliaryStructure/SNVT\_ex\_control**

| Datentypen                         | Beschreibung                   |
|------------------------------------|--------------------------------|
| ST_LON_Control_device_addr [▶ 549] | Genutzt durch: SNVT_ex_control |

**Structure/AuxiliaryStructure/SNVT\_file\_req**

| Datentypen                  | Beschreibung                 |
|-----------------------------|------------------------------|
| ST_LON_addrt [▶ 550]        | Genutzt durch: SNVT_file_req |
| ST_LON_dest_address [▶ 550] | Genutzt durch: SNVT_file_req |
| ST_LON_gp [▶ 550]           | Genutzt durch: SNVT_file_req |
| ST_LON_sn [▶ 551]           | Genutzt durch: SNVT_file_req |

**Structure/AuxiliaryStructure/SNVT\_file\_status**

| Datentypen                | Beschreibung   |
|---------------------------|--|
| ST_LON_address [▶ 551]    | Genutzt durch: FB_Write_Address_Table /<br>FB_Read_Address_Table |
| ST_LON_adr [▶ 551]        | Genutzt durch: SNVT_file_status                                  |
| ST_LON_descriptor [▶ 552] | Genutzt durch: SNVT_file_status                                  |

**Structure/AuxiliaryStructure/SNVT\_lamp\_status**

| Datentypen                  | Beschreibung                    |
|-----------------------------|---------------------------------|
| ST_LON_Alarm_actual [▶ 552] | Genutzt durch: SNVT_lamp_status |

| Datentypen  | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_alarm_previous</a> [ <a href="#">▶ 554</a> ] | Genutzt durch: SNVT_lamp_status |

**Structure/AuxiliaryStructure/SNVT\_pos\_ctrl**

| Datentypen  | Beschreibung                 |
|---|------------------------------|
| <a href="#">ST_LON_abspos</a> [ <a href="#">▶ 556</a> ] | Genutzt durch: SNVT_pos_ctrl |
| <a href="#">ST_LON_Value</a> [ <a href="#">▶ 556</a> ]  | Genutzt durch: SNVT_pos_ctrl |

**Structure/AuxiliaryStructure/SNVT\_rac\_ctrl**

| Datentypen   | Beschreibung                                |
|--|---|
| <a href="#">ST_LON_addr_dest</a> [ <a href="#">▶ 556</a> ] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_addr_init</a> [ <a href="#">▶ 557</a> ] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_addr_talk</a> [ <a href="#">▶ 557</a> ] | Genutzt durch: SNVT_rac_ctrl                |
| <a href="#">ST_LON_p2m</a> [ <a href="#">▶ 557</a> ]       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_p2p</a> [ <a href="#">▶ 557</a> ]       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |

**Structure/AuxiliaryStructure/SNVT\_rac\_req**

| Datentypen   | Beschreibung |
|--|--------------|
| <a href="#">ST_LON_rac_req_addr_dest</a> [ <a href="#">▶ 558</a> ] |              |
| <a href="#">ST_LON_rac_req_addr_init</a> [ <a href="#">▶ 558</a> ] |              |

**Structure/AuxiliaryStructure/SNVT\_switch\_2**

| Datentypen   | Beschreibung                 |
|--|------------------------------|
| <a href="#">ST_LON_setting</a> [ <a href="#">▶ 559</a> ] | Genutzt durch: SNVT_switch_2 |

**Structure/AuxiliaryStructure/SNVT\_time\_zone**

| Datentypen   | Beschreibung                  |
|--|-------------------------------|
| <a href="#">ST_LON_end_DST</a> [ <a href="#">▶ 559</a> ]     | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_M_end_DST</a> [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_M_start_DST</a> [ <a href="#">▶ 560</a> ] | Genutzt durch: SNVT_time_zone |
| <a href="#">ST_LON_start_DST</a> [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |

**Structure**

| Datentypen  | Beschreibung  |
|---|---|
| <a href="#">ST_KL6401</a> [ <a href="#">▶ 563</a> ]               | Struktur für die Konfiguration                                |
| <a href="#">ST_LON_AddressTable</a> [ <a href="#">▶ 564</a> ]     | Genutzt durch: FB_Write_Address_Table / FB_Read_Address_Table |
| <a href="#">ST_LON_ConfigTable</a> [ <a href="#">▶ 564</a> ]      | Genutzt durch: FB_Write_Config_Table / FB_Read_Config_Table   |
| <a href="#">ST_LON_DomainTable</a> [ <a href="#">▶ 565</a> ]      | Genutzt durch: FB_Write_Domain_Table / FB_Read_Domain_Table   |
| <a href="#">ST_LON_SNVT_alarm</a> [ <a href="#">▶ 565</a> ]       | Genutzt durch: SNVT_alarm                                     |
| <a href="#">ST_LON_SNVT_alarm_2</a> [ <a href="#">▶ 566</a> ]     | Genutzt durch: SNVT_alarm_2                                   |
| <a href="#">ST_LON_SNVT_chlr_status</a> [ <a href="#">▶ 566</a> ] | Genutzt durch: SNVT_chlr_status                               |
| <a href="#">ST_LON_SNVT_clothes_w_a</a> [ <a href="#">▶ 566</a> ] | Genutzt durch: SNVT_clothes_w_a                               |

| <b>Datentypen</b>                                  | <b>Beschreibung</b>                |
|--|------------------------------------|
| <a href="#">ST_LON_SNVT_clothes_w_c</a> [► 568]    | Genutzt durch: SNVT_clothes_w_c    |
| <a href="#">ST_LON_SNVT_clothes_w_m</a> [► 568]    | Genutzt durch: SNVT_clothes_w_m    |
| <a href="#">ST_LON_SNVT_clothes_w_s</a> [► 569]    | Genutzt durch: SNVT_clothes_w_s    |
| <a href="#">ST_LON_SNVT_color</a> [► 569]          | Genutzt durch: SNVT_color          |
| <a href="#">ST_LON_SNVT_color_2</a> [► 569]        | Genutzt durch: SNVT_color_2        |
| <a href="#">ST_LON_SNVT_ctrl_req</a> [► 569]       | Genutzt durch: SNVT_ctrl_req       |
| <a href="#">ST_LON_SNVT_ctrl_resp</a> [► 570]      | Genutzt durch: SNVT_ctrl_resp      |
| <a href="#">ST_LON_SNVT_currency</a> [► 570]       | Genutzt durch: SNVT_currency       |
| <a href="#">ST_LON_SNVT_date_event</a> [► 570]     | Genutzt durch: SNVT_date_event     |
| <a href="#">ST_LON_SNVT_dev_fault</a> [► 570]      | Genutzt durch: SNVT_dev_fault      |
| <a href="#">ST_LON_SNVT_dev_maint</a> [► 571]      | Genutzt durch: SNVT_dev_maint      |
| <a href="#">ST_LON_SNVT_dev_status</a> [► 571]     | Genutzt durch: SNVT_dev_status     |
| <a href="#">ST_LON_SNVT_earth_pos</a> [► 571]      | Genutzt durch: SNVT_earth_pos      |
| <a href="#">ST_LON_SNVT_elapsed_tm</a> [► 572]     | Genutzt durch: SNVT_elapsed_tm     |
| <a href="#">ST_LON_SNVT_ent_status</a> [► 572]     | Genutzt durch: SNVT_ent_status     |
| <a href="#">ST_LON_SNVT_environment</a> [► 573]    | Genutzt durch: SNVT_environment    |
| <a href="#">ST_LON_SNVT_ex_control</a> [► 574]     | Genutzt durch: SNVT_ex_control     |
| <a href="#">ST_LON_SNVT_file_pos</a> [► 574]       | Genutzt durch: SNVT_file_pos       |
| <a href="#">ST_LON_SNVT_file_req</a> [► 574]       | Genutzt durch: SNVT_file_req       |
| <a href="#">ST_LON_SNVT_file_status</a> [► 575]    | Genutzt durch: SNVT_file_status    |
| <a href="#">ST_LON_SNVT_geo_loc</a> [► 575]        | Genutzt durch: SNVT_geo_loc        |
| <a href="#">ST_LON_SNVT_hvac_overid</a> [► 575]    | Genutzt durch: SNVT_hvac_overid    |
| <a href="#">ST_LON_SNVT_hvac_satsts</a> [► 575]    | Genutzt durch: SNVT_hvac_satsts    |
| <a href="#">ST_LON_SNVT_hvac_status</a> [► 576]    | Genutzt durch: SNVT_hvac_status    |
| <a href="#">ST_LON_SNVT_lamp_status</a> [► 577]    | Genutzt durch: SNVT_lamp_status    |
| <a href="#">ST_LON_SNVT_log_fx_request</a> [► 577] | Genutzt durch: SNVT_log_fx_request |
| <a href="#">ST_LON_SNVT_log_fx_status</a> [► 577]  | Genutzt durch: SNVT_log_fx_status  |
| <a href="#">ST_LON_SNVT_log_status</a> [► 578]     | Genutzt durch: SNVT_log_status     |
| <a href="#">ST_LON_SNVT_muldiv</a> [► 578]         | Genutzt durch: SNVT_muldiv         |
| <a href="#">ST_LON_SNVT_nv_type</a> [► 578]        | Genutzt durch: SNVT_nv_type        |
| <a href="#">ST_LON_SNVT_obj_request</a> [► 579]    | Genutzt durch: SNVT_obj_request    |
| <a href="#">ST_LON_SNVT_obj_status</a> [► 579]     | Genutzt durch: SNVT_obj_status     |
| <a href="#">ST_LON_SNVT_pos_ctrl</a> [► 580]       | Genutzt durch: SNVT_pos_ctrl       |
| <a href="#">ST_LON_SNVT_preset</a> [► 581]         | Genutzt durch: SNVT_preset         |
| <a href="#">ST_LON_SNVT_privacyzone</a> [► 581]    | Genutzt durch: SNVT_privacyzone    |
| <a href="#">ST_LON_SNVT_ptz</a> [► 582]            | Genutzt durch: SNVT_ptz            |
| <a href="#">ST_LON_SNVT_pump_sensor</a> [► 582]    | Genutzt durch: SNVT_pump_sensor    |
| <a href="#">ST_LON_SNVT_pumpset_mn</a> [► 583]     | Genutzt durch: SNVT_pumpset_mn     |
| <a href="#">ST_LON_SNVT_pumpset_sn</a> [► 583]     | Genutzt durch: SNVT_pumpset_sn     |
| <a href="#">ST_LON_SNVT_rac_ctrl</a> [► 584]       | Genutzt durch: SNVT_rac_ctrl       |
| <a href="#">ST_LON_SNVT_rac_req</a> [► 584]        | Genutzt durch: SNVT_rac_req        |
| <a href="#">ST_LON_SNVT_reg_val</a> [► 585]        | Genutzt durch: SNVT_rac_val        |
| <a href="#">ST_LON_SNVT_reg_val_ts</a> [► 585]     | Genutzt durch: SNVT_rac_val_ts     |

| Datentypen  | Beschreibung   |
|---|--|
| <a href="#">ST_LON_SNVT_sblnd_state [▶ 585]</a>   | Genutzt durch: SNVT_sblnd_state  |
| <a href="#">ST_LON_SNVT_scene [▶ 586]</a>         | Genutzt durch: SNVT_scene  |
| <a href="#">ST_LON_SNVT_scene_cfg [▶ 586]</a>     | Genutzt durch: SNVT_scene_cfg  |
| <a href="#">ST_LON_SNVT_setting [▶ 586]</a>       | Genutzt durch: SNVT_setting  |
| <a href="#">ST_LON_SNVT_str_int [▶ 587]</a>       | Genutzt durch: SNVT_str_int  |
| <a href="#">ST_LON_SNVT_switch [▶ 587]</a>        | Genutzt durch: SNVT_switch   |
| <a href="#">ST_LON_SNVT_switch_2 [▶ 587]</a>      | Genutzt durch: SNVT_switch_2   |
| <a href="#">ST_LON_SNVT_temp_setpt [▶ 587]</a>    | Genutzt durch: SNVT_temp_setpt   |
| <a href="#">ST_LON_SNVT_time_zone [▶ 588]</a>     | Genutzt durch: SNVT_time_zone  |
| <a href="#">ST_LON_SNVT_tod_event [▶ 588]</a>     | Genutzt durch: SNVT_tod_event  |
| <a href="#">ST_LON_SNVT_trans_table [▶ 589]</a>   | Genutzt durch: SNVT_trans_table  |
| <a href="#">ST_LON_SNVT_zerospan [▶ 589]</a>      | Genutzt durch: SNVT_zerospan   |
| <a href="#">str_AddressTable [▶ 589]</a>          | Adress Tabelle   |
| Datentypen  | Beschreibung   |
| <a href="#">E_LON_ERROR [▶ 467]</a>               | Fehlermeldungen  |
| <a href="#">ST_ExplicitMessage [▶ 561]</a>        | Explicit Message   |
| <a href="#">ST_LON_Communication [▶ 562]</a>      | Verbindung zwischen "FB_LON_KL6401" und den Sende-/ Empfangsbausteinen |
| <a href="#">ST_LON_ParameterInterface [▶ 562]</a> | LON Parameterinterface   |
| <a href="#">ST_LON_WriteData [▶ 563]</a>          | Struktur des Sendepuffer   |
| <a href="#">ST_Prm [▶ 563]</a>                    | Struktur für Konfiguration   |

### 7.3.1 E\_LON\_ERROR

Fehlermeldungen der Bibliothek



Der NV Index in der SPS wird beim Senden nicht mit dem NV Index (Spalte Id) in der KS2000 verglichen. Stimmen beide Indizes nicht überein, können falsche/ungültige Werte gesendet werden.

Senden ohne Binding gibt keine Fehlermeldung aus.

```

TYPE E_LON_ERROR :
(
  eLON_no_Error                := 0,
  eLON_Value_out_of_range     := 1,
  eLON_Terminal_not_ready     := 2,
  eLON_Wrong_SNVT_Typ        := 3,
  eLON_Wrong_wNVIndex        := 4,
  eKL6401_Wrong_Terminal     := 5,
  eKL6401_Error              := 6,
  eKL6401_Terminal_is_not_initialized := 7,

  eLON_L_star_Out_of_range   := 50,
  eLON_A_star_Out_of_range   := 51,
  eLON_B_star_Out_of_range   := 52,

  eLON_eRequest_Out_of_range := 55,

  eLON_wYear_Out_of_range    := 66,
  eLON_wMonth_Out_of_range   := 67,
  eLON_wDay_Out_of_range     := 68,
  eLON_wHour_Out_of_range    := 69,
  eLON_wMinute_Out_of_range  := 70,
  eLON_wSecond_Out_of_range  := 71,
  eLON_wMillisecond_Out_of_range := 72,

  eLON_rZero_Out_of_range    := 80,
  eLON_rSpan_Out_of_range    := 81,

```

```

eLON_arrValue01_Out_of_range := 85,
eLON_arrValue02_Out_of_range := 86,
eLON_arrValue03_Out_of_range := 87,
eLON_arrValue04_Out_of_range := 88,
eLON_arrValue05_Out_of_range := 89,
eLON_arrValue06_Out_of_range := 90,
eLON_arrValue07_Out_of_range := 91,
eLON_arrValue08_Out_of_range := 92,
eLON_arrValue09_Out_of_range := 93,

eLON_arrValue10_Out_of_range := 100,
eLON_arrValue11_Out_of_range := 101,
eLON_arrValue12_Out_of_range := 102,
eLON_arrValue13_Out_of_range := 103,
eLON_arrValue14_Out_of_range := 104,
eLON_arrValue15_Out_of_range := 105,
eLON_arrValue16_Out_of_range := 106,
eLON_arrValue17_Out_of_range := 107,
eLON_arrValue18_Out_of_range := 108,
eLON_arrValue19_Out_of_range := 109,

eLON_arrValue20_Out_of_range := 115,
eLON_arrValue21_Out_of_range := 116,
eLON_arrValue22_Out_of_range := 117,
eLON_arrValue23_Out_of_range := 118,
eLON_arrValue24_Out_of_range := 119,
eLON_arrValue25_Out_of_range := 120,
eLON_arrValue26_Out_of_range := 121,
eLON_arrValue27_Out_of_range := 122,
eLON_arrValue28_Out_of_range := 123,
eLON_arrValue29_Out_of_range := 124,

eLON_arrValue30_Out_of_range := 130,
eLON_arrValue31_Out_of_range := 131,
eLON_arrValue32_Out_of_range := 132,
eLON_arrValue33_Out_of_range := 133,
eLON_arrValue34_Out_of_range := 134,
eLON_arrValue35_Out_of_range := 135,
eLON_arrValue36_Out_of_range := 136,
eLON_arrValue37_Out_of_range := 137,
eLON_arrValue38_Out_of_range := 138,
eLON_arrValue39_Out_of_range := 139,
eLON_arrValue40_Out_of_range := 140,

eLON_087uiDay_Out_of_range := 145,
eLON_087uiHour_Out_of_range := 146,
eLON_087uiMinute_Out_of_range := 147,
eLON_087uiSecond_Out_of_range := 148,
eLON_087uiMillisecond_Out_of_range := 149,

eLON_ePriority_level_Out_of_range := 155,
eLON_eAlarm_type_Out_of_range := 156,

eLON_Currency_Out_of_range := 160,

eLON_diRw_ptr_Out_of_range := 165,

eLON_Object_request_Out_of_range := 170,

eLON_094eLearn_Out_of_range := 175,
eLON_094uiHour_Out_of_range := 176,
eLON_094uiMinute_Out_of_range := 177,
eLON_094uiSecond_Out_of_range := 178,
eLON_094uiMillisecond_Out_of_range := 179,

eLON_095rValue_Out_of_range := 185,
eLON_095siState_Out_of_range := 186,

eLON_byInterp_pts_0_to_1_Out_of_range := 190,
eLON_byInterp_pts_1_to_2_Out_of_range := 191,
eLON_byInterp_pts_2_to_3_Out_of_range := 192,
eLON_byInterp_pts_3_to_4_Out_of_range := 193,
eLON_byInterp_pts_4_to_5_Out_of_range := 194,
eLON_byInterp_pts_5_to_6_Out_of_range := 195,
eLON_byInterp_pts_6_to_0_Out_of_range := 196,

eLON_rOccupied_cool_Out_of_range := 200,
eLON_rStandby_cool_Out_of_range := 201,
eLON_rUnoccupied_cool_Out_of_range := 202,

```



```

eLON_rOccupied_heat_Out_of_range := 203,
eLON_rStandby_heat_Out_of_range := 204,
eLON_rUnoccupied_heat_Out_of_range := 205,

eLON_111rPercent_Out_of_range := 210,
eLON_111eState_Out_of_range := 211,

eLON_eMode_Out_of_range := 215,
eLON_rHeat_output_primary_Out_of_range := 216,
eLON_rHeat_output_secondary_Out_of_range := 217,
eLON_rCool_output_Out_of_range := 218,
eLON_rEcon_output_Out_of_range := 219,
eLON_rFan_output_Out_of_range := 220,

eLON_115eFunction_Out_of_range := 225,

eLON_eFunction_Out_of_range := 226,
eLON_rSetting_Out_of_range := 227,
eLON_rRotation_Out_of_range := 228,
eLON_rFade_time_Out_of_range := 229,
eLON_rDelay_time_Out_of_range := 230,

eLON_eChlr_run_mode_Out_of_range := 235,
eLON_echlr_op_mode_Out_of_range := 236,

eLON_eNext_state_Out_of_range := 240,
eLON_eCurrent_state_Out_of_range := 241,

eLON_diSecond_time_offset_Out_of_range := 245,
eLON_eType_of_description_Out_of_range := 246,
eLON_byHour_of_start_DST_Out_of_range := 250,
eLON_byMinute_of_start_DST_Out_of_range := 251,
eLON_bySecond_of_start_DST_Out_of_range := 252,
eLON_byHour_of_end_DST_Out_of_range := 260,
eLON_byMinute_of_end_DST_Out_of_range := 261,
eLON_bySecond_of_end_DST_Out_of_range := 262,
eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range := 263,
eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range := 264,
eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range := 265,
eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range := 266,
eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range := 267,
eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range := 268,
eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range := 269,
eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range := 270,
eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range := 271,
eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range := 272,

eLON_byLatitude_deg_Out_of_range := 280,
eLON_rLatitude_min_Out_of_range := 281,
eLON_bylongitude_deg_Out_of_range := 282,
eLON_rLongitude_min_Out_of_range := 283,

eLON_byNr_decimals_Out_of_range := 290,
eLON_eUnit_Out_of_range := 291,

eLON_137eUnit_Out_of_range := 295,
eLON_137byNr_decimals_Out_of_range := 296,
eLON_137byStatus_Out_of_range := 297,
eLON_137uiYear_Out_of_range := 298,
eLON_137uiMonth_Out_of_range := 299,
eLON_137uiDay_Out_of_range := 300,
eLON_137uiHour_Out_of_range := 301,
eLON_137uiMinute_Out_of_range := 302,
eLON_137uiSecond_Out_of_range := 303,

eLON_bySender_prio_Out_of_range := 310,

eLON_eStatus_Out_of_range := 315,
eLON_stSender_uiID_Out_of_range := 316,
eLON_stSender_stRange_uiLower_Out_of_range := 317,
eLON_stSender_stRange_uiUpper_Out_of_range := 318,
eLON_uiController_id_Out_of_range := 319,

eLON_ePan_dir_Out_of_range := 325,
eLON_rPan_speed_Out_of_range := 326,
eLON_eTilt_dir_Out_of_range := 327,
eLON_rTilt_speed_Out_of_range := 328,
eLON_eZoom_Out_of_range := 329,
eLON_rZoom_speed_Out_of_range := 330,

```



```

eLON_eAction_Out_of_range           := 335,

eLON_byController_prio_Out_of_range := 340,
eLON_152eFunction_Out_of_range      := 341,
eLON_152eAction_Out_of_range        := 342,
eLON_stValue_stAbspos_rZoom_Out_of_range := 343,
eLON_stValue_stAbspos_rTilt_Out_of_range := 344,
eLON_stValue_stAbspos_rPan_Out_of_range := 345,

eLON_eMain_pump_Out_of_range        := 350,
eLON_eBooster_pump_Out_of_range     := 351,
eLON_ePriority_level_Out_of_range    := 352,
eLON_eProcess_ready_Out_of_range    := 353,
eLON_eEmergency_stop_activated_Out_of_range := 354,
eLON_eMain_pump_drive_enabled_Out_of_range := 355,
eLON_eBooster_pump_drive_enabled_Out_of_range := 356,
eLON_eMaintenance_required_Out_of_range := 357,

eLON_eControl_status_Out_of_range   := 365,
eLON_stControl_device_addr_byDomain_length_Out_of_range := 366,
eLON_stControl_device_addr_bySubnet_Out_of_range := 367,
eLON_stControl_device_addr_byNode_Out_of_range := 368,

eLON_rExhaust_temperature_Out_of_range := 375,
eLON_rExhaust_pressure_Out_of_range   := 376,
eLON_rShaft_seal_purge_pressure_Out_of_range := 377,
eLON_rSupply_voltage_Out_of_range     := 378,
eLON_eCoolant_flow_low_Out_of_range   := 379,
eLON_eDilution_active_Out_of_range   := 380,
eLON_eBallast_dilution_active_Out_of_range := 381,
eLON_eInlet_purge_dilution_active_Out_of_range := 382,
eLON_eExhaust_dilution_active_Out_of_range := 383,
eLON_eDilution_flow_Out_of_range     := 384,
eLON_ePower_supply_on_Out_of_range    := 385,

eLON_rRotational_speed_Out_of_range  := 390,
eLON_rBody_temperature_Out_of_range  := 391,
eLON_rMotor_external_temperature_Out_of_range := 392,
eLON_rMotor_internal_temperature_Out_of_range := 393,
eLON_eMotor_overloaded_Out_of_range  := 394,
eLON_eOil_level_low_Out_of_range     := 395,
eLON_ePhase_imbalance_detected_Out_of_range := 396,
eLON_rCurrent_usage_Out_of_range     := 397,
eLON_rPower_usage_Out_of_range       := 398,
eLON_eTemperature_control_Out_of_range := 399,
eLON_eElectromagnetic_brake_active_Out_of_range := 400,
eLON_eFriction_brake_active_Out_of_range := 401,
eLON_eGas_brake_active_Out_of_range  := 402,

eLON_164iMilliseconds_Out_of_range   := 410,
eLON_164ePriority_level_Out_of_range  := 411,
eLON_164eAlarm_type_Out_of_range     := 412,

eLON_byType_scope_Out_of_range       := 420,
eLON_uiType_index_Out_of_range       := 421,
eLON_eType_category_Out_of_range     := 422,
eLON_byType_length_Out_of_range      := 423,

eLON_eCmd_fb_Out_of_range            := 430,

eLON_byManufacturer_Out_of_range     := 435,

eLON_eDevice_select_Out_of_range     := 440,

eLON_stPos_eFunction_Out_of_range    := 445,
eLON_stPos_rSetting_Out_of_range     := 446,
eLON_stPos_rRotation_Out_of_range    := 447,
eLON_eCmd_source_Out_of_range       := 448,
eLON_eError_code_Out_of_range       := 449,

eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range := 455,
eLON_181stAddr_talk_byCar_id_Out_of_range := 456,
eLON_181stAddr_talk_byLocation_Out_of_range := 457,
eLON_181stAddr_talk_byUnit_id_Out_of_range := 458,
eLON_181stAddr_init_eAudio_sensor_type_Out_of_range := 459,
eLON_181stAddr_init_byCar_id_Out_of_range := 450,
eLON_181stAddr_init_byLocation_Out_of_range := 461,
eLON_181stAddr_init_byUnit_id_Out_of_range := 462,
eLON_181eAudio_type_Out_of_range     := 463,
eLON_181byAudio_line_Out_of_range    := 464,

```

```

eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range := 465,
eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range           := 466,
eLON_181stAddr_dest_stP2p_byLocation_Out_of_range        := 467,
eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range         := 468,

eLON_stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range  := 475,
eLON_stAddr_dest_stP2p_byCar_id_Out_of_range             := 476,
eLON_stAddr_dest_stP2p_byLocation_Out_of_range           := 477,
eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range           := 478,
eLON_stAddr_init_eAudio_sensor_type_Out_of_range         := 479,
eLON_stAddr_init_byCar_id_Out_of_range                   := 480,
eLON_stAddr_init_byLocation_Out_of_range                 := 481,
eLON_stAddr_init_byUnit_id_Out_of_range                  := 482,
eLON_eAudio_type_Out_of_range                            := 483,

eLON_eCycle_Out_of_range                                 := 490,
eLON_eSubcycle_Out_of_range                              := 491,
eLON_stFunction_eProgram_Out_of_range                    := 492,
eLON_stFunction_stWash_eLoad_level_Out_of_range         := 493,
eLON_stFunction_stWash_ePrewash_Out_of_range            := 494,
eLON_stFunction_stRinse_eOption_Out_of_range            := 495,
eLON_stFunction_stRinse_byRepeat_Out_of_range           := 496,
eLON_stFunction_stSpin_eHold_Out_of_range                := 497,
eLON_stFunction_stDry_byTemp_Out_of_range                := 498,
eLON_stFunction_stDry_stDuration_eDryness_Out_of_range  := 499,

eLON_186eCycle_Out_of_range                              := 505,
eLON_186eSubcycle_Out_of_range                          := 506,
eLON_stWasher_command_data_eCycle_Out_of_range          := 507,
eLON_stWasher_command_data_eSubcycle_Out_of_range       := 508,
eLON_stWasher_command_data_stFunction_eProgram_Out_of_range := 509,
eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range := 510,
eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range := 511,
eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range := 512,
eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range := 513,
eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range := 514,
eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range := 515,
eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range := 516,

eLON_eState_Out_of_range                                 := 518,
eLON_stSetting_rValue_Out_of_range                      := 519,
eLON_stSetting_rChange_Out_of_range                    := 520,
eLON_stSetting_rMultiplier_Out_of_range                := 521,
eLON_stSetting_iAngle_Out_of_range                     := 522,
eLON_stSetting_byGroup_number_Out_of_range              := 523,
eLON_stSetting_siFan_level_Out_of_range                 := 524,

eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range     := 525,
eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range     := 526,
eLON_stColor_value_stCIE1931_lumen_udiAbsolute_Y_Out_of_range := 527,
eLON_stColor_value_stCIE1931_percent_rX_Out_of_range   := 528,
eLON_stColor_value_stCIE1931_percent_rY_Out_of_range   := 529,
eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range := 530,
eLON_stColor_value_uiColor_temperature_Out_of_range    := 531,

eLON_191eStatus_Out_of_range                            := 535,
eLON_uiLog_number_Out_of_range                          := 536,
eLON_rLevel_Out_of_range                                := 537,
eLON_stCurrent_notify_time_rHundredths_Out_of_range   := 538,
eLON_stPrevious_notify_time_rHundredths_Out_of_range  := 539,

eLON_rHundredths_Out_of_range                           := 545,

eLON_stStart_time_rHundredths_Out_of_range              := 550,
eLON_stEnd_time_rHundredths_Out_of_range                := 551,

eLON_rComplete_Out_of_range                             := 565,

eLON_stTime_actual_rHundredths_Out_of_range             := 570,
eLON_stTime_previous_rHundredths_Out_of_range           := 571,

eLON_lrEnergy_Out_of_range                              := 585,
eLON_rPowerFactor_Out_of_range                          := 586,
eLON_rPower_Out_of_range                                := 587,
eLON_rBallastTemp_Out_of_range                          := 588,

eLON_lrLongitude_Out_of_range                           := 595,

```

```
eLON_lrLatitude_Out_of_range           := 596
)
END_TYPE
```

eLON\_no\_Error: Es steht kein Fehler an.

eLON\_Value\_out\_of\_range: Die Eingangsvariable "Value" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. "Value" kann in verschiedenen Formaten vorliegen und besitzt dann einen entsprechenden Präfix (z.B. LREAL = lrValue).

eLON\_Terminal\_not\_ready: Der Baustein "FB\_LON\_KL6401" durchläuft beim Starten der SPS eine Initialisierungsschrittfolge (Klemmentyp abfragen, Firmware abfragen u.s.w.). So lange diese nicht fertig gemeldet hat, wird diese Meldung ausgegeben. Steht der Fehler nach einem SPS Reset an, muss der Controller einmal spannungslos gemacht werden.

eLON\_Wrong\_SNVT\_Typ: Der empfangene SNVT Typ entspricht nicht dem SNVT Typ des angesprochenen NV Index (Eingangsvariable "wId")

eLON\_Wrong\_wNVIndex: Falscher NV Index.

eKL6401\_Wrong\_Terminal: Es wurde keine KL6401 erkannt.

eKL6401\_Error: Der Baustein "FB\_LON\_KL6401" hat einen Fehler. Der Fehlercode steht am Ausgang "dwErrorKL" an.

eKL6401\_Terminal\_is\_not\_initialized: Die Klemme ist nicht initialisiert. Üblicherweise gibt es bei dieser Meldung keine Verbindung zur Klemme. Klemme im System Manager mit den Variablen verknüpft? Klemme falsch gesteckt? Alles bereinigen, alles Übersetzen und im System Manager neu eingelesen?

eLON\_L\_star\_Out\_of\_range: SNVT 70 / Die Eingangsvariable "stValue.L\_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_A\_star\_Out\_of\_range: SNVT 70 / Die Eingangsvariable "stValue.A\_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_B\_star\_Out\_of\_range: SNVT 70 / Die Eingangsvariable "stValue.B\_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eRequest\_Out\_of\_range: SNVT 73 / Die Eingangsvariable "stValue.eRequest" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wYear\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wYear" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wMonth\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wMonth" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wDay\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wHour\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wMinute\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wSecond\_Out\_of\_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_wMillisecond\_Out\_of\_range: SNVT 088 / Die Eingangsvariable "stValue.wMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rZero\_Out\_of\_range: SNVT 085 / Die Eingangsvariable "stValue.rZero" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rSpan\_Out\_of\_range: SNVT 085 / Die Eingangsvariable "stValue.rSpan" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue01\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[1]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue02\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[2]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue03\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[3]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue04\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[4]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue05\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[5]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue06\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[6]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue07\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[7]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue08\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[8]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue09\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[9]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue10\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[10]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue11\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[11]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue12\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[12]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue13\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[13]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue14\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[14]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue15\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[15]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue16\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[16]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue17\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[17]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue18\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[18]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue19\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[19]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue20\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[20]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue21\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[21]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue22\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[22]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue23\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[23]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue24\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[24]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue25\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[25]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue26\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[26]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue27\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[27]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue28\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[28]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue29\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[29]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue30\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[30]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue31\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[31]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue32\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[32]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue33\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[33]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue34\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[34]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue35\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[35]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue36\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[36]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue37\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[37]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue38\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[38]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue39\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[39]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_arrValue40\_Out\_of\_range: SNVT 086 / Die Eingangsvariable "arrValue[40]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_087uiDay\_Out\_of\_range: SNVT 087 / Die Eingangsvariable "stValue.uiDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_087uiHour\_Out\_of\_range: SNVT 087 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_087uiMinute\_Out\_of\_range: SNVT 087 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_087uiSecond\_Out\_of\_range: SNVT 087 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_087uiMillisecond\_Out\_of\_range: SNVT 087 / Die Eingangsvariable "stValue.uiMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_ePriority\_level\_Out\_of\_range: SNVT 088 / Die Eingangsvariable "stValue.ePriority\_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.



eLON\_eAlarm\_type\_Out\_of\_range: SNVT 088 / Die Eingangsvariable "stValue.eAlarm\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_Currency\_Out\_of\_range: SNVT 089 / Die Eingangsvariable "stValue.Currency" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_diRw\_ptr\_Out\_of\_range: SNVT 090 / Die Eingangsvariable "stValue.diRw\_ptr" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_Object\_request\_Out\_of\_range: SNVT 092 / Die Eingangsvariable "stValue.Object\_request" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_094eLearn\_Out\_of\_range: SNVT 094 / Die Eingangsvariable "stValue.eLearn" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_094uiHour\_Out\_of\_range: SNVT 094 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_094uiMinute\_Out\_of\_range: SNVT 094 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_094uiSecond\_Out\_of\_range: SNVT 094 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_094uiMillisecond\_Out\_of\_range: SNVT 094 / Die Eingangsvariable "stValue.uiMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_095rValue\_Out\_of\_range: SNVT 095 / Die Eingangsvariable "stValue.rValue" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_095siState\_Out\_of\_range: SNVT 095 / Die Eingangsvariable "stValue.siState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_0\_to\_1\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_0\_to\_1" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_1\_to\_2\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_1\_to\_2" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_2\_to\_3\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_2\_to\_3" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_3\_to\_4\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_3\_to\_4" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_4\_to\_5\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_4\_to\_5" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_5\_to\_6\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_5\_to\_6" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byInterp\_pts\_6\_to\_0\_Out\_of\_range: SNVT 096 / Die Eingangsvariable "stValue.byInterp\_pts\_6\_to\_0" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rOccupied\_cool\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rOccupied\_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rStandby\_cool\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rStandby\_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rUnoccupied\_cool\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rUnoccupied\_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rOccupied\_heat\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rOccupied\_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rStandby\_heat\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rStandby\_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rUnoccupied\_heat\_Out\_of\_range: SNVT 106 / Die Eingangsvariable "stValue.rUnoccupied\_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_111rPercent\_Out\_of\_range: SNVT 111 / Die Eingangsvariable "stValue.rPercent" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_111eState\_Out\_of\_range: SNVT 111 / Die Eingangsvariable "stValue.eState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eMode\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.eMode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rHeat\_output\_primary\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.rHeat\_output\_primary" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rHeat\_output\_secondary\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.rHeat\_output\_secondary" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rCool\_output\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.rCool\_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rEcon\_output\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.rEcon\_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rFan\_output\_Out\_of\_range: SNVT 112 / Die Eingangsvariable "stValue.rFan\_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_115eFunction\_Out\_of\_range: SNVT 115 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eFunction\_Out\_of\_range: SNVT 116 / 117 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rSetting\_Out\_of\_range: SNVT 116 / 117 / Die Eingangsvariable "stValue.rSetting" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rRotation\_Out\_of\_range: SNVT 116 / 117 / Die Eingangsvariable "stValue.rRotation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rFade\_time\_Out\_of\_range: SNVT 116 / Die Eingangsvariable "stValue.rFade\_time" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rDelay\_time\_Out\_of\_range: SNVT 116 / Die Eingangsvariable "stValue.rDelay\_time" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eChlr\_run\_mode\_Out\_of\_range: SNVT 127 / Die Eingangsvariable "stValue.eChlr\_run\_mode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_echlr\_op\_mode\_Out\_of\_range: SNVT 127 / Die Eingangsvariable "stValue.echlr\_op\_mode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eNext\_state\_Out\_of\_range: SNVT 128 / Die Eingangsvariable "stValue.eNext\_state" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eCurrent\_state\_Out\_of\_range: SNVT 128 / Die Eingangsvariable "stValue.eCurrent\_state" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_diSecond\_time\_offset\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.diSecond\_time\_offset" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eType\_of\_description\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.eType\_of\_description" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byHour\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.byHour\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byMinute\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.byMinute\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_bySecond\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.bySecond\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byHour\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.byHour\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byMinute\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.byMinute\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_bySecond\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.bySecond\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_DST\_uiG\_day\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stStart\_DST.uiG\_day\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_DST\_uiJ\_day\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stStart\_DST.uiJ\_day\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_DST\_stM\_start\_DST\_byMonth\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stStart\_DST.stM\_start\_DST.byMonth\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_DST\_stM\_start\_DST\_byWeek\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stStart\_DST.stM\_start\_DST.byWeek\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_DST\_stM\_start\_DST\_eDateday\_of\_start\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stStart\_DST.stM\_start\_DST.eDateday\_of\_start\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_DST\_uiG\_day\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stEnd\_DST.uiG\_day\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_DST\_uiJ\_day\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stEnd\_DST.uiJ\_day\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_DST\_stM\_end\_DST\_byMonth\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stEnd\_DST.stM\_end\_DST.byMonth\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_DST\_stM\_end\_DST\_byWeek\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stEnd\_DST.stM\_end\_DST.byWeek\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_DST\_stM\_end\_DST\_eDateday\_of\_end\_DST\_Out\_of\_range: SNVT 134 / Die Eingangsvariable "stValue.stEnd\_DST.stM\_end\_DST.eDateday\_of\_end\_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byLatitude\_deg\_Out\_of\_range: SNVT 135 / Die Eingangsvariable "stValue.byLatitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rLatitude\_min\_Out\_of\_range: SNVT 135 / Die Eingangsvariable "stValue.rLatitude\_min" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_bylongitude\_deg\_Out\_of\_range: SNVT 135 / Die Eingangsvariable "stValue.bylongitude\_deg" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rLongitude\_min\_Out\_of\_range: SNVT 135 / Die Eingangsvariable "stValue.rLongitude\_min" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byNr\_decimals\_Out\_of\_range: SNVT 136 / Die Eingangsvariable "stValue.byNr\_decimals" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.



eLON\_eUnit\_Out\_of\_range: SNVT 136 / Die Eingangsvariable "stValue.eUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137eUnit\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.eUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137byNr\_decimals\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.byNr\_decimals" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137byStatus\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.byStatus" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiYear\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiYear" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiMonth\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiMonth" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiDay\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiHour\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiMinute\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_137uiSecond\_Out\_of\_range: SNVT 137 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_bySender\_prio\_Out\_of\_range: SNVT 148 / Die Eingangsvariable "stValue.bySender\_prio" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eStatus\_Out\_of\_range: SNVT 149 / Die Eingangsvariable "stValue.eStatus" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSender\_uiID\_Out\_of\_range: SNVT 149 / Die Eingangsvariable "stValue.stSender.uiID" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSender\_stRange\_uiLower\_Out\_of\_range: SNVT 149 / Die Eingangsvariable "stValue.stSender.stRange.uiLower" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSender\_stRange\_uiUpper\_Out\_of\_range: SNVT 149 / Die Eingangsvariable "stValue.stSender.stRange.uiUpper" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_uiController\_id\_Out\_of\_range: SNVT 149 / Die Eingangsvariable "stValue.uiController" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_ePan\_dir\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rPan\_speed\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eTilt\_dir\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rTilt\_speed\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eZoom\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rZoom\_speed\_Out\_of\_range: SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eAction\_Out\_of\_range: SNVT 151 / Die Eingangsvariable "stValue.eAction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byController\_prio\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.byController" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_152eFunction\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_152eAction\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.eAction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stValue\_stAbspos\_rZoom\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rZoom" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stValue\_stAbspos\_rTilt\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rTilt" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stValue\_stAbspos\_rPan\_Out\_of\_range: SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rPan" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eMain\_pump\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eMain\_pump" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eBooster\_pump\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eBooster\_pump" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_ePriority\_level\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.ePriority\_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eProcess\_ready\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eProcess\_ready" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eEmergency\_stop\_activated\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eEmergency\_stop\_activated" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eMain\_pump\_drive\_enabled\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eMain\_pump\_drive\_enabled" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eBooster\_pump\_drive\_enabled\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eBooster\_pump\_drive\_enabled" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eMaintenance\_required\_Out\_of\_range: SNVT 156 / Die Eingangsvariable "stValue.eMaintenance\_required" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eControl\_status\_Out\_of\_range: SNVT 157 / Die Eingangsvariable "stValue.eControl\_status" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stControl\_device\_addr\_byDomain\_length\_Out\_of\_range: SNVT 157 / Die Eingangsvariable "stValue.stControl\_device\_addr.byDomain\_length" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stControl\_device\_addr\_bySubnet\_Out\_of\_range: SNVT 157 / Die Eingangsvariable "stValue.stControl\_device\_addr.bySubnet" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stControl\_device\_addr\_byNode\_Out\_of\_range: SNVT 157 / Die Eingangsvariable "stValue.stControl\_device\_addr.byNode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rExhaust\_temperature\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.rExhaust\_temperature" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rExhaust\_pressure\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.rExhaust\_pressure" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rShaft\_seal\_purge\_pressure\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.rShaft\_seal\_purge\_pressure" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rSupply\_voltage\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.rSupply\_voltage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eCoolant\_flow\_low\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eCoolant\_flow\_low" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eDilution\_active\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eDilution\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eBallast\_dilution\_active\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eBallast\_dilution\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eInlet\_purge\_dilution\_active\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eInlet\_purge\_dilution\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eExhaust\_dilution\_active\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eExhaust\_dilution\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eDilution\_flow\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.eDilution\_flow" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_ePower\_supply\_on\_Out\_of\_range: SNVT 158 / Die Eingangsvariable "stValue.ePower\_supply\_on" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rRotational\_speed\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.rRotational\_speed" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rBody\_temperature\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.rBody" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rMotor\_external\_temperature\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.rMotor\_external\_temperature" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rMotor\_internal\_temperature\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.eMotor\_overloaded" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eMotor\_overloaded\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eOil\_level\_low\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.ePhase\_imbalance\_detected" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_ePhase\_imbalance\_detected\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rCurrent\_usage\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.rCurrent\_usage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rPower\_usage\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.Power\_usage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eTemperature\_control\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.eElectromagnetic\_brake\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eElectromagnetic\_brake\_active\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eFriction\_brake\_active\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.eFriction\_brake\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eGas\_brake\_active\_Out\_of\_range: SNVT 159 / Die Eingangsvariable "stValue.eGas\_brake\_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_164iMilliseconds\_Out\_of\_range: SNVT 164 / Die Eingangsvariable "stValue.iMilliseconds" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_164ePriority\_level\_Out\_of\_range: SNVT 164 / Die Eingangsvariable "stValue.ePriority\_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_164eAlarm\_type\_Out\_of\_range: SNVT 164 / Die Eingangsvariable "stValue.eAlarm" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byType\_scope\_Out\_of\_range: SNVT 166 / Die Eingangsvariable "stValue.byType\_scope" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_uiType\_index\_Out\_of\_range: SNVT 166 / Die Eingangsvariable "stValue.uiType\_index" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eType\_category\_Out\_of\_range: SNVT 166 / Die Eingangsvariable "stValue.eType\_category" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byType\_length\_Out\_of\_range: SNVT 166 / Die Eingangsvariable "stValue.byType" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eCmd\_fb\_Out\_of\_range: SNVT 170 / Die Eingangsvariable "stValue.eCmd\_fb" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_byManufacturer\_Out\_of\_range: SNVT 172 / Die Eingangsvariable "stValue.byManufacturer" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eDevice\_select\_Out\_of\_range: SNVT 175 / Die Eingangsvariable "stValue.eDevice\_select" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stPos\_eFunction\_Out\_of\_range: SNVT 180 / Die Eingangsvariable "stValue.stPos\_eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stPos\_rSetting\_Out\_of\_range: SNVT 180 / Die Eingangsvariable "stValue.stPos.rSetting" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stPos\_rRotation\_Out\_of\_range: SNVT 180 / Die Eingangsvariable "stValue.stPos.rRotation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eCmd\_source\_Out\_of\_range: SNVT 180 / Die Eingangsvariable "stValue.eCmd\_source" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eError\_code\_Out\_of\_range: SNVT 180 / Die Eingangsvariable "stValue.\_eError\_code" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_talk\_eAudio\_sensor\_type\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_talk.eAudio\_sensor" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_talk\_byCar\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_talk.byCar\_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_talk\_byLocation\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_talk.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_talk\_byUnit\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_talk.byUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_init\_eAudio\_sensor\_type\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_init.eAudio\_sensor\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_init\_byCar\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_init.byCar" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_init\_byLocation\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_init.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_init\_byUnit\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_init.byUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181eAudio\_type\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.eAudio\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181byAudio\_line\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.byAudio\_line" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_dest\_stP2p\_eAudio\_sensor\_type\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.eAudio\_sensor\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_dest\_stP2p\_byCar\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_dest\_stP2p\_byLocation\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_181stAddr\_dest\_stP2p\_byUnit\_id\_Out\_of\_range: SNVT 181 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.byUnit\_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_dest\_stP2p\_eAudio\_sensor\_type\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.eAudio\_sensor\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_dest\_stP2p\_byCar\_id\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.byCar" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_dest\_stP2p\_byLocation\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_dest.stP2p.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_dest\_stP2p\_byUnit\_id\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_init.eAudio\_sensor\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_init\_eAudio\_sensor\_type\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_init.byCar\_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_init\_byCar\_id\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_init\_byLocation\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_init.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stAddr\_init\_byUnit\_id\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.stAddr\_init.byUnit\_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eAudio\_type\_Out\_of\_range: SNVT 182 / Die Eingangsvariable "stValue.eAudio\_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eCycle\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eSubcycle\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_eProgram\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.eProgram" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.



eLON\_stFunction\_stWash\_eLoad\_level\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stWash\_eLoad\_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stWash\_ePrewash\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stWash.ePrewash" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stRinse\_eOption\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stRinse.eOption" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stRinse\_byRepeat\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stRinse.byRepeat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stSpin\_eHold\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stSpin.eHold" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stDry\_byTemp\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stDry.byTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stFunction\_stDry\_stDuration\_eDryness\_Out\_of\_range: SNVT 184 / Die Eingangsvariable "stValue.stFunction.stDry.stDuration.eDryness" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_186eCycle\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_186eSubcycle\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_eCycle\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_eSubcycle\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_eProgram\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.eProgram" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stWash\_eLoad\_level\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stWash.eLoad" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stWash\_ePrewash\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stWash.ePrewash" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stRinse\_eOption\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stRinse.eOption" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stRinse\_byRepeat\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stRinse.byRepeat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stSpin\_eHold\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stSpin.eHold" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stDry\_byTemp\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stDry.byTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stWasher\_command\_data\_stFunction\_stDry\_stDuration\_eDryness\_Out\_of\_range: SNVT 186 / Die Eingangsvariable "stValue.stWasher\_command\_data.stFunction.stDry.stDuration.eDryness" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_eState\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.eState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_rValue\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.rValue" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_rChange\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.rChange" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_rMultiplier\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.rMultiplier" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_iAngle\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.iAngle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_byGroup\_number\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.byGroup\_number" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stSetting\_siFan\_level\_Out\_of\_range: SNVT 189 / Die Eingangsvariable "stValue.stSettings.siFan\_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_lumen\_rX\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_lumen.rX" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_lumen\_rY\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_lumen.rY" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_lumen\_udiAbsolute\_Y\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_lumen.udiAbsolute\_Y" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_percent\_rX\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_percent.rX" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_percent\_rY\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_percent.rY" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_stCIE1931\_percent\_rPercent\_Y\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.stCIE1931\_percent.rPercent\_Y" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stColor\_value\_uiColor\_temperature\_Out\_of\_range: SNVT 190 / Die Eingangsvariable "stValue.stColor\_value.uiColor\_temperature" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_191eStatus\_Out\_of\_range: SNVT 191 / Die Eingangsvariable "stValue.Status" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_uiLog\_number\_Out\_of\_range: SNVT 191 / Die Eingangsvariable "stValue.uiLog\_number" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rLevel\_Out\_of\_range: SNVT 191 / Die Eingangsvariable "stValue.rLevel" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stCurrent\_notify\_time\_rHundredths\_Out\_of\_range: SNVT 191 / Die Eingangsvariable "stValue.stCurrent\_notify\_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stPrevious\_notify\_time\_rHundredths\_Out\_of\_range: SNVT 191 / Die Eingangsvariable "stValue.stPrevious\_notify\_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rHundredths\_Out\_of\_range: SNVT 192 / Die Eingangsvariable "stValue.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stStart\_time\_rHundredths\_Out\_of\_range: SNVT 193 / Die Eingangsvariable "stValue.stStart\_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stEnd\_time\_rHundredths\_Out\_of\_range: SNVT 193 / Die Eingangsvariable "stValue.stEnd\_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rComplete\_Out\_of\_range: SNVT 194 / Die Eingangsvariable "stValue.rComplete" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stTime\_actual\_rHundredths\_Out\_of\_range: SNVT 199 / Die Eingangsvariable "stValue.stTime\_actual.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_stTime\_previous\_rHundredths\_Out\_of\_range: SNVT 199 / Die Eingangsvariable "stValue.stTime\_previous.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_lrEnergy\_Out\_of\_range: SNVT 200 / Die Eingangsvariable "stValue.lrEnergy" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rPowerFactor\_Out\_of\_range: SNVT 200 / Die Eingangsvariable "stValue.rPowerFactor" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rPower\_Out\_of\_range: SNVT 200 / Die Eingangsvariable "stValue.rPower" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_rBallastTemp\_Out\_of\_range: SNVT 200 / Die Eingangsvariable "stValue.rBallastTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_lrLongitude\_Out\_of\_range: SNVT 201 / Die Eingangsvariable "stValue.lrLongitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

eLON\_lrLatitude\_Out\_of\_range: SNVT 201 / Die Eingangsvariable "stValue.lrLatitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.

## 7.3.2 E\_LON\_Parameter\_Datatypes

### Enums SNVT Typen

```

TYPE E_LON_Parameter_Datatypes :
(
  eEmpty           := 0,
  eSNVT_amp        := 1,
  eSNVT_amp_mil    := 2,
  eSNVT_angle      := 3,
  eSNVT_angle_vel  := 4,
  eSNVT_btu_kilo   := 5,
  eSNVT_btu_mega   := 6,
  eSNVT_char_ascii := 7,
  eSNVT_count      := 8,
  eSNVT_count_inc  := 9,
  eSNVT_date_cal   := 10,
  eSNVT_date_day   := 11,
  eSNVT_date_time  := 12,
  eSNVT_elec_kwh   := 13,
  eSNVT_elec_whr   := 14,
  eSNVT_flow       := 15,
  eSNVT_flow_mil   := 16,
  eSNVT_length     := 17,
  eSNVT_length_kilo := 18,
  eSNVT_length_micr := 19,
  eSNVT_length_mil := 20,
  eSNVT_lev_cont   := 21,
  eSNVT_lev_disc   := 22,

```



```

eSNVT_mass      := 23,
eSNVT_mass_kilo := 24,
eSNVT_mass_mega := 25,
eSNVT_mass_mil  := 26,
eSNVT_power     := 27,
eSNVT_power_kilo := 28,
eSNVT_ppm       := 29,
eSNVT_press     := 30,
eSNVT_res       := 31,
eSNVT_res_kilo  := 32,
eSNVT_sound_db  := 33,
eSNVT_speed     := 34,
eSNVT_speed_mil := 35,
eSNVT_str_asc   := 36,
eSNVT_str_int   := 37,
eSNVT_telcom    := 38,
eSNVT_temp      := 39,
eSNVT_time_passed := 40,
eSNVT_vol       := 41,
eSNVT_vol_kilo  := 42,
eSNVT_vol_mil   := 43,
eSNVT_volt      := 44,
eSNVT_volt_dbmv := 45,
eSNVT_volt_kilo := 46,
eSNVT_volt_mil  := 47,
eSNVT_amp_f     := 48,
eSNVT_angle_f   := 49,
eSNVT_angle_vel_f := 50,
eSNVT_count_f   := 51,
eSNVT_count_inc_f := 52,
eSNVT_flow_f    := 53,
eSNVT_length_f  := 54,
eSNVT_lev_cont_f := 55,
eSNVT_mass_f    := 56,
eSNVT_power_f   := 57,
eSNVT_ppm_f     := 58,
eSNVT_press_f   := 59,
eSNVT_res_f     := 60,
eSNVT_sound_db_f := 61,
eSNVT_speed_f   := 62,
eSNVT_temp_f    := 63,
eSNVT_time_f    := 64,
eSNVT_vol_f     := 65,
eSNVT_volt_f    := 66,
eSNVT_btu_f     := 67,
eSNVT_elec_whr_f := 68,
eSNVT_config_src := 69,
eSNVT_color     := 70,
eSNVT_grammage  := 71,
eSNVT_grammage_f := 72,
eSNVT_file_req  := 73,
eSNVT_file_status := 74,
eSNVT_freq_f    := 75,
eSNVT_freq_hz   := 76,
eSNVT_freq_kilohz := 77,
eSNVT_freq_milhz := 78,
eSNVT_lux       := 79,
eSNVT_ISO_7811 := 80,
eSNVT_lev_percent := 81,
eSNVT_multiplier := 82,
eSNVT_state     := 83,
eSNVT_time_stamp := 84,
eSNVT_zerospans := 85,
eSNVT_magcard   := 86,
eSNVT_elapsed_tm := 87,
eSNVT_alarm     := 88,
eSNVT_currency  := 89,
eSNVT_file_pos  := 90,
eSNVT_muldiv    := 91,
eSNVT_obj_request := 92,
eSNVT_obj_status := 93,
eSNVT_preset    := 94,
eSNVT_switch    := 95,
eSNVT_trans_table := 96,
eSNVT_override  := 97,
eSNVT_pwr_fact  := 98,
eSNVT_pwr_fact_f := 99,
eSNVT_density   := 100,
eSNVT_density_f := 101,
eSNVT_rpm       := 102,

```

```

eSNVT_hvac_emerg      := 103,
eSNVT_angle_deg      := 104,
eSNVT_temp_p         := 105,
eSNVT_temp_setpt     := 106,
eSNVT_time_sec       := 107,
eSNVT_hvac_mode      := 108,
eSNVT_occupancy      := 109,
eSNVT_area           := 110,
eSNVT_hvac_overid    := 111,
eSNVT_hvac_status    := 112,
eSNVT_press_p        := 113,
eSNVT_address        := 114,
eSNVT_scene          := 115,
eSNVT_scene_cfg      := 116,
eSNVT_setting        := 117,
eSNVT_evap_state     := 118,
eSNVT_therm_mode     := 119,
eSNVT_defr_mode      := 120,
eSNVT_defr_term      := 121,
eSNVT_defr_state     := 122,
eSNVT_time_min       := 123,
eSNVT_time_hour      := 124,
eSNVT_ph             := 125,
eSNVT_ph_f           := 126,
eSNVT_chlr_status    := 127,
eSNVT_tod_event      := 128,
eSNVT_smo_obscur     := 129,
eSNVT_fire_test      := 130,
eSNVT_temp_ror       := 131,
eSNVT_fire_init      := 132,
eSNVT_fire_indctc    := 133,
eSNVT_time_zone      := 134,
eSNVT_earth_pos      := 135,
eSNVT_reg_val        := 136,
eSNVT_reg_val_ts     := 137,
eSNVT_volt_ac        := 138,
eSNVT_amp_ac         := 139,

eSNVT_turbidity      := 143,
eSNVT_turbidity_f    := 144,
eSNVT_hvac_type       := 145,
eSNVT_elec_kwh_l     := 146,
eSNVT_temp_diff_p    := 147,
eSNVT_ctrl_req       := 148,
eSNVT_ctrl_resp      := 149,
eSNVT_ptz            := 150,
eSNVT_privacyzone    := 151,
eSNVT_pos_ctrl       := 152,
eSNVT_enthalpy       := 153,
eSNVT_gfci_status    := 154,
eSNVT_motor_state    := 155,
eSNVT_pumpset_mn     := 156,
eSNVT_ex_control     := 157,
eSNVT_pumpset_sn     := 158,
eSNVT_pump_sensor    := 159,
eSNVT_abs_humid      := 160,
eSNVT_flow_p         := 161,
eSNVT_dev_c_mode     := 162,
eSNVT_valve_mode     := 163,
eSNVT_alarm_2       := 164,
eSNVT_state_64       := 165,
eSNVT_nv_type        := 166,

eSNVT_ent_opmode     := 168,
eSNVT_ent_state      := 169,
eSNVT_ent_status     := 170,
eSNVT_flow_dir       := 171,
eSNVT_hvac_satsts    := 172,
eSNVT_dev_status     := 173,
eSNVT_dev_fault      := 174,
eSNVT_dev_maint      := 175,
eSNVT_date_event     := 176,
eSNVT_sched_val      := 177,
eSNVT_sec_state      := 178,
eSNVT_sec_status     := 179,
eSNVT_sbldnd_state   := 180,
eSNVT_rac_ctrl       := 181,
eSNVT_rac_req        := 182,
eSNVT_count_32       := 183,
eSNVT_clothes_w_c    := 184,

```

```

eSNVT_clothes_w_m      := 185,
eSNVT_clothes_w_s      := 186,
eSNVT_clothes_w_a      := 187,
eSNVT_multiplier_s     := 188,
eSNVT_switch_2        := 189,
eSNVT_color_2         := 190,
eSNVT_log_status       := 191,
eSNVT_time_stamp_p     := 192,
eSNVT_log_fx_request   := 193,
eSNVT_log_fx_status    := 194,
eSNVT_log_request      := 195,
eSNVT_enthalpy_d       := 196,
eSNVT_amp_ac_mil       := 197,
eSNVT_time_hour_p      := 198,
eSNVT_lamp_status      := 199,
eSNVT_environment      := 200,
eSNVT_geo_loc          := 201
)
END_TYPE

```

eEmpty:

eSNVT\_amp: SNVT\_amp

eSNVT\_amp\_mil: SNVT\_amp\_mil

eSNVT\_angle: SNVT\_angle

eSNVT\_angle\_vel: SNVT\_angle\_vel

eSNVT\_btu\_kilo: SNVT\_btu\_kilo

eSNVT\_btu\_mega: SNVT\_btu\_mega

eSNVT\_char\_ascii: SNVT\_char\_ascii

eSNVT\_count: SNVT\_count

eSNVT\_count\_inc: SNVT\_count\_inc

eSNVT\_date\_cal: SNVT\_date\_cal

eSNVT\_date\_day: SNVT\_date\_day

eSNVT\_date\_time: SNVT\_date\_time

eSNVT\_elec\_kwh: SNVT\_elec\_kwh

eSNVT\_elec\_whr: SNVT\_elec\_whr

eSNVT\_flow: SNVT\_flow

eSNVT\_flow\_mil: SNVT\_flow\_mil

eSNVT\_length: SNVT\_length

eSNVT\_length\_kilo: SNVT\_length\_kilo

eSNVT\_length\_micr: SNVT\_length\_micr

eSNVT\_length\_mil: SNVT\_length\_mil

eSNVT\_lev\_cont: SNVT\_lev\_cont

eSNVT\_lev\_disc: SNVT\_lev\_disc

eSNVT\_mass: SNVT\_mass

eSNVT\_mass\_kilo: SNVT\_mass\_kilo

eSNVT\_mass\_mega: SNVT\_mass\_mega

eSNVT\_mass\_mil: SNVT\_mass\_mil

eSNVT\_power: SNVT\_power  
eSNVT\_power\_kilo: SNVT\_power\_kilo  
eSNVT\_ppm: SNVT\_ppm  
eSNVT\_press: SNVT\_press  
eSNVT\_res: SNVT\_res  
eSNVT\_res\_kilo: SNVT\_res\_kilo  
eSNVT\_sound\_db: SNVT\_sound\_db  
eSNVT\_speed: SNVT\_speed  
eSNVT\_speed\_mil: SNVT\_speed\_mil  
eSNVT\_str\_asc: SNVT\_str\_asc  
eSNVT\_str\_int: SNVT\_str\_int  
eSNVT\_telcom: SNVT\_telcom  
eSNVT\_temp: SNVT\_temp  
eSNVT\_time\_passed: SNVT\_time\_passed  
eSNVT\_vol: SNVT\_vol  
eSNVT\_vol\_kilo: SNVT\_vol\_kilo  
eSNVT\_vol\_mil: SNVT\_vol\_mil  
eSNVT\_volt: SNVT\_volt  
eSNVT\_volt\_dbmv: SNVT\_volt\_dbmv  
eSNVT\_volt\_kilo: SNVT\_volt\_kilo  
eSNVT\_volt\_mil: SNVT\_volt\_mil  
eSNVT\_amp\_f: SNVT\_amp\_f  
eSNVT\_angle\_f: SNVT\_angle\_f  
eSNVT\_angle\_vel\_f: SNVT\_angle\_vel\_f  
eSNVT\_count\_f: SNVT\_count\_f  
eSNVT\_count\_inc\_f: SNVT\_count\_inc\_f  
eSNVT\_flow\_f: SNVT\_flow\_f  
eSNVT\_length\_f: SNVT\_length\_f  
eSNVT\_lev\_cont\_f: SNVT\_lev\_cont\_f  
eSNVT\_mass\_f: SNVT\_mass\_f  
eSNVT\_power\_f: SNVT\_power\_f  
eSNVT\_ppm\_f: SNVT\_ppm\_f  
eSNVT\_press\_f: SNVT\_press\_f  
eSNVT\_res\_f: SNVT\_res\_f  
eSNVT\_sound\_db\_f: SNVT\_sound\_db\_f  
eSNVT\_speed\_f: SNVT\_speed\_f

eSNVT\_temp\_f: SNVT\_temp\_f  
eSNVT\_time\_f: SNVT\_time\_f  
eSNVT\_vol\_f: SNVT\_vol\_f  
eSNVT\_volt\_f: SNVT\_volt\_f  
eSNVT\_btu\_f: SNVT\_btu\_f  
eSNVT\_elec\_whr\_f: SNVT\_elec\_whr\_f  
eSNVT\_config\_src: SNVT\_config\_src  
eSNVT\_color: SNVT\_color  
eSNVT\_grammage: SNVT\_grammage  
eSNVT\_grammage\_f: SNVT\_grammage\_f  
eSNVT\_file\_req: SNVT\_file\_req  
eSNVT\_file\_status: SNVT\_file\_status  
eSNVT\_freq\_f: SNVT\_freq\_f  
eSNVT\_freq\_hz: SNVT\_freq\_hz  
eSNVT\_freq\_kilohz: SNVT\_freq\_kilohz  
eSNVT\_freq\_milhz: SNVT\_freq\_milhz  
eSNVT\_lux: SNVT\_lux  
eSNVT\_ISO\_7811: SNVT\_ISO\_7811  
eSNVT\_lev\_percent: SNVT\_lev\_percent  
eSNVT\_multiplier: SNVT\_multiplier  
eSNVT\_state: SNVT\_state  
eSNVT\_time\_stamp: SNVT\_time\_stamp  
eSNVT\_zerospan: SNVT\_zerospan  
eSNVT\_magcard: SNVT\_magcard  
eSNVT\_elapsed\_tm: SNVT\_elapsed\_tm  
eSNVT\_alarm: SNVT\_alarm  
eSNVT\_currency: SNVT\_currency  
eSNVT\_file\_pos: SNVT\_file\_pos  
eSNVT\_muldiv: SNVT\_muldiv  
eSNVT\_obj\_request: SNVT\_obj\_request  
eSNVT\_obj\_status: SNVT\_obj\_status  
eSNVT\_preset: SNVT\_preset  
eSNVT\_switch: SNVT\_switch  
eSNVT\_trans\_table: SNVT\_trans\_table  
eSNVT\_override: SNVT\_override  
eSNVT\_pwr\_fact: SNVT\_pwr\_fact

eSNVT\_pwr\_fact\_f: SNVT\_pwr\_fact\_f  
eSNVT\_density: SNVT\_density  
eSNVT\_density\_f: SNVT\_density\_f  
eSNVT\_rpm: SNVT\_rpm  
eSNVT\_hvac\_emerg: SNVT\_hvac\_emerg  
eSNVT\_angle\_deg: SNVT\_angle\_deg  
eSNVT\_temp\_p: SNVT\_temp\_p  
eSNVT\_temp\_setpt: SNVT\_temp\_setpt  
eSNVT\_time\_sec: SNVT\_time\_sec  
eSNVT\_hvac\_mode: SNVT\_hvac\_mode  
eSNVT\_occupancy: SNVT\_occupancy  
eSNVT\_area: SNVT\_area  
eSNVT\_hvac\_overid: SNVT\_hvac\_overid  
eSNVT\_hvac\_status: SNVT\_hvac\_status  
eSNVT\_press\_p: SNVT\_press\_p  
eSNVT\_address: SNVT\_address  
eSNVT\_scene: SNVT\_scene  
eSNVT\_scene\_cfg: SNVT\_scene\_cfg  
eSNVT\_setting: SNVT\_setting  
eSNVT\_evap\_state: SNVT\_evap\_state  
eSNVT\_therm\_mode: SNVT\_therm\_mode  
eSNVT\_defr\_mode: SNVT\_defr\_mode  
eSNVT\_defr\_term: SNVT\_defr\_term  
eSNVT\_defr\_state: SNVT\_defr\_state  
eSNVT\_time\_min: SNVT\_time\_min  
eSNVT\_time\_hour: SNVT\_time\_hour  
eSNVT\_ph: SNVT\_ph  
eSNVT\_ph\_f: SNVT\_ph\_f  
eSNVT\_chlr\_status: SNVT\_chlr\_status  
eSNVT\_tod\_event: SNVT\_tod\_event  
eSNVT\_smo\_obscur: SNVT\_smo\_obscur  
eSNVT\_fire\_test: SNVT\_fire\_test  
eSNVT\_temp\_ror: SNVT\_temp\_ror  
eSNVT\_fire\_init: SNVT\_fire\_init  
eSNVT\_fire\_indcte: SNVT\_fire\_indcte  
eSNVT\_time\_zone: SNVT\_time\_zone

eSNVT\_earth\_pos: SNVT\_earth\_pos  
eSNVT\_reg\_val: SNVT\_reg\_val  
eSNVT\_reg\_val\_ts: SNVT\_reg\_val\_ts  
eSNVT\_volt\_ac: SNVT\_volt\_ac  
eSNVT\_amp\_ac: SNVT\_amp\_ac  
eSNVT\_turbidity: SNVT\_turbidity  
eSNVT\_turbidity\_f: SNVT\_turbidity\_f  
eSNVT\_hvac\_type: SNVT\_hvac\_type  
eSNVT\_elec\_kwh\_l: SNVT\_elec\_kwh\_l  
eSNVT\_temp\_diff\_p: SNVT\_temp\_diff\_p  
eSNVT\_ctrl\_req: SNVT\_ctrl\_req  
eSNVT\_ctrl\_resp: SNVT\_ctrl\_resp  
eSNVT\_ptz: SNVT\_ptz  
eSNVT\_privacyzone: SNVT\_privacyzone  
eSNVT\_pos\_ctrl: SNVT\_pos\_ctrl  
eSNVT\_enthalpy: SNVT\_enthalpy  
eSNVT\_gfci\_status: SNVT\_gfci\_status  
eSNVT\_motor\_state: SNVT\_motor\_state  
eSNVT\_pumpset\_mn: SNVT\_pumpset\_mn  
eSNVT\_ex\_control: SNVT\_ex\_control  
eSNVT\_pumpset\_sn: SNVT\_pumpset\_sn  
eSNVT\_pump\_sensor: SNVT\_pump\_sensor  
eSNVT\_abs\_humid: SNVT\_abs\_humid  
eSNVT\_flow\_p: SNVT\_flow\_p  
eSNVT\_dev\_c\_mode: SNVT\_dev\_c\_mode  
eSNVT\_valve\_mode: SNVT\_valve\_mode  
eSNVT\_alarm\_2: SNVT\_alarm\_2  
eSNVT\_state\_64: SNVT\_state\_64  
eSNVT\_nv\_type: SNVT\_nv\_type  
eSNVT\_ent\_opmode: SNVT\_ent\_opmode  
eSNVT\_ent\_state: SNVT\_ent\_state  
eSNVT\_ent\_status: SNVT\_ent\_status  
eSNVT\_flow\_dir: SNVT\_flow\_dir  
eSNVT\_hvac\_satsts: SNVT\_hvac\_satsts  
eSNVT\_dev\_status: SNVT\_dev\_status  
eSNVT\_dev\_fault: SNVT\_dev\_fault

eSNVT\_dev\_maint: SNVT\_dev\_maint  
eSNVT\_date\_event: SNVT\_date\_event  
eSNVT\_sched\_val: SNVT\_sched\_val  
eSNVT\_sec\_state: SNVT\_sec\_state  
eSNVT\_sec\_status: SNVT\_sec\_status  
eSNVT\_sblnd\_state: SNVT\_sblnd\_state  
eSNVT\_rac\_ctrl: SNVT\_rac\_ctrl  
eSNVT\_rac\_req: SNVT\_rac\_req  
eSNVT\_count\_32: SNVT\_count\_32  
eSNVT\_clothes\_w\_c: SNVT\_clothes\_w\_c  
eSNVT\_clothes\_w\_m: SNVT\_clothes\_w\_m  
eSNVT\_clothes\_w\_s: SNVT\_clothes\_w\_s  
eSNVT\_clothes\_w\_a: SNVT\_clothes\_w\_a  
eSNVT\_multiplier\_s: SNVT\_multiplier\_s  
eSNVT\_switch\_2: SNVT\_switch\_2  
eSNVT\_color\_2: SNVT\_color\_2  
eSNVT\_log\_status: SNVT\_log\_status  
eSNVT\_time\_stamp\_p: SNVT\_time\_stamp\_p  
eSNVT\_log\_fx\_request: SNVT\_log\_fx\_request  
eSNVT\_log\_fx\_status: SNVT\_log\_fx\_status  
eSNVT\_log\_request: SNVT\_log\_request  
eSNVT\_enthalpy\_d: SNVT\_enthalpy\_d  
eSNVT\_amp\_ac\_mil: SNVT\_amp\_ac\_mil  
eSNVT\_time\_hour\_p: SNVT\_time\_hour\_p  
eSNVT\_lamp\_status: SNVT\_lamp\_status  
eSNVT\_environment: SNVT\_environment  
eSNVT\_geo\_loc: SNVT\_geo\_loc

### 7.3.3 E\_LON\_alarm\_type\_t

Genutzt durch: SNVT\_alarm / SNVT\_alarm\_2

```
TYPE E_LON_alarm_type_t :  
(  
    eLON_AL_HEADER           := -13,  
    eLON_AL_FOOTER           := -12,  
    eLON_AL_DEBUG            := -11,  
    eLON_AL_INFO             := -10,  
    eLON_AL_SYSTEM_INFO      := -6,  
    eLON_AL_VALUE_INVALID    := -5,  
    eLON_AL_CONSTANT         := -4,  
    eLON_AL_OFFLINE          := -3,  
    eLON_AL_UNKNOWN          := -2,  
    eLON_AL_NUL              := -1,  
    eLON_AL_NO_CONDITION     := 0,  
    eLON_AL_ALM_CONDITION    := 1,  
    eLON_AL_TOT_SVC_ALM_1    := 2,  
)
```



```
eLON_AL_TOT_SVC_ALM_2 := 3,
eLON_AL_TOT_SVC_ALM_3 := 4,
eLON_AL_LOW_LMT_CLR_1 := 5,
eLON_AL_LOW_LMT_CLR_2 := 6,
eLON_AL_HIGH_LMT_CLR_1 := 7,
eLON_AL_HIGH_LMT_CLR_2 := 8,
eLON_AL_LOW_LMT_ALM_1 := 9,
eLON_AL_LOW_LMT_ALM_2 := 10,
eLON_AL_HIGH_LMT_ALM_1 := 11,
eLON_AL_HIGH_LMT_ALM_2 := 12,
eLON_AL_FIR_ALM := 13,
eLON_AL_FIR_PRE_ALM := 14,
eLON_AL_FIR_TREB := 15,
eLON_AL_FIR_SUPV := 16,
eLON_AL_FIR_TEST_ALM := 17,
eLON_AL_FIR_TEST_PRE_ALM := 18,
eLON_AL_FIR_ENVCOMP_MAX := 19,
eLON_AL_FIR_MONITOR_COND := 20,
eLON_AL_FIR_MAINT_ALERT := 21,
eLON_AL_FATAL_ERROR := 30,
eLON_AL_ERROR := 31,
eLON_AL_WARNING := 32
)
END_TYPE
```

eLON\_AL\_HEADER: Update sequence header

eLON\_AL\_FOOTER: Update sequence footer

eLON\_AL\_DEBUG: Debug information (not an alarm)

eLON\_AL\_INFO: Information update (not an alarm)

eLON\_AL\_SYSTEM\_INFO: System information (not an alarm)

eLON\_AL\_VALUE\_INVALID: The value is invalid

eLON\_AL\_CONSTANT: The value is a constant value (not an alarm)

eLON\_AL\_OFFLINE: The device is offline

eLON\_AL\_UNKNOWN: Alarm condition unknown (may be due to a communication failure or hardware failure)

eLON\_AL\_NUL: Invalid alarm type value (alarm condition not specified)

eLON\_AL\_NO\_CONDITION: No alarm condition present

eLON\_AL\_ALM\_CONDITION: Unspecified alarm condition present

eLON\_AL\_TOT\_SVC\_ALM\_1: Total/service interval alarm 1 (component requires service or maintenance)

eLON\_AL\_TOT\_SVC\_ALM\_2: Total/service interval alarm 2

eLON\_AL\_TOT\_SVC\_ALM\_3: Total/service interval alarm 3

eLON\_AL\_LOW\_LMT\_CLR\_1: Alarm low limit alarm clear 1

eLON\_AL\_LOW\_LMT\_CLR\_2: Alarm low limit alarm clear 2

eLON\_AL\_HIGH\_LMT\_CLR\_1: Alarm high limit alarm clear 1

eLON\_AL\_HIGH\_LMT\_CLR\_2: Alarm high limit alarm clear 2

eLON\_AL\_LOW\_LMT\_ALM\_1: Alarm low limit alarm 1

eLON\_AL\_LOW\_LMT\_ALM\_2: Alarm low limit alarm 2

eLON\_AL\_HIGH\_LMT\_ALM\_1: Alarm high limit alarm 1

eLON\_AL\_HIGH\_LMT\_ALM\_2: Alarm high limit alarm 2

eLON\_AL\_FIR\_ALM: Fire alarm condition

eLON\_AL\_FIR\_PRE\_ALM: Fire pre-alarm condition

eLON\_AL\_FIR\_TRBL: Fire-related trouble (fault) condition

eLON\_AL\_FIR\_SUPV: Fire-related supervisory condition (e.g., sprinkler pressure)

eLON\_AL\_FIR\_TEST\_ALM: Fire-related test-mode alarm condition

eLON\_AL\_FIR\_TEST\_PRE\_ALM: Fire-related test-mode pre-alarm condition

eLON\_AL\_FIR\_ENVCOMP\_MAX: Fire-related maximum environmental compensation level reached

eLON\_AL\_FIR\_MONITOR\_COND: Fire-related abnormal input condition

eLON\_AL\_FIR\_MAINT\_ALERT: Fire-related maintenance alert

eLON\_AL\_FATAL\_ERROR: Fatal application error

eLON\_AL\_ERROR: Other error condition

eLON\_AL\_WARNING: Other warning condition

### 7.3.4 E\_LON\_appl\_cwc\_t

Genutzt durch: SNVT\_clothes\_w\_c / SNVT\_clothes\_w\_s

```
TYPE E_LON_appl_cwc_t :
(
  eLON_CWC_NUL      := -1,
  eLON_CWC_WASH     := 0,
  eLON_CWC_RINSE    := 1,
  eLON_CWC_SPIN     := 2,
  eLON_CWC_DRY      := 3
)
END_TYPE
```

eLON\_CWC\_NUL: Invalid Value

eLON\_CWC\_WASH: Wash

eLON\_CWC\_RINSE: Rinse

eLON\_CWC\_SPIN: Spin

eLON\_CWC\_DRY: Dry

### 7.3.5 E\_LON\_appl\_cwp\_t

Genutzt durch: SNVT\_clothes\_w\_c

```
TYPE E_LON_appl_cwp_t :
(
  eLON_CWP_NUL      := -1,
  eLON_CWP_GENERAL  := 0,
  eLON_CWP_BOIL     := 1,
  eLON_CWP_FAST_WASH := 2,
  eLON_CWP_LINGERIE := 3,
  eLON_CWP_WOOL     := 4,
  eLON_CWP_TOWEL    := 5,
  eLON_CWP_BED_LINENS := 6,
  eLON_CWP_CURTAIN  := 7,
  eLON_CWP_RINSE_SPIN_ONLY := 8,
  eLON_CWP_DELICATE_RINSE := 9,
  eLON_CWP_SPIN_ONLY := 10,
  eLON_CWP_DRY_ONLY := 11
)
END_TYPE
```

eLON\_CWP\_NUL: Invalid Value

eLON\_CWP\_GENERAL: Normal Wash

eLON\_CWP\_BOIL: Boil

eLON\_CWP\_FAST\_WASH: Fast Wash  
 eLON\_CWP\_LINGERIE: Lingerie  
 eLON\_CWP\_WOOL: Wool  
 eLON\_CWP\_TOWEL: Towel  
 eLON\_CWP\_BED\_LINENS: Bed Linens  
 eLON\_CWP\_CURTAIN: Curtain  
 eLON\_CWP\_RINSE\_SPIN\_ONLY: Rinse and Spin Only  
 eLON\_CWP\_DELICATE\_RINSE: Delicate Rinse  
 eLON\_CWP\_SPIN\_ONLY: Spin Only  
 eLON\_CWP\_DRY\_ONLY: Dry Only

### 7.3.6 E\_LON\_appl\_cws\_t

Genutzt durch: SNVT\_clothes\_w\_c / SNVT\_clothes\_w\_s

```

TYPE E_LON_appl_cws_t :
(
    eLON_CWS_NUL           := -1,
    eLON_CWS_LOAD_SENSING := 0,
    eLON_CWS_WETTING      := 1,
    eLON_CWS_DETERGENT    := 2,
    eLON_CWS_WASHING      := 3,
    eLON_CWS_WATERING     := 4,
    eLON_CWS_RINSING      := 5,
    eLON_CWS_ARRANGING    := 6,
    eLON_CWS_DRAIN        := 7,
    eLON_CWS_SPINNING     := 8,
    eLON_CWS_FINAL_SPINNING := 9,
    eLON_CWS_FLUFFING     := 10,
    eLON_CWS_DRYING       := 11,
    eLON_CWS_COOLING      := 12
)
END_TYPE
    
```

eLON\_CWS\_NUL: Invalid Value  
 eLON\_CWS\_LOAD\_SENSING: Sensing Load  
 eLON\_CWS\_WETTING: Wetting  
 eLON\_CWS\_DETERGENT: Detergent  
 eLON\_CWS\_WASHING: Washing  
 eLON\_CWS\_WATERING: Watering  
 eLON\_CWS\_RINSING: Rinsing  
 eLON\_CWS\_ARRANGING: Arranging  
 eLON\_CWS\_DRAIN: Drain  
 eLON\_CWS\_SPINNING: Spinning  
 eLON\_CWS\_FINAL\_SPINNING: In Final Spin  
 eLON\_CWS\_FLUFFING: Fluffing  
 eLON\_CWS\_DRYING: Drying  
 eLON\_CWS\_COOLING: Cooling

### 7.3.7 E\_LON\_appl\_rin\_t

Genutzt durch: SNVT\_clothes\_w\_c

```
TYPE E_LON_appl_rin_t :
(
  eLON_RIN_NUL      := -1,
  eLON_RIN_PRE_WASH := 0,
  eLON_RIN_WATER_PLUS := 1,
  eLON_RIN_DETERGENT_PLUS := 2,
  eLON_RIN_RINSE_HOLD := 3
)
END_TYPE
```

eLON\_RIN\_NUL: Invalid Value

eLON\_RIN\_PRE\_WASH: Pre-wash

eLON\_RIN\_WATER\_PLUS: Water Plus

eLON\_RIN\_DETERGENT\_PLUS: Detergent Plus

eLON\_RIN\_RINSE\_HOLD: Rinse Hold

### 7.3.8 E\_LON\_boolean\_t

Genutzt durch: SCPTautoAnswer / SCPTcoolingResetEnable / SCPTdefrostHold / SCPTdefrostInternalSchedule / SCPTheatingResetEnable / SCPTHighLimit1Enable / SCPTHighLimit2Enable / SCPTlowLimit1Enable / SCPTlowLimit2Enable / SCPTscheduleInternal / SNVT\_clothes\_w\_c / SNVT\_pump\_sensor / SNVT\_pumpset\_mn / SNVT\_pumpset\_sn

```
TYPE E_LON_boolean_t :
(
  eLON_BOOL_NUL      := -1,
  eLON_BOOL_FALSE := 0,
  eLON_BOOL_TRUE := 1
)
END_TYPE
```

eLON\_BOOL\_NUL: Invalid Value

eLON\_BOOL\_FALSE: False

eLON\_BOOL\_TRUE: True

### 7.3.9 E\_LON\_calendar\_type\_t

Genutzt durch: SNVT\_time\_zone

```
TYPE E_LON_calendar_type_t :
(
  eLON_CAL_NUL      := -1,
  eLON_CAL_GREG := 0,
  eLON_CAL_JUL := 1,
  eLON_CAL_MEU := 2
)
END_TYPE
```

eLON\_CAL\_NUL: Invalid Value

eLON\_CAL\_GREG: Gregorian calendar

eLON\_CAL\_JUL: Julian calendar

eLON\_CAL\_MEU: Calendar Method European/US "MEU"

### 7.3.10 E\_LON\_cam\_act\_t

Genutzt durch: SNVT\_pos\_ctrl

```

TYPE E_LON_cam_act_t :
(
  eLON_CMA_NUL    := -1,
  eLON_CMA_SAVE  := 0,
  eLON_CMA_CALL  := 1,
  eLON_CMA_READ  := 2
)
END_TYPE

```

eLON\_CMA\_NUL: Invalid action call response

eLON\_CMA\_SAVE: Save the values defined by the function

eLON\_CMA\_CALL: Preposition tour tables

eLON\_CMA\_READ: Absolute positions

### 7.3.11 E\_LON\_cam\_func\_t

Genutzt durch: SNVT\_pos\_ctrl

```

TYPE E_LON_cam_func_t :
(
  eLON_CMF_NUL    := -1,
  eLON_CMF_REL    := 0,
  eLON_CMF_TOUR   := 1,
  eLON_CMF_ABS    := 2
)
END_TYPE

```

eLON\_CMF\_NUL: Invalid function call response

eLON\_CMF\_REL: Relative positions, prepositions

eLON\_CMF\_TOUR: Preposition tour tables

eLON\_CMF\_ABS: Absolute positions

### 7.3.12 E\_LON\_chiller\_t

Genutzt durch: SNVT\_chlr\_status

```

TYPE E_LON_chiller_t :
(
  eLON_CHLR_NUL    := -1,
  eLON_CHLR_OFF    := 0,
  eLON_CHLR_START  := 1,
  eLON_CHLR_RUN    := 2,
  eLON_CHLR_PRESHUTDN := 3,
  eLON_CHLR_SERVICE := 4
)
END_TYPE

```

eLON\_CHLR\_NUL: Invalid Value

eLON\_CHLR\_OFF: Chiller off

eLON\_CHLR\_START: Chiller in start mode

eLON\_CHLR\_RUN: Chiller in run mode

eLON\_CHLR\_PRESHUTDN: Chiller in pre shutdown mode

eLON\_CHLR\_SERVICE: Chiller in service mode

### 7.3.13 E\_LON\_color\_encoding\_t

Genutzt durch: SNVT\_color\_2

```

TYPE E_LON_color_encoding_t :
(
  eLON_COLOR_NUL    := -1,

```

```
eLON_COLOR_CIE31_LUMEN := 0,
eLON_COLOR_CIE31_PERCENT := 1,
eLON_COLOR_RGB := 2,
eLON_COLOR_TEMPERATURE := 3
)
END_TYPE
```

eLON\_COLOR\_NUL: Invalid value

eLON\_COLOR\_CIE31\_LUMEN: CIE 1931 color space; Y output in lumen

eLON\_COLOR\_CIE31\_PERCENT: CIE 1931 color space; Y output in percent of maximum lumen output of the lamp

eLON\_COLOR\_RGB: No color space, RGB color value

eLON\_COLOR\_TEMPERATURE: Color temperature

### 7.3.14 E\_LON\_config\_source\_t

Genutzt durch: SNVT\_config\_src

```
TYPE E_LON_config_source_t :
(
  eLON_CFG_NUL := -1,
  eLON_CFG_LOCAL := 0,
  eLON_CFG_EXTERNAL := 1
)
END_TYPE
```

eLON\_CFG\_NUL: Invalid Value

eLON\_CFG\_LOCAL: Device will use self-installation functions to set its own network image

eLON\_CFG\_EXTERNAL: Device's network image will be set by an outside source

### 7.3.15 E\_LON\_control\_resp\_t

Genutzt durch: SNVT\_ctrl\_resp

```
TYPE E_LON_control_resp_t :
(
  eLON_CTRLR_NUL := -1,
  eLON_CTRLR_NO := 0,
  eLON_CTRLR_PEND := 1,
  eLON_CTRLR_REL := 2,
  eLON_CTRLR_QUERY := 3,
  eLON_CTRLR_RES := 4,
  eLON_CTRLR_ERR := 5
)
END_TYPE
```

eLON\_CTRLR\_NUL: Invalid value

eLON\_CTRLR\_NO: Number of current controller

eLON\_CTRLR\_PEND: Request pending due to control query to current operator

eLON\_CTRLR\_REL: Current control released

eLON\_CTRLR\_QUERY: Query to current controller

eLON\_CTRLR\_RES: Controllable device has been reset

eLON\_CTRLR\_ERR: Error in control

### 7.3.16 E\_LON\_currency\_t

Genutzt durch: SNVT\_currency

```

TYPE E_LON_currency_t :
(
  eLON_CU_NUL                := -1,
  eLON_CU_ARGENTINA_PESO     := 0,
  eLON_CU_AUSTRALIA_DOLLAR   := 1,
  eLON_CU_AUSTRIA_SCHILLING  := 2,
  eLON_CU_BAHRAIN_DINAR     := 3,
  eLON_CU_BELGIUM_FRANC     := 4,
  eLON_CU_BRAZIL_CRUZEIRO_REAL := 5,
  eLON_CU_BRITAIN_POUND     := 6,
  eLON_CU_CANADA_DOLLAR     := 7,
  eLON_CU_CZECH_KORUNA      := 8,
  eLON_CU_CHILE_PESO        := 9,
  eLON_CU_CHINA_RENMINBI    := 10,
  eLON_CU_COLOMBIA_PESO     := 11,
  eLON_CU_DENMARK_KRONE     := 12,
  eLON_CU_ECUADOR_SUCRE    := 13,
  eLON_CU_EUROPEAN_CURRENCY_UNIT := 14,
  eLON_CU_FINLAND_MARKKA    := 15,
  eLON_CU_FRANCE_FRANC      := 16,
  eLON_CU_GERMANY_MARK      := 17,
  eLON_CU_GREECE_DRACHMA    := 18,
  eLON_CU_HONG_KONG_DOLLAR  := 19,
  eLON_CU_HUNGARY_FORINT    := 20,
  eLON_CU_INDIA_RUPEE       := 21,
  eLON_CU_INDONESIA_RUPIAH   := 22,
  eLON_CU_IRELAND_PUNT      := 23,
  eLON_CU_ISRAEL_SHEKEL     := 24,
  eLON_CU_ITALY_LIRA        := 25,
  eLON_CU_JAPAN_YEN         := 26,
  eLON_CU_JORDAN_DINAR      := 27,
  eLON_CU_KUWAIT_DINAR     := 28,
  eLON_CU_LEBANON_POUND     := 29,
  eLON_CU_MALAYSIA_RINGGIT  := 30,
  eLON_CU_MALTA_LIRA        := 31,
  eLON_CU_MEXICO_PESO       := 32,
  eLON_CU_NETHERLANDS_GUILDER := 33,
  eLON_CU_NEW_ZEALAND_DOLLAR := 34,
  eLON_CU_NORWAY_KRONE      := 35,
  eLON_CU_PAKISTAN_RUPEE    := 36,
  eLON_CU_PERU_NEW_SOL      := 37,
  eLON_CU_PHILIPPINES_PESO  := 38,
  eLON_CU_POLAND_ZLOTY      := 39,
  eLON_CU_PORTUGAL_ESCUDO   := 40,
  eLON_CU_SAUDI_ARABIA_RIYAL := 41,
  eLON_CU_SINGAPORE_DOLLAR  := 42,
  eLON_CU_SLOVAK_KORUNA     := 43,
  eLON_CU_SOUTH_AFRICA_RAND := 44,
  eLON_CU_SOUTH_KOREA_WON   := 45,
  eLON_CU_SPAIN_PESETA      := 46,
  eLON_CU_SPECIAL_DRAWING_RIGHTS := 47,
  eLON_CU_SWEDEN_KRONA     := 48,
  eLON_CU_SWITZERLAND_FRANC := 49,
  eLON_CU_TAIWAN_DOLLAR     := 50,
  eLON_CU_THAILAND_BAHT     := 51,
  eLON_CU_TURKEY_LIRA       := 52,
  eLON_CU_UNITED_ARAB_DIRHAM := 53,
  eLON_CU_UNITED_STATES_DOLLAR := 54,
  eLON_CU_URUGUAY_NEW_PESO  := 55,
  eLON_CU_VENEZUELA_BOLIVAR := 56
)
END_TYPE

```

eLON\_CU\_NUL: Invalid Value

eLON\_CU\_ARGENTINA\_PESO: Argentine Peso

eLON\_CU\_AUSTRALIA\_DOLLAR: Australian Dollar

eLON\_CU\_AUSTRIA\_SCHILLING: Austrian Schilling

eLON\_CU\_BAHRAIN\_DINAR: Bahraini Dinar

eLON\_CU\_BELGIUM\_FRANC: Belgian Franc

eLON\_CU\_BRAZIL\_CRUZEIRO\_REAL: Brazilian Cruzeiro Real

eLON\_CU\_BRITAIN\_POUND: British Pound

eLON\_CU\_CANADA\_DOLLAR: Canadian Dollar  
eLON\_CU\_CZECH\_KORUNA: Czechoslovakian Koruna  
eLON\_CU\_CHILE\_PESO: Chilean Peso  
eLON\_CU\_CHINA\_RENMINBI: Chinese Renminbi Yuan  
eLON\_CU\_COLOMBIA\_PESO: Colombian Peso  
eLON\_CU\_DENMARK\_KRONE: Danish Krone  
eLON\_CU\_ECUADOR\_SUCRE: Ecuadorian Sucre  
eLON\_CU\_EUROPEAN\_CURRENCY\_UNIT: European Euro  
eLON\_CU\_FINLAND\_MARKKA: Finnish Markka  
eLON\_CU\_FRANCE\_FRANC: French Franc  
eLON\_CU\_GERMANY\_MARK: German Mark  
eLON\_CU\_GREECE\_DRACHMA: Greek Drachma  
eLON\_CU\_HONG\_KONG\_DOLLAR: Hong Kong Dollar  
eLON\_CU\_HUNGARY\_FORINT: Hungarian Forint  
eLON\_CU\_INDIA\_RUPEE: Indian Rupee  
eLON\_CU\_INDONESIA\_RUPIAH: Indonesian Rupiah  
eLON\_CU\_IRELAND\_PUNT: Irish Punt  
eLON\_CU\_ISRAEL\_SHEKEL: Israeli Shekel  
eLON\_CU\_ITALY\_LIRA: Italian Lira  
eLON\_CU\_JAPAN\_YEN: Japanese Yen  
eLON\_CU\_JORDAN\_DINAR: Jordanian Dinar  
eLON\_CU\_KUWAIT\_DINAR: Kuwaiti Dinar  
eLON\_CU\_LEBANON\_POUND: Lebanese Pound  
eLON\_CU\_MALAYSIA\_RINGGIT: Malaysian Ringgit  
eLON\_CU\_MALTA\_LIRA: Maltese Lira  
eLON\_CU\_MEXICO\_PESO: Mexican New Peso  
eLON\_CU\_NETHERLANDS\_GUILDER: Netherlands Guilder  
eLON\_CU\_NEW\_ZEALAND\_DOLLAR: New Zealand Dollar  
eLON\_CU\_NORWAY\_KRONE: Norwegian Krone  
eLON\_CU\_PAKISTAN\_RUPEE: Pakistani Rupee  
eLON\_CU\_PERU\_NEW\_SOL: Peruvian New Sol  
eLON\_CU\_PHILIPPINES\_PESO: Philippine Peso  
eLON\_CU\_POLAND\_ZLOTY: Polish Zloty  
eLON\_CU\_PORTUGAL\_ESCUDO: Portuguese Escudo  
eLON\_CU\_SAUDI\_ARABIA\_RIYAL: Saudi Arabian Riyal  
eLON\_CU\_SINGAPORE\_DOLLAR: Singaporean Dollar



eLON\_CU\_SLOVAK\_KORUNA: Slavic Koruna  
 eLON\_CU\_SOUTH\_AFRICA\_RAND: South African Rand  
 eLON\_CU\_SOUTH\_KOREA\_WON: South Korean Won  
 eLON\_CU\_SPAIN\_PESETA: Spanish Peseta  
 eLON\_CU\_SPECIAL\_DRAWING\_RIGHTS: international governmental exchange  
 eLON\_CU\_SWEDEN\_KRONA: Swedish Krona  
 eLON\_CU\_SWITZERLAND\_FRANC: Swiss Franc  
 eLON\_CU\_TAIWAN\_DOLLAR: Taiwanese Dollar  
 eLON\_CU\_THAILAND\_BAHT: Thai Baht  
 eLON\_CU\_TURKEY\_LIRA: Turkish Lira  
 eLON\_CU\_UNITED\_ARAB\_DIRHAM: United Arab Emirates Dirham  
 eLON\_CU\_UNITED\_STATES\_DOLLAR: United States Dollar  
 eLON\_CU\_URUGUAY\_NEW\_PESO: Uruguayan New Peso  
 eLON\_CU\_VENEZUELA\_BOLIVAR: Venezuelan Bolivar

### 7.3.17 E\_LON\_days\_of\_week\_t

Genutzt durch: SCPTtimePeriod / SNVT\_date\_day / SNVT\_time\_zone

```
TYPE E_LON_days_of_week_t :
(
    eLON_DAY_NUL    := -1,
    eLON_DAY_SUN    := 0,
    eLON_DAY_MON    := 1,
    eLON_DAY_TUE    := 2,
    eLON_DAY_WED    := 3,
    eLON_DAY_THU    := 4,
    eLON_DAY_FRI    := 5,
    eLON_DAY_SAT    := 6
)
END_TYPE
```

eLON\_DAY\_NUL: Invalid Value  
 eLON\_DAY\_SUN: Sunday  
 eLON\_DAY\_MON: Monday  
 eLON\_DAY\_TUE: Tuesday  
 eLON\_DAY\_WED: Wednesday  
 eLON\_DAY\_THU: Thursday  
 eLON\_DAY\_FRI: Friday  
 eLON\_DAY\_SAT: Saturday

### 7.3.18 E\_LON\_defrost\_mode\_t

Genutzt durch: SNVT\_defr\_mode

```
TYPE E_LON_defrost_mode_t :
(
    eLON_DFM_NUL          := -1,
    eLON_DFM_MODE_AMBIENT := 0,
    eLON_DFM_MODE_FORCED  := 1,

```

```
eLON_DFM_MODE_SYNC      := 2
)
END_TYPE
```

eLON\_DFM\_NUL: Invalid Value

eLON\_DFM\_MODE\_AMBIENT: No forced heating required

eLON\_DFM\_MODE\_FORCED: Start-up after defrost ignored

eLON\_DFM\_MODE\_SYNC: Synchronized

### 7.3.19 E\_LON\_defrost\_state\_t

Genutzt durch: SNVT\_defr\_state

```
TYPE E_LON_defrost_state_t :
(
  eLON_DFS_NUL      := -1,
  eLON_DFS_STANDBY  := 0,
  eLON_DFS_PUMPDOWN := 1,
  eLON_DFS_DEFROST  := 2,
  eLON_DFS_DRAINDOWN := 3,
  eLON_DFS_INJECT_DLY := 4
)
END_TYPE
```

eLON\_DFS\_NUL: Invalid Value

eLON\_DFS\_STANDBY: Defrost in standby

eLON\_DFS\_PUMPDOWN: Defrost in pump-down mode

eLON\_DFS\_DEFROST: In defrost mode

eLON\_DFS\_DRAINDOWN: Defrost in drain-down

eLON\_DFS\_INJECT\_DLY: Defrost in injection delay

### 7.3.20 E\_LON\_defrost\_term\_t

Genutzt durch: SNVT\_defr\_term

```
TYPE E_LON_defrost_term_t :
(
  eLON_DFT_NUL      := -1,
  eLON_DFT_TERM_TEMP := 0,
  eLON_DFT_TERM_TIME := 1,
  eLON_DFT_TERM_FIRST := 2,
  eLON_DFT_TERM_LAST := 3,
  eLON_DFT_TERM_SENSOR := 4,
  eLON_DFT_TERM_DISCHARGE := 5,
  eLON_DFT_TERM_RETURN := 6,
  eLON_DFT_TERM_SW_OPEN := 7,
  eLON_DFT_TERM_SW_CLOSE := 8,
  eLON_DFT_TERM_MANUF := 100
)
END_TYPE
```

eLON\_DFT\_NUL: Invalid Value

eLON\_DFT\_TERM\_TEMP: Terminate on temperature

eLON\_DFT\_TERM\_TIME: Terminate on time

eLON\_DFT\_TERM\_FIRST: Terminate on first occurring

eLON\_DFT\_TERM\_LAST: Terminate on last occurring

eLON\_DFT\_TERM\_SENSOR: Terminate on sensor

eLON\_DFT\_TERM\_DISCHARGE: Terminate on discharge

eLON\_DFT\_TERM\_RETURN: Terminate on return

eLON\_DFT\_TERM\_SW\_OPEN: Terminate on "Switch Open"

eLON\_DFT\_TERM\_SW\_CLOSE: Terminate on "Switch Closed"

eLON\_DFT\_TERM\_MANUF: Manufacturer-Defined termination state

### 7.3.21 E\_LON\_device\_c\_mode\_t

Genutzt durch: SNVT\_dev\_c\_mode

```

TYPE E_LON_device_c_mode_t :
(
  eLON_DCM_NUL           := -1,
  eLON_DCM_SPEED_CONST  := 0,
  eLON_DCM_PRESS_CONST  := 1,
  eLON_DCM_PRESS_COMP   := 2,
  eLON_DCM_FLOW_CONST   := 3,
  eLON_DCM_FLOW_COMP    := 4,
  eLON_DCM_TEMP_CONST   := 5,
  eLON_DCM_TEMP_COMP    := 6,
  eLON_DCM_PRESS_AUTO   := 7,
  eLON_DCM_QUICK_OPEN   := 20,
  eLON_DCM_LINEAR       := 21,
  eLON_DCM_EQUAL_PERCENT := 22,
  eLON_DCM_QUADRATIC    := 23,
  eLON_DCM_FREE_DEFINED := 24,
  eLON_DCM_2WAY_VALVE   := 27,
  eLON_DCM_MIXING_VALVE := 28,
  eLON_DCM_DIVERTING_VALVE := 29,
  eLON_DCM_INVFNC_QCK_OPN := 30,
  eLON_DCM_INVFNC_EQL_PERC := 31,
  eLON_DCM_INVFNC_QUAD  := 32
)
END_TYPE
    
```

eLON\_DCM\_NUL: Invalid Value

eLON\_DCM\_SPEED\_CONST:

eLON\_DCM\_PRESS\_CONST:

eLON\_DCM\_PRESS\_COMP:

eLON\_DCM\_FLOW\_CONST:

eLON\_DCM\_FLOW\_COMP:

eLON\_DCM\_TEMP\_CONST:

eLON\_DCM\_TEMP\_COMP:

eLON\_DCM\_PRESS\_AUTO:

eLON\_DCM\_QUICK\_OPEN: Valve works with Quick-Open flow characteristic

eLON\_DCM\_LINEAR: Valve works with Linear flow characteristic

eLON\_DCM\_EQUAL\_PERCENT: Valve works with Equal Percent flow characteristic

eLON\_DCM\_QUADRATIC: Valve works with Quadratic flow characteristic

eLON\_DCM\_FREE\_DEFINED: Valve works with free defined flow characteristic

eLON\_DCM\_2WAY\_VALVE:

eLON\_DCM\_MIXING\_VALVE:

eLON\_DCM\_DIVERTING\_VALVE:

eLON\_DCM\_INVFNC\_QCK\_OPN:

eLON\_DCM\_INVFNC\_EQL\_PERC:

eLON\_DCM\_INVFNC\_QUAD:

### 7.3.22 E\_LON\_device\_select\_t

Genutzt durch: SNVT\_dev\_fault / SNVT\_dev\_maint / SNVT\_dev\_status

```
TYPE E_LON_device_select_t :
(
  eLON_DV_NUL      := -1,
  eLON_DV_PUMP_CTRL := 0,
  eLON_DV_VALVE_POS := 1
)
END_TYPE
```

eLON\_DV\_NUL: Invalid value

eLON\_DV\_PUMP\_CTRL: Use union for SFPTpumpController values

eLON\_DV\_VALVE\_POS: Use union for SFPTvalvePositioner values

### 7.3.23 E\_LON\_discrete\_levels\_t

Genutzt durch: SNVT\_clothes\_w\_c / SNVT\_lev\_disc

```
TYPE E_LON_discrete_levels_t :
(
  eLON_ST_NUL      := -1,
  eLON_ST_OFF      := 0,
  eLON_ST_LOW      := 1,
  eLON_ST_MED      := 2,
  eLON_ST_HIGH     := 3,
  eLON_ST_ON       := 4
)
END_TYPE
```

eLON\_ST\_NUL:

eLON\_ST\_OFF:

eLON\_ST\_LOW:

eLON\_ST\_MED:

eLON\_ST\_HIGH:

eLON\_ST\_ON:

### 7.3.24 E\_LON\_emerg\_t

Genutzt durch: SNVT\_hvac\_emerg

```
TYPE E_LON_emerg_t :
(
  eLON_EMERG_NUL      := -1,
  eLON_EMERG_NORMAL   := 0,
  eLON_EMERG_PRESSURIZE := 1,
  eLON_EMERG_DEPRESSURIZE := 2,
  eLON_EMERG_PURGE    := 3,
  eLON_EMERG_SHUTDOWN := 4,
  eLON_EMERG_FIRE     := 5
)
END_TYPE
```

eLON\_EMERG\_NUL: Invalid Value

eLON\_EMERG\_NORMAL: No emergency mode

eLON\_EMERG\_PRESSURIZE: Emergency pressurize mode

eLON\_EMERG\_DEPRESSURIZE: Emergency depressurize mode

eLON\_EMERG\_PURGE: Emergency purge mode

eLON\_EMERG\_SHUTDOWN: Emergency shutdown mode

eLON\_EMERG\_FIRE: Emergency fire mode

### 7.3.25 E\_LON\_ent\_cmd\_t

Genutzt durch: SNVT\_ent\_state

```

TYPE E_LON_ent_cmd_t :
(
  eLON_ES_NUL           := -1,
  eLON_ES_UNDEFINED    := 0,
  eLON_ES_OPEN_PULS    := 1,
  eLON_ES_OPEN         := 2,
  eLON_ES_CLOSE        := 3,
  eLON_ES_STOP         := 4,
  eLON_ES_STOP_RESUME  := 5,
  eLON_ES_ENTRY_REQ    := 6,
  eLON_ES_EXIT_REQ     := 7,
  eLON_ES_KEY_REQ      := 8,
  eLON_ES_SAFETY_EXT_REQ := 9,
  eLON_ES_EMERGENCY_REQ := 10,
  eLON_ES_UPDATE_STATE := 11,
  eLON_ES_SAF_EXT_RESUME := 12,
  eLON_ES_EMERG_RESUME := 13
)
END_TYPE
    
```

eLON\_ES\_NUL: Invalid Value

eLON\_ES\_UNDEFINED: State is not yet defined

eLON\_ES\_OPEN\_PULS: Open the device and close it when back in normal position

eLON\_ES\_OPEN: Open the device if not locked

eLON\_ES\_CLOSE: Close the device

eLON\_ES\_STOP: Stop the device

eLON\_ES\_STOP\_RESUME: Continue after stop command

eLON\_ES\_ENTRY\_REQ: Entry request, access in to the area

eLON\_ES\_EXIT\_REQ: Exit request, access out from the area

eLON\_ES\_KEY\_REQ: Exit request, access out from the area

eLON\_ES\_SAFETY\_EXT\_REQ: Safety request, the device will go to a pre defined safety position/mode

eLON\_ES\_EMERGENCY\_REQ: Emergency request, the device will go to an pre defined emergency position/mode

eLON\_ES\_UPDATE\_STATE: Update the current state and mode

eLON\_ES\_SAF\_EXT\_RESUME: Resume after Safety function

eLON\_ES\_EMERG\_RESUME: Resume after Emergency function

### 7.3.26 E\_LON\_ent\_opmode\_cmd\_t

Genutzt durch: SNVT\_ent\_opmode / SNVT\_ent\_status

```

TYPE E_LON_ent_opmode_cmd_t :
(
  eLON_EM_NUL           := -1,
  eLON_EM_UNDEFINED    := 0,
  eLON_EM_AUTO         := 1,
  eLON_EM_AUTO_RED     := 2,
  eLON_EM_CLOSE_LOCK   := 3,
  eLON_EM_CLOSE_UNLOCK := 4,
)
    
```

```

eLON_EM_EXIT_ONLY      := 5,
eLON_EM_OPEN          := 6,
eLON_EM_OPEN_ONCE     := 7,
eLON_EM_MANUAL        := 8,
eLON_EM_FIRE          := 9,
eLON_EM_EVAC          := 10,
eLON_EM_WEATHER       := 11,
eLON_EM_DAY_LOCKING   := 12,
eLON_EM_NIGHT_LOCKING := 13,
eLON_EM_BLOCKED       := 14,
eLON_EM_SERVICE       := 15,
eLON_EM_ENTRY_ONLY    := 16
)
END_TYPE

```

eLON\_EM\_NUL: Invalid Value

eLON\_EM\_UNDEFINED: Operation mode is not defined

eLON\_EM\_AUTO: Operation mode is AUTOMATIC

eLON\_EM\_AUTO\_RED: Operation mode is AUTOMATIC with reduced width

eLON\_EM\_CLOSE\_LOCK: Operation mode is CLOSE AND LOCK

eLON\_EM\_CLOSE\_UNLOCK: Operation mode is CLOSE AND UNLOCK

eLON\_EM\_EXIT\_ONLY: Operation mode is EXIT ONLY

eLON\_EM\_OPEN: Operation mode is OPEN

eLON\_EM\_OPEN\_ONCE: Operation mode is OPEN AND CLOSE ONCE

eLON\_EM\_MANUAL: Operation mode is MANUAL

eLON\_EM\_FIRE: Operation mode is FIRE

eLON\_EM\_EVAC: Operation mode is EVACUATION

eLON\_EM\_WEATHER: Operation mode is WEATHER MODE

eLON\_EM\_DAY\_LOCKING: Operation mode is DAY\_LOCKING, locking with reduced level of security

eLON\_EM\_NIGHT\_LOCKING: Operation mode is NIGHT\_LOCKING, locking with maximum level of security

eLON\_EM\_BLOCKED: Operation mode is BLOCKED, no operations is allowed

eLON\_EM\_SERVICE: Operation mode is SERVICE

eLON\_EM\_ENTRY\_ONLY: Operation mode is ENTRY\_ONLY

### 7.3.27 E\_LON\_evap\_t

Genutzt durch: SNVT\_evap\_state

```

TYPE E_LON_evap_t :
(
  eLON_EVAP_NUL      := -1,
  eLON_EVAP_NO_COOLING := 0,
  eLON_EVAP_COOLING  := 1,
  eLON_EVAP_EMERG_COOLING := 2
)
END_TYPE

```

eLON\_EVAP\_NUL: Invalid Value

eLON\_EVAP\_NO\_COOLING: Object not performing cooling (off cycle or disabled)

eLON\_EVAP\_COOLING: Object currently cooling

eLON\_EVAP\_EMERG\_COOLING: Object performing emergency cooling

### 7.3.28 E\_LON\_ex\_control\_t

Genutzt durch: SNVT\_ex\_control

```
TYPE E_LON_ex_control_t :
(
  eLON_EX_CONTROL_NUL      := -1,
  eLON_EX_CONTROL_NONE    := 0,
  eLON_EX_CONTROL_OTHER   := 1,
  eLON_EX_CONTROL_THIS_ADDR := 2
)
END_TYPE
```

eLON\_EX\_CONTROL\_NUL: The control status of the item is unknown

eLON\_EX\_CONTROL\_NONE: Nothing has control of the item.

eLON\_EX\_CONTROL\_OTHER: Some unidentified entity has control of the item.

eLON\_EX\_CONTROL\_THIS\_ADDR: A device has control of the item. The network address of this device is specified in the control\_device\_addr

### 7.3.29 E\_LON\_file\_request\_t

Genutzt durch: SNVT\_file\_req

```
TYPE E_LON_file_request_t :
(
  eLON_FR_NUL              := -1,
  eLON_FR_OPEN_TO_SEND    := 0,
  eLON_FR_OPEN_TO_RECEIVE := 1,
  eLON_FR_CLOSE_FILE      := 2,
  eLON_FR_CLOSE_DELETE_FILE := 3,
  eLON_FR_DIRECTORY_LOOKUP := 4,
  eLON_FR_OPEN_TO_SEND_RA := 5,
  eLON_FR_OPEN_TO_RECEIVE_RA := 6
)
END_TYPE
```

eLON\_FR\_NUL: Invalid Value

eLON\_FR\_OPEN\_TO\_SEND: Sequential access read

eLON\_FR\_OPEN\_TO\_RECEIVE: Sequential access write

eLON\_FR\_CLOSE\_FILE: Close and save file

eLON\_FR\_CLOSE\_DELETE\_FILE: Close and delete file

eLON\_FR\_DIRECTORY\_LOOKUP: Retrieve directory entry

eLON\_FR\_OPEN\_TO\_SEND\_RA: Random access read

eLON\_FR\_OPEN\_TO\_RECEIVE\_RA: Random access write

### 7.3.30 E\_LON\_file\_status\_t

Genutzt durch: SNVT\_file\_status

```
TYPE E_LON_file_status_t :
(
  eLON_FS_NUL              := -1,
  eLON_FS_XFER_OK         := 0,
  eLON_FS_LOOKUP_OK       := 1,
  eLON_FS_OPEN_FAIL       := 2,
  eLON_FS_LOOKUP_ERR      := 3,
  eLON_FS_XFER_UNDERWAY   := 4,
  eLON_FS_IO_ERR          := 5,
  eLON_FS_TIMEOUT_ERR     := 6,
  eLON_FS_WINDOW_ERR      := 7,
  eLON_FS_AUTH_ERR        := 8,
  eLON_FS_ACCESS_UNAVAIL  := 9,
  eLON_FS_SEEK_INVALID    := 10,

```

```
eLON_FS_SEEK_WAIT      := 11
)
END_TYPE
```

eLON\_FS\_NUL: Invalid Value

eLON\_FS\_XFER\_OK: File transfer successful

eLON\_FS\_LOOKUP\_OK: Directory lookup successful

eLON\_FS\_OPEN\_FAIL: Error on opening file

eLON\_FS\_LOOKUP\_ERR: Error on directory lookup

eLON\_FS\_XFER\_UNDERWAY: File transfer in progress

eLON\_FS\_IO\_ERR: Error on reading/writing file

eLON\_FS\_TIMEOUT\_ERR: File transfer timed out

eLON\_FS\_WINDOW\_ERR: Window sequence error

eLON\_FS\_AUTH\_ERR: Authentication failure

eLON\_FS\_ACCESS\_UNAVAIL: Access mode not supported

eLON\_FS\_SEEK\_INVALID: Random access beyond EOF

eLON\_FS\_SEEK\_WAIT:

### 7.3.31 E\_LON\_fire\_indicator\_t

Genutzt durch: SNVT\_fire\_indcte

```
TYPE E_LON_fire_indicator_t :
(
  eLON_FN_NUL      := -1,
  eLON_FN_UNDEFINED := 0,
  eLON_FN_STROBE_U := 1,
  eLON_FN_STROBE_S := 2,
  eLON_FN_HORN     := 3,
  eLON_FN_CHIME    := 4,
  eLON_FN_BELL     := 5,
  eLON_FN_SOUNDER  := 6,
  eLON_FN_SPEAKER  := 7,
  eLON_FN_UNIVERSAL := 8
)
END_TYPE
```

eLON\_FN\_NUL: Invalid Value

eLON\_FN\_UNDEFINED: Undefined indicator

eLON\_FN\_STROBE\_U: The indicator is un-synchronized

eLON\_FN\_STROBE\_S: The indicator is synchronized

eLON\_FN\_HORN: The indicator is a DC input, pre coded Horn

eLON\_FN\_CHIME: The indicator is a DC input, pre coded Chime

eLON\_FN\_BELL: The indicator is a DC input

eLON\_FN\_SOUNDER: The indicator is powered from the device

eLON\_FN\_SPEAKER: The indicator is an AC input for the speaker

eLON\_FN\_UNIVERSAL: General purpose indicator



### 7.3.32 E\_LON\_fire\_initiator\_t

Genutzt durch: SNVT\_fire\_init

```

TYPE E_LON_fire_initiator_t :
(
  eLON_FI_NUL           := -1,
  eLON_FI_UNDEFINED     := 0,
  eLON_FI_THERMAL_FIXED := 1,
  eLON_FI_SMOKE_ION     := 2,
  eLON_FI_MULTI_ION_THERMAL := 3,
  eLON_FI_SMOKE_PHOTO  := 4,
  eLON_FI_MULTI_PHOTO_THERMAL := 5,
  eLON_FI_MULTI_PHOTO_ION := 6,
  eLON_FI_MULTI_PHOTO_ION_THERMAL := 7,
  eLON_FI_THERMAL_ROR  := 8,
  eLON_FI_MULTI_THERMAL_ROR := 9,
  eLON_FI_MANUAL_PULL  := 10,
  eLON_FI_WATER_FLOW   := 11,
  eLON_FI_WATER_FLOW_TAMPER := 12,
  eLON_FI_STATUS_ONLY  := 13,
  eLON_FI_MANUAL_CALL  := 14,
  eLON_FI_FIREMAN_CALL := 15,
  eLON_FI_UNIVERSAL    := 16
)
END_TYPE

```

eLON\_FI\_NUL: Invalid Value

eLON\_FI\_UNDEFINED: Initiator is undefined

eLON\_FI\_THERMAL\_FIXED: Initiator is thermal fixed (heat)

eLON\_FI\_SMOKE\_ION: Initiator is smoke and ion

eLON\_FI\_MULTI\_ION\_THERMAL: Initiator is multi-ion and thermal

eLON\_FI\_SMOKE\_PHOTO: Initiator is smoke and photo

eLON\_FI\_MULTI\_PHOTO\_THERMAL: Initiator is multi-photo and thermal

eLON\_FI\_MULTI\_PHOTO\_ION: Initiator is multi-photo and ion

eLON\_FI\_MULTI\_PHOTO\_ION\_THERMAL: Initiator is multi-photo, ion and thermal

eLON\_FI\_THERMAL\_ROR: Initiator is thermal fixed and Rate of Rise

eLON\_FI\_MULTI\_THERMAL\_ROR: Initiator is multi-thermal and Rate of Rise

eLON\_FI\_MANUAL\_PULL: Initiator is manual pull

eLON\_FI\_WATER\_FLOW: Initiator is water flow

eLON\_FI\_WATER\_FLOW\_TAMPER: Initiator is water flow and tamper

eLON\_FI\_STATUS\_ONLY: Initiator is status only

eLON\_FI\_MANUAL\_CALL: Initiator is a manual call point

eLON\_FI\_FIREMAN\_CALL: Initiator is a fireman call point

eLON\_FI\_UNIVERSAL: General purpose initiator definition

### 7.3.33 E\_LON\_fire\_test\_t

Genutzt durch: SNVT\_fire\_test

```

TYPE E_LON_fire_test_t :
(
  eLON_FT_NUL      := -1,
  eLON_FT_NORMAL   := 0,
  eLON_FT_RESET    := 1,
  eLON_FT_TEST     := 2,

```

```
eLON_FT_NOTEST := 3
)
END_TYPE
```

eLON\_FT\_NUL: Invalid Value

eLON\_FT\_NORMAL: Return object to normal status

eLON\_FT\_RESET: Perform a RESET function (for smoke detectors)

eLON\_FT\_TEST: Go into TEST mode

eLON\_FT\_NOTEST: Exit TEST mode

### 7.3.34 E\_LON\_flow\_direction\_t

Genutzt durch: SNVT\_flow\_dir

```
TYPE E_LON_flow_direction_t :
(
  eLON_FD_NUL      := -1,
  eLON_FD_NONE     := 0,
  eLON_FD_OUT      := 1,
  eLON_FD_IN       := 2,
  eLON_FD_ANY      := 3
)
END_TYPE
```

eLON\_FD\_NUL: Invalid Value

eLON\_FD\_NONE: No flow/movement allowed

eLON\_FD\_OUT: Exit/out/away direction only

eLON\_FD\_IN: Entry/in/toward direction only

eLON\_FD\_ANY: No restriction on flow/movement

### 7.3.35 E\_LON\_gfci\_status\_t

Genutzt durch: SNVT\_gfci\_status

```
TYPE E_LON_gfci_status_t :
(
  eLON_GFCI_NUL          := -1,
  eLON_GFCI_UNKNOWN     := 0,
  eLON_GFCI_NORMAL      := 1,
  eLON_GFCI_TRIPPED     := 2,
  eLON_GFCI_TEST_FAILED := 3,
  eLON_GFCI_TEST_PASSED := 4,
  eLON_GFCI_TEST_NOW    := 5
)
END_TYPE
```

eLON\_GFCI\_NUL: Invalid Value

eLON\_GFCI\_UNKNOWN: Unknown response

eLON\_GFCI\_NORMAL: Normal GFCI operating condition

eLON\_GFCI\_TRIPPED: A ground-fault has caused the GFCI to interrupt the circuit

eLON\_GFCI\_TEST\_FAILED: The GFCI failed testing

eLON\_GFCI\_TEST\_PASSED: The GFCI passed testing

eLON\_GFCI\_TEST\_NOW: The GFCI needs to be tested

### 7.3.36 E\_LON\_hvac\_hvt\_t

Genutzt durch: SNVT\_hvac\_type

```

TYPE E_LON_hvac_hvt_t :
(
  eLON_HVT_NUL      := -1,
  eLON_HVT_GENERIC := 0,
  eLON_HVT_FAN_COIL := 1,
  eLON_HVT_VAV     := 2,
  eLON_HVT_HEAT_PUMP := 3,
  eLON_HVT_ROOFTOP := 4,
  eLON_HVT_UNIT_VENT := 5,
  eLON_HVT_CHILL_CEIL := 6,
  eLON_HVT_RADIATOR := 7,
  eLON_HVT_AHU     := 8,
  eLON_HVT_SELF_CONT := 9
)
END_TYPE

```

eLON\_HVT\_NUL: Invalid Value

eLON\_HVT\_GENERIC: Generic

eLON\_HVT\_FAN\_COIL: Fan Coil

eLON\_HVT\_VAV: Variable Air Volume Terminal

eLON\_HVT\_HEAT\_PUMP: Heat Pump

eLON\_HVT\_ROOFTOP: Rooftop Unit

eLON\_HVT\_UNIT\_VENT: Unit Ventilator

eLON\_HVT\_CHILL\_CEIL: Chilled Ceiling

eLON\_HVT\_RADIATOR: Radiator

eLON\_HVT\_AHU: Air Handling Unit

eLON\_HVT\_SELF\_CONT: Self-Contained Unit

### 7.3.37 E\_LON\_hvac\_overid\_t

Genutzt durch: SNVT\_hvac\_overid

```

TYPE E_LON_hvac_overid_t :
(
  eLON_HVO_NUL      := -1,
  eLON_HVO_OFF     := 0,
  eLON_HVO_POSITION := 1,
  eLON_HVO_FLOW_VALUE := 2,
  eLON_HVO_FLOW_PERCENT := 3,
  eLON_HVO_OPEN    := 4,
  eLON_HVO_CLOSE   := 5,
  eLON_HVO_MINIMUM := 6,
  eLON_HVO_MAXIMUM := 7,
  eLON_HVO_UNUSED8 := 8,
  eLON_HVO_UNUSED9 := 9,
  eLON_HVO_UNUSED10 := 10,
  eLON_HVO_UNUSED11 := 11,
  eLON_HVO_UNUSED12 := 12,
  eLON_HVO_UNUSED13 := 13,
  eLON_HVO_UNUSED14 := 14,
  eLON_HVO_UNUSED15 := 15,
  eLON_HVO_UNUSED16 := 16,
  eLON_HVO_POSITION_1 := 17,
  eLON_HVO_FLOW_VALUE_1 := 18,
  eLON_HVO_FLOW_PERCENT_1 := 19,
  eLON_HVO_OPEN_1 := 20,
  eLON_HVO_CLOSE_1 := 21,
  eLON_HVO_MINIMUM_1 := 22,
  eLON_HVO_MAXIMUM_1 := 23,
  eLON_HVO_UNUSED24 := 24,
  eLON_HVO_UNUSED25 := 25,
  eLON_HVO_UNUSED26 := 26,
  eLON_HVO_UNUSED27 := 27,
  eLON_HVO_UNUSED28 := 28,
  eLON_HVO_UNUSED29 := 29,
  eLON_HVO_UNUSED30 := 30,

```

```

eLON_HVO_UNUSED31 := 31,
eLON_HVO_UNUSED32 := 32,
eLON_HVO_POSITION_2 := 33,
eLON_HVO_FLOW_VALUE_2 := 34,
eLON_HVO_FLOW_PERCENT_2 := 35,
eLON_HVO_OPEN_2 := 36,
eLON_HVO_CLOSE_2 := 37,
eLON_HVO_MINIMUM_2 := 38,
eLON_HVO_MAXIMUM_2 := 39,
eLON_HVO_UNUSED40 := 40,
eLON_HVO_UNUSED41 := 41,
eLON_HVO_UNUSED42 := 42,
eLON_HVO_UNUSED43 := 43,
eLON_HVO_UNUSED44 := 44,
eLON_HVO_UNUSED45 := 45,
eLON_HVO_UNUSED46 := 46,
eLON_HVO_UNUSED47 := 47,
eLON_HVO_UNUSED48 := 48
)
END_TYPE

```

eLON\_HVO\_NUL: Invalid Value

eLON\_HVO\_OFF: Not overridden

eLON\_HVO\_POSITION:

eLON\_HVO\_FLOW\_VALUE: Override flow in liters/sec - use flow field

eLON\_HVO\_FLOW\_PERCENT: Override flow percentage - use percent field

eLON\_HVO\_OPEN: Override to position = 100%

eLON\_HVO\_CLOSE: Override to position = 0%

eLON\_HVO\_MINIMUM: Override to configured minimum

eLON\_HVO\_MAXIMUM: Override to configured maximum

eLON\_HVO\_UNUSED8:

eLON\_HVO\_UNUSED9:

eLON\_HVO\_UNUSED10:

eLON\_HVO\_UNUSED11:

eLON\_HVO\_UNUSED12:

eLON\_HVO\_UNUSED13:

eLON\_HVO\_UNUSED14:

eLON\_HVO\_UNUSED15:

eLON\_HVO\_UNUSED16:

eLON\_HVO\_POSITION\_1:

eLON\_HVO\_FLOW\_VALUE\_1: Override flow in liters/sec - use flow field

eLON\_HVO\_FLOW\_PERCENT\_1: Override flow percentage - use percent field

eLON\_HVO\_OPEN\_1: Override to position = 100%

eLON\_HVO\_CLOSE\_1: Override to position = 0%

eLON\_HVO\_MINIMUM\_1: Override to configured minimum

eLON\_HVO\_MAXIMUM\_1: Override to configured maximum

eLON\_HVO\_UNUSED24:

eLON\_HVO\_UNUSED25:

eLON\_HVO\_UNUSED26:  
 eLON\_HVO\_UNUSED27:  
 eLON\_HVO\_UNUSED28:  
 eLON\_HVO\_UNUSED29:  
 eLON\_HVO\_UNUSED30:  
 eLON\_HVO\_UNUSED31:  
 eLON\_HVO\_UNUSED32:  
 eLON\_HVO\_POSITION\_2:  
 eLON\_HVO\_FLOW\_VALUE\_2: Override flow in liters/sec - use flow field  
 eLON\_HVO\_FLOW\_PERCENT\_2: Override flow percentage - use percent field  
 eLON\_HVO\_OPEN\_2: Override to position = 100%  
 eLON\_HVO\_CLOSE\_2: Override to position = 0%  
 eLON\_HVO\_MINIMUM\_2: Override to configured minimum  
 eLON\_HVO\_MAXIMUM\_2: Override to configured maximum  
 eLON\_HVO\_UNUSED40:  
 eLON\_HVO\_UNUSED41:  
 eLON\_HVO\_UNUSED42:  
 eLON\_HVO\_UNUSED43:  
 eLON\_HVO\_UNUSED44:  
 eLON\_HVO\_UNUSED45:  
 eLON\_HVO\_UNUSED46:  
 eLON\_HVO\_UNUSED47:  
 eLON\_HVO\_UNUSED48:

### 7.3.38 E\_LON\_hvac\_t

Genutzt durch: SNVT\_chlr\_status / SNVT\_hvac\_mode / SNVT\_hvac\_status

```

TYPE E_LON_hvac_t :
(
    eLON_HVAC_NUL           := -1,
    eLON_HVAC_AUTO         := 0,
    eLON_HVAC_HEAT         := 1,
    eLON_HVAC_MRNG_WRMUP  := 2,
    eLON_HVAC_COOL        := 3,
    eLON_HVAC_NIGHT_PURGE := 4,
    eLON_HVAC_PRE_COOL    := 5,
    eLON_HVAC_OFF         := 6,
    eLON_HVAC_TEST        := 7,
    eLON_HVAC_EMERG_HEAT  := 8,
    eLON_HVAC_FAN_ONLY    := 9,
    eLON_HVAC_FREE_COOL   := 10,
    eLON_HVAC_ICE         := 11,
    eLON_HVAC_MAX_HEAT    := 12,
    eLON_HVAC_ECONOMY     := 13,
    eLON_HVAC_DEHUMID     := 14,
    eLON_HVAC_CALIBRATE   := 15,
    eLON_HVAC_EMERG_COOL  := 16,
    eLON_HVAC_EMERG_STEAM := 17,
    eLON_HVAC_MAX_COOL    := 18,
    eLON_HVAC_HVC_LOAD    := 19,

```

```
eLON_HVAC_NO_LOAD      := 20
)
END_TYPE
```

eLON\_HVAC\_NUL: Invalid value

eLON\_HVAC\_AUTO: Controller automatically changes between application modes

eLON\_HVAC\_HEAT: Heating only

eLON\_HVAC\_MRNG\_WRMUP: Application-specific morning warm-up

eLON\_HVAC\_COOL: Cooling only

eLON\_HVAC\_NIGHT\_PURGE: Application-specific night purge

eLON\_HVAC\_PRE\_COOL: Application-specific pre-cool

eLON\_HVAC\_OFF: Controller not controlling outputs

eLON\_HVAC\_TEST: Equipment being tested

eLON\_HVAC\_EMERG\_HEAT: Emergency heat mode (heat pump)

eLON\_HVAC\_FAN\_ONLY: Air not conditioned, fan turned on

eLON\_HVAC\_FREE\_COOL: Cooling with compressor not running

eLON\_HVAC\_ICE: Ice-making mode

eLON\_HVAC\_MAX\_HEAT: Maximum heating mode

eLON\_HVAC\_ECONOMY: Economic Heat/Cool mode

eLON\_HVAC\_DEHUMID: Dehumidification mode

eLON\_HVAC\_CALIBRATE: Calibration mode

eLON\_HVAC\_EMERG\_COOL: Emergency cool mode

eLON\_HVAC\_EMERG\_STEAM: Emergency steam mode

eLON\_HVAC\_MAX\_COOL:

eLON\_HVAC\_HVC\_LOAD:

eLON\_HVAC\_NO\_LOAD:

### 7.3.39 E\_LON\_learn\_mode\_t

Genutzt durch: SNVT\_preset

```
TYPE E_LON_learn_mode_t :
(
  eLON_LN_NUL      := -1,
  eLON_LN_RECALL   := 0,
  eLON_LN_LEARN_CURRENT := 1,
  eLON_LN_LEARN_VALUE := 2,
  eLON_LN_REPORT_VALUE := 3
)
END_TYPE
```

eLON\_LN\_NUL: Invalid Value

eLON\_LN\_RECALL: Recall

eLON\_LN\_LEARN\_CURRENT: Learn present value

eLON\_LN\_LEARN\_VALUE: Learn given value

eLON\_LN\_REPORT\_VALUE: Report the value

### 7.3.40 E\_LON\_log\_status\_t

Genutzt durch: SCPTlogRecord / SNVT\_log\_status

```
TYPE E_LON_log_status_t :
(
  eLON_LS_NUL           := -1,
  eLON_LS_ENABLED      := 0,
  eLON_LS_DISABLED     := 1,
  eLON_LS_FULL         := 2,
  eLON_LS_OVERFLOW_ERR := 3,
  eLON_LS_INVALID_LOG_ERR := 4,
  eLON_LS_APP_ERR      := 5
)
END_TYPE
```

eLON\_LS\_NUL: Invalid value

eLON\_LS\_ENABLED: Log enabled

eLON\_LS\_DISABLED: Log disabled

eLON\_LS\_FULL: Log enabled and full

eLON\_LS\_OVERFLOW\_ERR: Log enabled, overflow occurred

eLON\_LS\_INVALID\_LOG\_ERR: Invalid log selected

eLON\_LS\_APP\_ERR: Other application error

### 7.3.41 E\_LON\_motor\_state\_t

Genutzt durch: SNVT\_motor\_state / SNVT\_pumpset\_mn

```
TYPE E_LON_motor_state_t :
(
  eLON_MOTOR_NUL           := -1,
  eLON_MOTOR_STOPPED      := 0,
  eLON_MOTOR_STARTING     := 1,
  eLON_MOTOR_ACCELERATING := 2,
  eLON_MOTOR_AT_STANDBY   := 3,
  eLON_MOTOR_AT_NORMAL    := 4,
  eLON_MOTOR_AT_REFERENCE := 5,
  eLON_MOTOR_DECELERATING := 6,
  eLON_MOTOR_STOPPING     := 7
)
END_TYPE
```

eLON\_MOTOR\_NUL: The state of the motor is unknown (invalid value)

eLON\_MOTOR\_STOPPED: The motor is not running

eLON\_MOTOR\_STARTING: The motor is performing its start-up sequence

eLON\_MOTOR\_ACCELERATING: The motor is running. Speed is increasing.

eLON\_MOTOR\_AT\_STANDBY: The motor is running in its standby mode

eLON\_MOTOR\_AT\_NORMAL: The motor is running in its normal operational mode

eLON\_MOTOR\_AT\_REFERENCE: The motor is running at its reference speed.

eLON\_MOTOR\_DECELERATING: The motor is running. Speed is decreasing.

eLON\_MOTOR\_STOPPING: The motor is running, beginning its shutdown sequence.

### 7.3.42 E\_LON\_nv\_type\_category\_t

Genutzt durch: SNVT\_nv\_type

```
TYPE E_LON_nv_type_category_t :
(
  eLON_NVT_CAT_NUL       := -1,
```

```

eLON_NVT_CAT_INITIAL      := 0,
eLON_NVT_CAT_SIGNED_CHAR := 1,
eLON_NVT_CAT_UNSIGNED_CHAR := 2,
eLON_NVT_CAT_SIGNED_SHORT := 3,
eLON_NVT_CAT_UNSIGNED_SHORT := 4,
eLON_NVT_CAT_SIGNED_LONG := 5,
eLON_NVT_CAT_UNSIGNED_LONG := 6,
eLON_NVT_CAT_ENUM        := 7,
eLON_NVT_CAT_ARRAY       := 8,
eLON_NVT_CAT_STRUCT      := 9,
eLON_NVT_CAT_UNION       := 10,
eLON_NVT_CAT_BITFIELD    := 11,
eLON_NVT_CAT_FLOAT       := 12,
eLON_NVT_CAT_SIGNED_QUAD := 13,
eLON_NVT_CAT_REFERENCE   := 14
)
END_TYPE

```

eLON\_NVT\_CAT\_NUL: Invalid Value

eLON\_NVT\_CAT\_INITIAL:

eLON\_NVT\_CAT\_SIGNED\_CHAR: 8-bit signed character

eLON\_NVT\_CAT\_UNSIGNED\_CHAR: 8-bit unsigned character

eLON\_NVT\_CAT\_SIGNED\_SHORT: 8-bit signed integer

eLON\_NVT\_CAT\_UNSIGNED\_SHORT: 8-bit unsigned integer

eLON\_NVT\_CAT\_SIGNED\_LONG: 16-bit signed integer

eLON\_NVT\_CAT\_UNSIGNED\_LONG: 16-bit unsigned integer

eLON\_NVT\_CAT\_ENUM: 8-bit enumeration

eLON\_NVT\_CAT\_ARRAY: Array

eLON\_NVT\_CAT\_STRUCT: Structure

eLON\_NVT\_CAT\_UNION: Union

eLON\_NVT\_CAT\_BITFIELD: Bitfield

eLON\_NVT\_CAT\_FLOAT: 32-bit IEC 60559 (IEEE 754) floating-point value

eLON\_NVT\_CAT\_SIGNED\_QUAD: 32-bit signed integer

eLON\_NVT\_CAT\_REFERENCE: Reference type

### 7.3.43 E\_LON\_object\_request\_t

Genutzt durch: SNVT\_obj\_request

```

TYPE E_LON_object_request_t :
(
  eLON_RQ_NUL           := -1,
  eLON_RQ_NORMAL       := 0,
  eLON_RQ_DISABLED     := 1,
  eLON_RQ_UPDATE_STATUS := 2,
  eLON_RQ_SELF_TEST    := 3,
  eLON_RQ_UPDATE_ALARM := 4,
  eLON_RQ_REPORT_MASK  := 5,
  eLON_RQ_OVERRIDE     := 6,
  eLON_RQ_ENABLE       := 7,
  eLON_RQ_RMV_OVERRIDE := 8,
  eLON_RQ_CLEAR_STATUS := 9,
  eLON_RQ_CLEAR_ALARM  := 10,
  eLON_RQ_ALARM_NOTIFY_ENABLED := 11,
  eLON_RQ_ALARM_NOTIFY_DISABLED := 12,
  eLON_RQ_MANUAL_CTRL  := 13,
  eLON_RQ_REMOTE_CTRL  := 14,
  eLON_RQ_PROGRAM      := 15,
  eLON_RQ_CLEAR_RESET  := 16,
  eLON_RQ_RESET        := 17,
)

```



```
eLON_RQ_CLEAR_LOG      := 18
)
END_TYPE
```

eLON\_RQ\_NUL: Invalid Value

eLON\_RQ\_NORMAL: Enable object and remove override

eLON\_RQ\_DISABLED: Disable object

eLON\_RQ\_UPDATE\_STATUS: Report object status

eLON\_RQ\_SELF\_TEST: Perform object self-test

eLON\_RQ\_UPDATE\_ALARM: Update alarm status

eLON\_RQ\_REPORT\_MASK: Report status bit mask

eLON\_RQ\_OVERRIDE: Override object

eLON\_RQ\_ENABLE: Enable object

eLON\_RQ\_RMV\_OVERRIDE: Remove object override

eLON\_RQ\_CLEAR\_STATUS: Clear object status

eLON\_RQ\_CLEAR\_ALARM: Clear object alarm

eLON\_RQ\_ALARM\_NOTIFY\_ENABLED: Enable alarm notification

eLON\_RQ\_ALARM\_NOTIFY\_DISABLED: Disable alarm notification

eLON\_RQ\_MANUAL\_CTRL: Enable object for manual control

eLON\_RQ\_REMOTE\_CTRL: Enable object for remote control

eLON\_RQ\_PROGRAM: Enable programming of special configuration properties

eLON\_RQ\_CLEAR\_RESET: Clear reset-complete flag (reset\_complete)

eLON\_RQ\_RESET: Execute reset-sequence of object

eLON\_RQ\_CLEAR\_LOG: Clear data log

### 7.3.44 E\_LON\_occup\_t

Genutzt durch: SNVT\_occupancy / SNVT\_tod\_event

```
TYPE E_LON_occup_t :
```

```
(
  eLON_OC_NUL      := -1,
  eLON_OC_OCCUPIED := 0,
  eLON_OC_UNOCCUPIED := 1,
  eLON_OC_BYPASS   := 2,
  eLON_OC_STANDBY  := 3
)
END_TYPE
```

eLON\_OC\_NUL: Invalid Value

eLON\_OC\_OCCUPIED: Area is occupied

eLON\_OC\_UNOCCUPIED: Area is unoccupied

eLON\_OC\_BYPASS: Area is temporarily occupied for the bypass period

eLON\_OC\_STANDBY: Area is temporarily unoccupied

### 7.3.45 E\_LON\_override\_t

Genutzt durch: SNVT\_override

```
TYPE E_LON_override_t :  
(  
  eLON_OV_NUL      := -1,  
  eLON_OV_RETAIN   := 0,  
  eLON_OV_SPECIFIED := 1,  
  eLON_OV_DEFAULT  := 2  
)  
END_TYPE
```

eLON\_OV\_NUL: Invalid Value

eLON\_OV\_RETAIN: Retain current level

eLON\_OV\_SPECIFIED: Go to specified level

eLON\_OV\_DEFAULT: Go to default level

### 7.3.46 E\_LON\_pan\_dir\_t

Genutzt durch: SNVT\_ptz

```
TYPE E_LON_pan_dir_t :  
(  
  eLON_PAN_NUL      := -1,  
  eLON_PAN_STOP     := 0,  
  eLON_PAN_RIGHT    := 1,  
  eLON_PAN_LEFT     := 2  
)  
END_TYPE
```

eLON\_PAN\_NUL: Invalid Value

eLON\_PAN\_STOP: Stop panning

eLON\_PAN\_RIGHT: Pan to the right

eLON\_PAN\_LEFT: Pan to the left

### 7.3.47 E\_LON\_priority\_level\_t

Genutzt durch: SNVT\_alarm / SNVT\_alarm\_2 / SNVT\_pumpset\_mn

```
TYPE E_LON_priority_level_t :  
(  
  eLON_PR_NUL      := -1,  
  eLON_PR_LEVEL_0 := 0,  
  eLON_PR_LEVEL_1 := 1,  
  eLON_PR_LEVEL_2 := 2,  
  eLON_PR_LEVEL_3 := 3,  
  eLON_PR_1       := 4,  
  eLON_PR_2       := 5,  
  eLON_PR_3       := 6,  
  eLON_PR_4       := 7,  
  eLON_PR_6       := 8,  
  eLON_PR_8       := 9,  
  eLON_PR_10      := 10,  
  eLON_PR_16      := 11  
)  
END_TYPE
```

eLON\_PR\_NUL: Invalid Value

eLON\_PR\_LEVEL\_0: Lowest alarm priority level

eLON\_PR\_LEVEL\_1:

eLON\_PR\_LEVEL\_2:

eLON\_PR\_LEVEL\_3: Highest alarm priority level

eLON\_PR\_1: Life Safety Fire Alarms (BACnet Priority 2)

eLON\_PR\_2: Property Safety Fire Alarms (BACnet Priority 3)

eLON\_PR\_3: Fire Supervisory Alarm (BACnet Priority 4)

eLON\_PR\_4: Fire Trouble/Fault (Display) (BACnet Priority 5)

eLON\_PR\_6: Fire Pre-Alarm, HVAC Critical Equipment Alarm (BACnet Priority 6)

eLON\_PR\_8: HVAC Alarms (BACnet Priority 8)

eLON\_PR\_10: HVAC Critical Equipment RTN, Fire RTN (Display) (BACnet Priority 10)

eLON\_PR\_16: HVAC RTN (lowest priority) (BACnet Priority 16)

### 7.3.48 E\_LON\_privacyzone\_t

Genutzt durch: SNVT\_privacyzone

```
TYPE E_LON_privacyzone_t :
(
  eLON_PZ_NUL           := -1,
  eLON_PZ_DISABLE      := 0,
  eLON_PZ_ENABLE       := 1,
  eLON_PZ_UPPER_LEFT   := 2,
  eLON_PZ_LOWER_RIGHT  := 3,
  eLON_PZ_ENTER        := 4,
  eLON_PZ_EXIT         := 5
)
END_TYPE
```

eLON\_PZ\_NUL: Invalid value

eLON\_PZ\_DISABLE: Disable privacy zone warning

eLON\_PZ\_ENABLE: Enable privacy zone warning

eLON\_PZ\_UPPER\_LEFT: Set upper left corner

eLON\_PZ\_LOWER\_RIGHT: Set lower right corner

eLON\_PZ\_ENTER: Privacy zone enter warning

eLON\_PZ\_EXIT: Privacy zone exit message

### 7.3.49 E\_LON\_rail\_audio\_sensor\_type\_t

Genutzt durch: SNVT\_rac\_ctrl / SNVT\_rac\_req

```
TYPE E_LON_rail_audio_sensor_type_t :
(
  eLON_RAST_NUL           := -1,
  eLON_RAST_CU_TYPE_1     := 0,
  eLON_RAST_CU_TYPE_2     := 1,
  eLON_RAST_CU_TYPE_3     := 2,
  eLON_RAST_CU_TYPE_4     := 3,
  eLON_RAST_LS_LINE_1     := 4,
  eLON_RAST_LS_LINE_2     := 5,
  eLON_RAST_LS_LINE_3     := 6,
  eLON_RAST_LS_LINE_4     := 7,
  eLON_RAST_LS_LINE_5     := 8,
  eLON_RAST_LS_LINE_6     := 9,
  eLON_RAST_LS_LINE_7     := 10,
  eLON_RAST_LS_LINE_8     := 11,
  eLON_RAST_PAU           := 12,
  eLON_RAST_CFA_TYPE_1    := 13,
  eLON_RAST_CFA_TYPE_2    := 14,
  eLON_RAST_CFA_TYPE_3    := 15,
  eLON_RAST_CFA_TYPE_4    := 16,
  eLON_RAST_DVA           := 17,
  eLON_RAST_ET_TYPE_1     := 18,
  eLON_RAST_ET_TYPE_2     := 19,
  eLON_RAST_USERDEF_TYPE_1 := 20,
  eLON_RAST_USERDEF_TYPE_2 := 21,
  eLON_RAST_USERDEF_TYPE_3 := 22,
)
```

```
eLON_RAST_USERDEF_TYPE_4 := 23
)
END_TYPE
```

eLON\_RAST\_NUL: Invalid Value  
eLON\_RAST\_CU\_TYPE\_1: CU Type 1  
eLON\_RAST\_CU\_TYPE\_2: CU Type 2  
eLON\_RAST\_CU\_TYPE\_3:  
eLON\_RAST\_CU\_TYPE\_4: CU Type 4  
eLON\_RAST\_LS\_LINE\_1: LS Line 1  
eLON\_RAST\_LS\_LINE\_2: LS Line 2  
eLON\_RAST\_LS\_LINE\_3: LS Line 3  
eLON\_RAST\_LS\_LINE\_4: LS Line 4  
eLON\_RAST\_LS\_LINE\_5: LS Line 5  
eLON\_RAST\_LS\_LINE\_6: LS Line 6  
eLON\_RAST\_LS\_LINE\_7: LS Line 7  
eLON\_RAST\_LS\_LINE\_8: LS Line 8  
eLON\_RAST\_PAU: Public-Address Unit  
eLON\_RAST\_CFA\_TYPE\_1: CFA Type 1  
eLON\_RAST\_CFA\_TYPE\_2: CFA Type 2  
eLON\_RAST\_CFA\_TYPE\_3: CFA Type 3  
eLON\_RAST\_CFA\_TYPE\_4: CFA Type 4  
eLON\_RAST\_DVA: DVA  
eLON\_RAST\_ET\_TYPE\_1: ET Type 1  
eLON\_RAST\_ET\_TYPE\_2: ET Type 2  
eLON\_RAST\_USERDEF\_TYPE\_1: User-defined Type 1  
eLON\_RAST\_USERDEF\_TYPE\_2: User-defined Type 2  
eLON\_RAST\_USERDEF\_TYPE\_3: User-defined Type 3  
eLON\_RAST\_USERDEF\_TYPE\_4: User-defined Type 4

### 7.3.50 E\_LON\_rail\_audio\_type\_t

Genutzt durch: SNVT\_rac\_ctrl / SNVT\_rac\_req

```
TYPE E_LON_rail_audio_type_t :
(
  eLON_RAT_NUL           := -1,
  eLON_RAT_IC_REQ       := 0,
  eLON_RAT_IC_JOIN      := 1,
  eLON_RAT_IC_QUIT      := 2,
  eLON_RAT_IC_END       := 3,
  eLON_RAT_HW_RADIO_REQ := 4,
  eLON_RAT_HW_RADIO_END := 5,
  eLON_RAT_HW_PA_REQ    := 6,
  eLON_RAT_HW_PA_END    := 7,
  eLON_RAT_SW_PA_REQ    := 8,
  eLON_RAT_SW_PA_END    := 9,
  eLON_RAT_SW_PA_OR_REQ := 10,
  eLON_RAT_SW_PA_OR_END := 11,
```

```
eLON_RAT_PAU_REQ      := 12,
eLON_RAT_PAU_ACCEPT  := 13,
eLON_RAT_PAU_CALL    := 14,
eLON_RAT_PAU_END     := 15,
eLON_RAT_ENTERT_REQ  := 16,
eLON_RAT_ENTERT_END  := 17
)
END_TYPE
```

eLON\_RAT\_NUL:

eLON\_RAT\_IC\_REQ:

eLON\_RAT\_IC\_JOIN:

eLON\_RAT\_IC\_QUIT:

eLON\_RAT\_IC\_END:

eLON\_RAT\_HW\_RADIO\_REQ:

eLON\_RAT\_HW\_RADIO\_END:

eLON\_RAT\_HW\_PA\_REQ:

eLON\_RAT\_HW\_PA\_END:

eLON\_RAT\_SW\_PA\_REQ:

eLON\_RAT\_SW\_PA\_END:

eLON\_RAT\_SW\_PA\_OR\_REQ:

eLON\_RAT\_SW\_PA\_OR\_END:

eLON\_RAT\_PAU\_REQ:

eLON\_RAT\_PAU\_ACCEPT:

eLON\_RAT\_PAU\_CALL:

eLON\_RAT\_PAU\_END:

eLON\_RAT\_ENTERT\_REQ:

eLON\_RAT\_ENTERT\_END:

### 7.3.51 E\_LON\_reg\_val\_unit\_t

Genutzt durch: SNVT\_reg\_val / SNVT\_reg\_val\_ts

```
TYPE E_LON_reg_val_unit_t :
(
  eLON_RVU_NUL      := -1,
  eLON_RVU_NONE    := 0,
  eLON_RVU_W       := 1,
  eLON_RVU_KW      := 2,
  eLON_RVU_MW      := 3,
  eLON_RVU_GW      := 4,
  eLON_RVU_VAR     := 5,
  eLON_RVU_KVAR    := 6,
  eLON_RVU_MVAR    := 7,
  eLON_RVU_GVAR    := 8,
  eLON_RVU_WH      := 9,
  eLON_RVU_KWH     := 10,
  eLON_RVU_MWH     := 11,
  eLON_RVU_GWH     := 12,
  eLON_RVU_VARH    := 13,
  eLON_RVU_KVARH   := 14,
  eLON_RVU_MVARH   := 15,
  eLON_RVU_GVARH   := 16,
  eLON_RVU_V       := 17,
  eLON_RVU_A       := 18,
  eLON_RVU_COSF    := 19,
```

```
eLON_RVU_M3      := 20,
eLON_RVU_L       := 21,
eLON_RVU_ML      := 22,
eLON_RVU_USGAL   := 23,
eLON_RVU_GJ      := 24,
eLON_RVU_MJ      := 25,
eLON_RVU_MCAL    := 26,
eLON_RVU_KCAL    := 27,
eLON_RVU_MBTU    := 28,
eLON_RVU_KBTU    := 29,
eLON_RVU_MJH     := 30,
eLON_RVU_MLS     := 31,
eLON_RVU_LS      := 32,
eLON_RVU_M3S     := 33,
eLON_RVU_C       := 34,
eLON_RVU_LH      := 35,
eLON_RVU_VA      := 36,
eLON_RVU_KVA     := 37,
eLON_RVU_MVA     := 38,
eLON_RVU_GVA     := 39,
eLON_RVU_VAH     := 40,
eLON_RVU_KVAH    := 41,
eLON_RVU_MVAH    := 42,
eLON_RVU_GVAH    := 43
)
END_TYPE
```

eLON\_RVU\_NUL: invalid unit of measure (INVALID)

eLON\_RVU\_NONE: no units specified ( )

eLON\_RVU\_W: Watts (W)

eLON\_RVU\_KW: kiloWatts (kW)

eLON\_RVU\_MW: megaWatts (MW)

eLON\_RVU\_GW: gigaWatts (GW)

eLON\_RVU\_VAR: Volt-Amperes reactive (var)

eLON\_RVU\_KVAR: kilo-Volt-Amperes reactive (kvar)

eLON\_RVU\_MVAR: mega-Volt-Amperes reactive (Mvar)

eLON\_RVU\_GVAR: giga-Volt-Amperes reactive (Gvar)

eLON\_RVU\_WH: Watt-hour (Wh)

eLON\_RVU\_KWH: kiloWatt-hour (kWh)

eLON\_RVU\_MWH: megaWatt-hour (MWh)

eLON\_RVU\_GWH: gigaWatt-hour (GWh)

eLON\_RVU\_VARH: Volt-Amperes reactive -hour (varh)

eLON\_RVU\_KVARH: kilo-Volt-Amperes reactive -hour (kvarh)

eLON\_RVU\_MVARH: mega-Volt-Amperes reactive -hour (Mvarh)

eLON\_RVU\_GVARH: giga-Volt-Amperes reactive -hour (Gvarh)

eLON\_RVU\_V: Volts (V)

eLON\_RVU\_A: Amps (A)

eLON\_RVU\_COSF: (cosf)

eLON\_RVU\_M3: cubic meters (m<sup>3</sup>)(cu.m)

eLON\_RVU\_L: liters (l)

eLON\_RVU\_ML: milliliters (ml)

eLON\_RVU\_USGAL: U.S. Gallons (USG)  
 eLON\_RVU\_GJ: giga-Joules (GJ)  
 eLON\_RVU\_MJ: mega-Joules (MJ)  
 eLON\_RVU\_MCAL: megacalories (Mcal)  
 eLON\_RVU\_KCAL: kilocalories (kcal) / Calories (Cal)  
 eLON\_RVU\_MBTU: mega-British thermal units (mBtu)  
 eLON\_RVU\_KBTU: kilo-British thermal units (kBtu)  
 eLON\_RVU\_MJH: mega-Joules per hour (MJ/h)  
 eLON\_RVU\_MLS: milliliters per second (ml/s)  
 eLON\_RVU\_LS: liters per second (l/s)  
 eLON\_RVU\_M3S: cubic-meters per second (m³/s) (cu.m/s)  
 eLON\_RVU\_C: (C)  
 eLON\_RVU\_LH: liters per hour (l/h)  
 eLON\_RVU\_VA: Volt-Amperes (VA)  
 eLON\_RVU\_KVA: kiloVolt-Amperes (kVA)  
 eLON\_RVU\_MVA: megaVolt-Amperes (MVA)  
 eLON\_RVU\_GVA: gigaVolt-Amperes (GVA)  
 eLON\_RVU\_VAH: Volt-Ampere hours (VAh)  
 eLON\_RVU\_KVAH: kiloVolt-Ampere hours (kVAh)  
 eLON\_RVU\_MVAH: megaVolt-Ampere hours (MVAh)  
 eLON\_RVU\_GVAH: giga-Volt-Ampere hours (GVAh)

### 7.3.52 E\_LON\_sblnd\_cmd\_source\_t

Genutzt durch: SNVT\_sblnd\_state

```

TYPE E_LON_sblnd_cmd_source_t :
(
    eLON_SBCS_NUL           := -1,
    eLON_SBCS_LOCAL        := 0,
    eLON_SBCS_GROUP        := 1,
    eLON_SBCS_WIND_SPEED   := 2,
    eLON_SBCS_SUN_LUX      := 3,
    eLON_SBCS_RAIN         := 4,
    eLON_SBCS_FROST        := 5,
    eLON_SBCS_DAWN         := 6,
    eLON_SBCS_DUSK         := 7,
    eLON_SBCS_OUTSIDE_TEMP := 8,
    eLON_SBCS_INDOOR_TEMP  := 9,
    eLON_SBCS_OUTDOOR_RH   := 10,
    eLON_SBCS_INDOOR_RH    := 11,
    eLON_SBCS_ILLUM_LEVEL  := 12,
    eLON_SBCS_SCENE        := 13,
    eLON_SBCS_GLOBAL       := 14,
    eLON_SBCS_WINDOW_CONTACT := 15,
    eLON_SBCS_AUTOMODE_CHANGED := 16,
    eLON_SBCS_OVERRIDE     := 17,
    eLON_SBCS_EMERGENCY    := 18,
    eLON_SBCS_MAINTENANCE  := 19,
    eLON_SBCS_INTRUSION    := 20,
    eLON_SBCS_TERMINAL_LOAD := 21,
    eLON_SBCS_ALARM        := 22,
    eLON_SBCS_OCC_SENSOR   := 23,
    eLON_SBCS_OCC_MAN_CMD  := 24,

```

```
eLON_SBCS_GLARE           := 25,  
eLON_SBCS_ALARM_2        := 26,  
eLON_SBCS_NOTIFY         := 27,  
eLON_SBCS_ELEVATION      := 28,  
eLON_SBCS_AZIMUTH        := 29,  
eLON_SBCS_SET_OVERRIDE   := 30,  
eLON_SBCS_SET_MAINTENANCE := 31,  
eLON_SBCS_TIMER          := 32,  
eLON_SBCS_UNKNOWN        := 127  
)  
END_TYPE
```

eLON\_SBCS\_NUL: Invalid value

eLON\_SBCS\_LOCAL: Local

eLON\_SBCS\_GROUP: Group

eLON\_SBCS\_WIND\_SPEED: Wind speed

eLON\_SBCS\_SUN\_LUX: Sun lux level

eLON\_SBCS\_RAIN: Rain

eLON\_SBCS\_FROST: Frost

eLON\_SBCS\_DAWN: Dawn

eLON\_SBCS\_DUSK: Dusk

eLON\_SBCS\_OUTSIDE\_TEMP: Outside temperature

eLON\_SBCS\_INDOOR\_TEMP: Indoor temperature

eLON\_SBCS\_OUTDOOR\_RH: Outdoor relative humidity

eLON\_SBCS\_INDOOR\_RH: Indoor relative humidity

eLON\_SBCS\_ILLUM\_LEVEL: Illumination level

eLON\_SBCS\_SCENE: Scene

eLON\_SBCS\_GLOBAL: Global

eLON\_SBCS\_WINDOW\_CONTACT: Window contact

eLON\_SBCS\_AUTOMODE\_CHANGED: Auto-mode changed

eLON\_SBCS\_OVERRIDE: Override

eLON\_SBCS\_EMERGENCY: Emergency

eLON\_SBCS\_MAINTENANCE: Maintenance

eLON\_SBCS\_INTRUSION: Intrusion

eLON\_SBCS\_TERMINAL\_LOAD: Terminal load

eLON\_SBCS\_ALARM: Alarm

eLON\_SBCS\_OCC\_SENSOR: Occupancy sensor

eLON\_SBCS\_OCC\_MAN\_CMD: Occupancy manual command

eLON\_SBCS\_GLARE: Glare

eLON\_SBCS\_ALARM\_2: Alarm 2

eLON\_SBCS\_NOTIFY: Notify

eLON\_SBCS\_ELEVATION: Elevation

eLON\_SBCS\_AZIMUTH: Azimuth



eLON\_SBCS\_SET\_OVERRIDE: Set override

eLON\_SBCS\_SET\_MAINTENANCE: Set maintenance

eLON\_SBCS\_TIMER: Timer

eLON\_SBCS\_UNKNOWN: Unknown command source

### 7.3.53 E\_LON\_sblnd\_error\_t

Genutzt durch: SNVT\_sblnd\_state

```

TYPE E_LON_sblnd_error_t :
(
  eLON_SBE_NUL           := -1,
  eLON_SBE_NO_ERROR     := 0,
  eLON_SBE_IN_PROGRESS  := 1,
  eLON_SBE_LIMITS       := 2,
  eLON_SBE_OBSTACLE_UP  := 3,
  eLON_SBE_OBSTACLE_DOWN := 4,
  eLON_SBE_OVERHEAT     := 5,
  eLON_SBE_POWER        := 6,
  eLON_SBE_SENSOR       := 7,
  eLON_SBE_MOTOR_CIRCUIT := 8,
  eLON_SBE_FUSE         := 9,
  eLON_SBE_REFERENCE_LOST := 10,
  eLON_SBE_HOST_COMM    := 11,
  eLON_SBE_VOLTAGE_1    := 12,
  eLON_SBE_VOLTAGE_2    := 13,
  eLON_SBE_CONTROLLER   := 14
)
END_TYPE

```

eLON\_SBE\_NUL: Invalid Value

eLON\_SBE\_NO\_ERROR: No error

eLON\_SBE\_IN\_PROGRESS: In progress

eLON\_SBE\_LIMITS: Limits

eLON\_SBE\_OBSTACLE\_UP: Obstacle up

eLON\_SBE\_OBSTACLE\_DOWN: Obstacle down

eLON\_SBE\_OVERHEAT: Overheat

eLON\_SBE\_POWER: Power

eLON\_SBE\_SENSOR: Sensor

eLON\_SBE\_MOTOR\_CIRCUIT: Motor circuit

eLON\_SBE\_FUSE: Fuse

eLON\_SBE\_REFERENCE\_LOST: Reference lost

eLON\_SBE\_HOST\_COMM: Host communication

eLON\_SBE\_VOLTAGE\_1: Voltage 1

eLON\_SBE\_VOLTAGE\_2: Voltage 2

eLON\_SBE\_CONTROLLER: Controller

### 7.3.54 E\_LON\_scene\_config\_t

Genutzt durch: SNVT\_scene\_cfg

```

TYPE E_LON_scene_config_t :
(
  eLON_SCF_NUL          := -1,

```

```

eLON_SCF_SAVE      := 0,
eLON_SCF_CLEAR    := 1,
eLON_SCF_REPORT   := 2,
eLON_SCF_SIZE     := 3,
eLON_SCF_FREE     := 4
)
END_TYPE

```

eLON\_SCF\_NUL: Invalid Value

eLON\_SCF\_SAVE: Overwrite this scene with new data

eLON\_SCF\_CLEAR: Delete this scene from the list

eLON\_SCF\_REPORT: Display this scene's data

eLON\_SCF\_SIZE: Report the number of programmed scenes

eLON\_SCF\_FREE: Report the number of free scene storage spaces

### 7.3.55 E\_LON\_scene\_t

Genutzt durch: SNVT\_scene

```

TYPE E_LON_scene_t :
(
eLON_SC_NUL          := -1,
eLON_SC_RECALL      := 0,
eLON_SC_LEARN       := 1,
eLON_SC_DISPLAY     := 2,
eLON_SC_GROUP_OFF   := 3,
eLON_SC_GROUP_ON    := 4,
eLON_SC_STATUS_OFF  := 5,
eLON_SC_STATUS_ON   := 6,
eLON_SC_STATUS_MIXED := 7,
eLON_SC_GROUP_STATUS := 8,
eLON_SC_FLICK       := 9,
eLON_SC_TIMEOUT     := 10,
eLON_SC_TIMEOUT_FLICK := 11,
eLON_SC_DELAYOFF    := 12,
eLON_SC_DELAYOFF_FLICK := 13,
eLON_SC_DELAYON     := 14,
eLON_SC_ENABLE_GROUP := 15,
eLON_SC_DISABLE_GROUP := 16,
eLON_SC_CLEANNON    := 17,
eLON_SC_CLEANOFF    := 18,
eLON_SC_WINK        := 19,
eLON_SC_RESET       := 20,
eLON_SC_MODE1       := 21,
eLON_SC_MODE2       := 22,
eLON_SC_MODE3       := 23
)
END_TYPE

```

eLON\_SC\_NUL: Invalid value

eLON\_SC\_RECALL: Recall a specified scene.

eLON\_SC\_LEARN: Store the current setting in the specified scene.

eLON\_SC\_DISPLAY: Display the current scene.

eLON\_SC\_GROUP\_OFF: Report current group is off.

eLON\_SC\_GROUP\_ON: Report current group is on.

eLON\_SC\_STATUS\_OFF: Report current status is off.

eLON\_SC\_STATUS\_ON: Report current status is on.

eLON\_SC\_STATUS\_MIXED: Report current status is mixed.

eLON\_SC\_GROUP\_STATUS: Get group status.

eLON\_SC\_FLICK: Toggle state off and then on.

- eLON\_SC\_TIMEOUT: Report a timeout occurred.
- eLON\_SC\_TIMEOUT\_FLICK: Report a timeout occurred for a flick warning.
- eLON\_SC\_DELAYOFF: Set the state to off after a delay.
- eLON\_SC\_DELAYOFF\_FLICK: Flick and then set the state to off after a delay.
- eLON\_SC\_DELAYON: Set the state to on after a delay.
- eLON\_SC\_ENABLE\_GROUP: Enable the current group.
- eLON\_SC\_DISABLE\_GROUP: Disable the current group.
- eLON\_SC\_CLEANON: Recall the cleaning scene.
- eLON\_SC\_CLEANOFF: Restore the previous scene.
- eLON\_SC\_WINK: Toggle to the opposite state and then restore the state.
- eLON\_SC\_RESET: Restore the factory default scene table.
- eLON\_SC\_MODE1: Manufacturer-specific mode 1.
- eLON\_SC\_MODE2: Manufacturer-specific mode 2.
- eLON\_SC\_MODE3: Manufacturer-specific mode 3.

### 7.3.56 E\_LON\_sec\_state\_t

Genutzt durch: SNVT\_sec\_state

```

TYPE E_LON_sec_state_t :
(
  eLON_SSE_NUL           := -1,
  eLON_SSE_OFF           := 0,
  eLON_SSE_ON            := 1,
  eLON_SSE_INHIBIT_RESET := 2,
  eLON_SSE_INHIBIT       := 3,
  eLON_SSE_WALK_TEST_OFF := 4,
  eLON_SSE_WALK_TEST_ON  := 5,
  eLON_SSE_TEST_MODE_OFF := 6,
  eLON_SSE_TEST_MODE_ON  := 7,
  eLON_SSE_POLL_STATUS   := 8,
  eLON_SSE_POLL_STATE    := 9,
  eLON_SSE_CONFIRM_ALARM_RESET := 10,
  eLON_SSE_CONFIRM_ALARM := 11,
  eLON_SSE_CONFIRM_TAMPER_RESET := 12,
  eLON_SSE_CONFIRM_TAMPER := 13,
  eLON_SSE_CONFIRM_MAINTENANCE := 14,
  eLON_SSE_CONFIRM_TROUBLE := 15,
  eLON_SSE_CONFIRM_FAULT := 16,
  eLON_SSE_CONFIRM_RECOVERED_SENSOR := 17,
  eLON_SSE_LOST_SENSOR := 18,
  eLON_SSE_CONFIRM_UNSUPPORTED := 19
)
END_TYPE

```

- eLON\_SSE\_NUL:
- eLON\_SSE\_OFF:
- eLON\_SSE\_ON:
- eLON\_SSE\_INHIBIT\_RESET:
- eLON\_SSE\_INHIBIT:
- eLON\_SSE\_WALK\_TEST\_OFF:
- eLON\_SSE\_WALK\_TEST\_ON:
- eLON\_SSE\_TEST\_MODE\_OFF:

eLON\_SSE\_TEST\_MODE\_ON:  
 eLON\_SSE\_POLL\_STATUS:  
 eLON\_SSE\_POLL\_STATE:  
 eLON\_SSE\_CONFIRM\_ALARM\_RESET:  
 eLON\_SSE\_CONFIRM\_ALARM:  
 eLON\_SSE\_CONFIRM\_TAMPER\_RESET:  
 eLON\_SSE\_CONFIRM\_TAMPER:  
 eLON\_SSE\_CONFIRM\_MAINTENANCE:  
 eLON\_SSE\_CONFIRM\_TROUBLE:  
 eLON\_SSE\_CONFIRM\_FAULT:  
 eLON\_SSE\_CONFIRM\_RECOVERED\_SENSOR:  
 eLON\_SSE\_LOST\_SENSOR:  
 eLON\_SSE\_CONFIRM\_UNSUPPORTED:

### 7.3.57 E\_LON\_sec\_status\_t

Genutzt durch: SNVT\_sec\_status

```

TYPE E_LON_sec_status_t :
(
    eLON_SSS_NUL                := -1,
    eLON_SSS_POWER_UP          := 0,
    eLON_SSS_ALARM_RESET       := 1,
    eLON_SSS_ALARM             := 2,
    eLON_SSS_TAMPER_RESET      := 3,
    eLON_SSS_TAMPER            := 4,
    eLON_SSS_MAINTENANCE       := 5,
    eLON_SSS_TROUBLE           := 6,
    eLON_SSS_FAULT             := 7,
    eLON_SSS_RECOVERED_SENSOR  := 8,
    eLON_SSS_LOST_SENSOR       := 9,
    eLON_SSS_POLL_ACTIVE       := 10,
    eLON_SSS_POLL_INACTIVE     := 11,
    eLON_SSS_POLL_TAMPER       := 12,
    eLON_SSS_POLL_ON           := 13,
    eLON_SSS_POLL_OFF          := 14,
    eLON_SSS_POLL_INHIBIT      := 15,
    eLON_SSS_POLL_TEST         := 16,
    eLON_SSS_CONFIRM_OFF       := 17,
    eLON_SSS_CONFIRM_ON        := 18,
    eLON_SSS_CONFIRM_INHIBIT_RESET := 19,
    eLON_SSS_CONFIRM_INHIBIT    := 20,
    eLON_SSS_CONFIRM_WALK_TEST_OFF := 21,
    eLON_SSS_CONFIRM_WALK_TEST_ON := 22,
    eLON_SSS_CONFIRM_TEST_MODE_OFF := 23,
    eLON_SSS_CONFIRM_TEST_MODE_ON := 24,
    eLON_SSS_CONFIRM_UNSUPPORTED := 25
)
END_TYPE
    
```

eLON\_SSS\_NUL:  
 eLON\_SSS\_POWER\_UP:  
 eLON\_SSS\_ALARM\_RESET:  
 eLON\_SSS\_ALARM:  
 eLON\_SSS\_TAMPER\_RESET:  
 eLON\_SSS\_TAMPER:

eLON\_SSS\_MAINTENANCE:  
 eLON\_SSS\_TROUBLE:  
 eLON\_SSS\_FAULT:  
 eLON\_SSS\_RECOVERED\_SENSOR:  
 eLON\_SSS\_LOST\_SENSOR:  
 eLON\_SSS\_POLL\_ACTIVE:  
 eLON\_SSS\_POLL\_INACTIVE:  
 eLON\_SSS\_POLL\_TAMPER:  
 eLON\_SSS\_POLL\_ON:  
 eLON\_SSS\_POLL\_OFF:  
 eLON\_SSS\_POLL\_INHIBIT:  
 eLON\_SSS\_POLL\_TEST:  
 eLON\_SSS\_CONFIRM\_OFF:  
 eLON\_SSS\_CONFIRM\_ON:  
 eLON\_SSS\_CONFIRM\_INHIBIT\_RESET:  
 eLON\_SSS\_CONFIRM\_INHIBIT:  
 eLON\_SSS\_CONFIRM\_WALK\_TEST\_OFF:  
 eLON\_SSS\_CONFIRM\_WALK\_TEST\_ON:  
 eLON\_SSS\_CONFIRM\_TEST\_MODE\_OFF:  
 eLON\_SSS\_CONFIRM\_TEST\_MODE\_ON:  
 eLON\_SSS\_CONFIRM\_UNSUPPORTED:

### 7.3.58 E\_LON\_setting\_t

Genutzt durch: SNVT\_setting

```

TYPE E_LON_setting_t :
(
  eLON_SET_NUL      := -1,
  eLON_SET_OFF      := 0,
  eLON_SET_ON       := 1,
  eLON_SET_DOWN     := 2,
  eLON_SET_UP       := 3,
  eLON_SET_STOP     := 4,
  eLON_SET_STATE    := 5
)
END_TYPE
  
```

eLON\_SET\_NUL: Invalid value

eLON\_SET\_OFF: Change state to off

eLON\_SET\_ON: Change state to on, restoring the last on setting

eLON\_SET\_DOWN: Decrease the setting by the offset supplied in the setting field

eLON\_SET\_UP: Increase the setting by the offset supplied in the setting field

eLON\_SET\_STOP: Stop any motion, for example for blinds

eLON\_SET\_STATE: Change the setting to the value specified

## 7.3.59 E\_LON\_switch\_state\_t

Genutzt durch: SNVT\_switch\_2

```

TYPE E_LON_switch_state_t :
(
  eLON_SW_NUL           := -1,
  eLON_SW_SET_OFF      := 0,
  eLON_SW_SET_ON       := 1,
  eLON_SW_REPORT_OFF   := 2,
  eLON_SW_REPORT_ON    := 3,
  eLON_SW_TOGGLE_STATE := 4,
  eLON_SW_SET_LEVEL    := 5,
  eLON_SW_INCREASE_LEVEL := 6,
  eLON_SW_DECREASE_LEVEL := 7,
  eLON_SW_RECALL_SCENE := 8,
  eLON_SW_STORE_SCENE  := 9,
  eLON_SW_LEARN_SCENE  := 10,
  eLON_SW_SET_OCCUPIED := 11,
  eLON_SW_SET_UNOCCUPIED := 12,
  eLON_SW_SET_MULTIPLIER := 13,
  eLON_SW_ENABLE_GROUP := 14,
  eLON_SW_DISABLE_GROUP := 15,
  eLON_SW_WINK         := 16,
  eLON_SW_RESET        := 17,
  eLON_SW_RESET_ENERGY_USAGE := 18,
  eLON_SW_RESET_RUNTIME := 19,
  eLON_SW_INCREASE_HUE := 20,
  eLON_SW_DECREASE_HUE := 21,
  eLON_SW_SET_BUTTON   := 22,
  eLON_SW_SET_FAN_UP   := 32,
  eLON_SW_SET_FAN_DOWN := 33,
  eLON_SW_TOGGLE_FAN_DIRECTION := 34,
  eLON_SW_INCREASE_FAN_LEVEL := 35,
  eLON_SW_DECREASE_FAN_LEVEL := 36,
  eLON_SW_SET_FAN_ON   := 37,
  eLON_SW_SET_FAN_OFF  := 38,
  eLON_SW_TOGGLE_FAN_STATE := 39,
  eLON_SW_MOVE_OPEN    := 48,
  eLON_SW_MOVE_CLOSED := 49,
  eLON_SW_SET_ANGLE    := 50,
  eLON_SW_ROTATE_OPEN  := 51,
  eLON_SW_ROTATE_CLOSED := 52,
  eLON_SW_STOP         := 53,
  eLON_SW_SET_STANDBY  := 54,
  eLON_SW_TOGGLE_STANDBY := 55,
  eLON_SW_SET_POSITION := 56,
  eLON_SW_REPORT_POSITION := 57,
  eLON_SW_REPORT_FAN_LEVEL := 58
)
END_TYPE

```

eLON\_SW\_NUL: Invalid value

eLON\_SW\_SET\_OFF: Set the state to off; ignored for blinds, drapes, shades, and fans

eLON\_SW\_SET\_ON: Set the state to on; ignored for blinds, drapes, shades, and fans

eLON\_SW\_REPORT\_OFF: Report that the state is off; output only; ignored for input

eLON\_SW\_REPORT\_ON: Report that the state is on; output only; ignored for input

eLON\_SW\_TOGGLE\_STATE: Toggle on-off state; same action as SW\_SET\_OFF if the on/off state was on, and SW\_SET\_ON if the on/off state was off; ignored for blinds, drapes, shades, and fans

eLON\_SW\_SET\_LEVEL: Set the level to the specified value; ignored for blinds, drapes, shades, and fans

eLON\_SW\_INCREASE\_LEVEL: Increase the level by the specified value; ignored for blinds, drapes, shades, and fans

eLON\_SW\_DECREASE\_LEVEL: Decrease the level by the specified amount; ignored for blinds, drapes, shades, and fans

eLON\_SW\_RECALL\_SCENE: Recall the state and level from the specified scene

eLON\_SW\_STORE\_SCENE: Store setting for the specified scene

eLON\_SW\_LEARN\_SCENE: Learn setting for the specified scene

eLON\_SW\_SET\_OCCUPIED: Set the occupancy state

eLON\_SW\_SET\_UNOCCUPIED: Clear the occupancy state

eLON\_SW\_SET\_MULTIPLIER: Set a multiplier for the level for 60 minutes; ignored for blinds, drapes, shades, and fans

eLON\_SW\_ENABLE\_GROUP: Enable a group; all groups are enabled by default

eLON\_SW\_DISABLE\_GROUP: Disable a group

eLON\_SW\_WINK: Blink state (toggle on-off state; pause; toggle on-off state again)

eLON\_SW\_RESET: Reset scene definitions, multiplier, occupancy state, group enable flags, and settings to factory defaults

eLON\_SW\_RESET\_ENERGY\_USAGE: Reset energy usage value to zero

eLON\_SW\_RESET\_RUNTIME: Reset runtime value to zero

eLON\_SW\_INCREASE\_HUE: Increase color hue

eLON\_SW\_DECREASE\_HUE: Decrease color hue

eLON\_SW\_SET\_BUTTON: Trigger the actions for pressing and releasing the button specified in the value field

eLON\_SW\_SET\_FAN\_UP: Set ceiling fan direction to up, with specified level

eLON\_SW\_SET\_FAN\_DOWN: Set ceiling fan direction to down, with specified level

eLON\_SW\_TOGGLE\_FAN\_DIRECTION: Toggle fan up-down direction

eLON\_SW\_INCREASE\_FAN\_LEVEL: Increase fan speed by the setting

eLON\_SW\_DECREASE\_FAN\_LEVEL: Decrease fan speed by the setting

eLON\_SW\_SET\_FAN\_ON: Set the fan state to on

eLON\_SW\_SET\_FAN\_OFF: Set the fan state to off

eLON\_SW\_TOGGLE\_FAN\_STATE: Toggle the fan on-off state

eLON\_SW\_MOVE\_OPEN: Move blinds, drapes, or shades open by the setting

eLON\_SW\_MOVE\_CLOSED: Move blinds, drapes, or shades closed by the setting

eLON\_SW\_SET\_ANGLE: Set the rotation angle of blinds to the setting

eLON\_SW\_ROTATE\_OPEN: Rotate blinds open by the setting

eLON\_SW\_ROTATE\_CLOSED: Rotate blinds closed by the setting

eLON\_SW\_STOP: Stop any motion of blinds, drapes, or shades

eLON\_SW\_SET\_STANDBY: Set Standby mode

eLON\_SW\_TOGGLE\_STANDBY: Toggle the standby state

eLON\_SW\_SET\_POSITION: Set blinds, drapes, or shades to the specified position; 100% is fully open, 0% is fully closed

eLON\_SW\_REPORT\_POSITION: Report the position of blinds, drapes, or shades output only; ignored for input

eLON\_SW\_REPORT\_FAN\_LEVEL: Report the fan speed in percent of full level output only; ignored for input

### 7.3.60 E\_LON\_telcom\_states\_t

Genutzt durch: SNVT\_telcom

```
TYPE E_LON_telcom_states_t :  
(  
  eLON_TEL_NUL      := -1,  
  eLON_TEL_NOTINUSE := 0,  
  eLON_TEL_OFFHOOK  := 1,  
  eLON_TEL_DIALING  := 2,  
  eLON_TEL_DIALCOMP := 3,  
  eLON_TEL_RINGBACK := 4,  
  eLON_TEL_INCOMING := 5,  
  eLON_TEL_RINGING  := 6,  
  eLON_TEL_ANSWERED := 7,  
  eLON_TEL_CONNECTED := 8,  
  eLON_TEL_TALKING  := 9,  
  eLON_TEL_HANGINGUP := 10,  
  eLON_TEL_HUNGUPX  := 11,  
  eLON_TEL_HOLD     := 12,  
  eLON_TEL_UNHOLD   := 13,  
  eLON_TEL_RELEASE  := 14,  
  eLON_TEL_FULLLDUP := 15,  
  eLON_TEL_BLOCKED  := 16,  
  eLON_TEL_CWAIT    := 17,  
  eLON_TEL_DESTBUSY := 18,  
  eLON_TEL_NETBUSY  := 19,  
  eLON_TEL_ERROR    := 20  
)  
END_TYPE
```

eLON\_TEL\_NUL: Invalid Value

eLON\_TEL\_NOTINUSE: "Null State (U0)" not in use

eLON\_TEL\_OFFHOOK: "Call Initiated (U1)"

eLON\_TEL\_DIALING: "Overlap Sending (U2)"

eLON\_TEL\_DIALCOMP: "Outgoing Call Proceeding (U3)"

eLON\_TEL\_RINGBACK: "Call Delivered (U4)" hearing ringback

eLON\_TEL\_INCOMING: "Call Present (U6)" incoming call has not yet started ringing (only on ISDN line)

eLON\_TEL\_RINGING: "Call Received (U7)" incoming call when the user has indicated alerting but has not yet answered

eLON\_TEL\_ANSWERED: "Connect Request (U8)" user has answered the call and is waiting to be awarded the call

eLON\_TEL\_CONNECTED:

eLON\_TEL\_TALKING: "Active (U10)" two parties are exchanging data

eLON\_TEL\_HANGINGUP: "Disconnect Request (U11)" user has hung up

eLON\_TEL\_HUNGUPX: "Disconnect Indication (U12)" the other side hung up

eLON\_TEL\_HOLD: "Suspend Request (U15)" user has requested the network suspend the call

eLON\_TEL\_UNHOLD: "Resume Request (U17)" resume a held call (usually go back to TEL\_TALKING)

eLON\_TEL\_RELEASE: "Release Request (U19)" user has requested the network to release

eLON\_TEL\_FULLLDUP: "Overlap Receiving (U25)" user has acknowledged the call and is prepared to receive additional

eLON\_TEL\_BLOCKED: connection with blocking, (call-waiting disabled)

eLON\_TEL\_CWAIT: call-waiting coming in

eLON\_TEL\_DESTBUSY: destination busy



eLON\_TEL\_NETBUSY: problem, network

eLON\_TEL\_ERROR: problem, non-network

### 7.3.61 E\_LON\_therm\_mode\_t

Genutzt durch: SNVT\_therm\_mode

```
TYPE E_LON_therm_mode_t :
(
  eLON_THERM_NUL      := -1,
  eLON_THERM_NO_CONTROL := 0,
  eLON_THERM_IN_OUT   := 1,
  eLON_THERM_MODULATING := 2
)
END_TYPE
```

eLON\_THERM\_NUL: Ungültiger Wert.

eLON\_THERM\_NO\_CONTROL: Thermostat deaktiviert.

eLON\_THERM\_IN\_OUT: Cut in/out control.

eLON\_THERM\_MODULATING: Stetige Regelung.

### 7.3.62 E\_LON\_tilt\_dir\_t

Genutzt durch: SNVT\_ptz

```
TYPE E_LON_tilt_dir_t :
(
  eLON_TILT_NUL      := -1,
  eLON_TILT_STOP     := 0,
  eLON_TILT_UP       := 1,
  eLON_TILT_DOWN     := 2
)
END_TYPE
```

eLON\_TILT\_NUL: Ungültiger Wert.

eLON\_TILT\_STOP: Kippen stoppen.

eLON\_TILT\_UP: Nach oben kippen.

eLON\_TILT\_DOWN: Nach unten kippen.

### 7.3.63 E\_LON\_unit\_temp\_t

Genutzt durch: SNVT\_pump\_sensor

```
TYPE E_LON_unit_temp_t :
(
  eLON_TEMP_NUL      := -1,
  eLON_TEMP_INACTIVE := 0,
  eLON_TEMP_AT_DESIRED := 1,
  eLON_TEMP_TOO_HOT  := 2,
  eLON_TEMP_TOO_COLD := 3
)
END_TYPE
```

eLON\_TEMP\_NUL: Der Status vom Gerät oder der Einheit ist unbekannt oder nicht zulässig (ungültiger Wert).

eLON\_TEMP\_INACTIVE: Das Gerät zur Temperaturerfassung ist präsent, aber derzeit nicht in Betrieb.

eLON\_TEMP\_AT\_DESIRED: Die Gerätetemperatur befindet sich im gewünschten Bereich.

eLON\_TEMP\_TOO\_HOT: Die Gerätetemperatur befindet sich über dem oberen Grenzwert.

eLON\_TEMP\_TOO\_COLD: Die Gerätetemperatur befindet sich unter dem unteren Grenzwert.

### 7.3.64 E\_LON\_valve\_mode\_t

Genutzt durch: SNVT\_valve\_mode

```

TYPE E_LON_valve_mode_t :
(
  eLON_VALVE_NUL      := -1,
  eLON_VALVE_NORMAL   := 0,
  eLON_VALVE_COOLING  := 1,
  eLON_VALVE_HEATING  := 2,
  eLON_VALVE_EMERGENCY := 3,
  eLON_VALVE_STROKE_ADP := 4,
  eLON_VALVE_STROKE_SYN := 5,
  eLON_VALVE_ERROR    := 6,
  eLON_VALVE_OVERRIDDEN := 7
)
END_TYPE
    
```

eLON\_VALVE\_NUL: Ungültiger Wert.

eLON\_VALVE\_NORMAL: Ventil arbeitet als normales Ventil.

eLON\_VALVE\_COOLING: Ventil arbeitet nur Kühlventil.

eLON\_VALVE\_HEATING: Ventil arbeitet nur als Heizventil.

eLON\_VALVE\_EMERGENCY: Ventil arbeitet im Notbetrieb.

eLON\_VALVE\_STROKE\_ADP: Ventil passt Hub und Endlagen an.

eLON\_VALVE\_STROKE\_SYN: Ventil synchronisiert seine Position.

eLON\_VALVE\_ERROR: Ventil befindet sich im Fehlerzustand.

eLON\_VALVE\_OVERRIDDEN: Ventil wird übersteuert.

### 7.3.65 E\_LON\_zoom\_t

Genutzt durch: SNVT\_ptz

```

TYPE E_LON_zoom_t :
(
  eLON_ZOOM_NUL      := -1,
  eLON_ZOOM_STOP     := 0,
  eLON_ZOOM_TELE     := 1,
  eLON_ZOOM_WIDE     := 2
)
END_TYPE
    
```

eLON\_ZOOM\_NUL: Ungültiger Wert.

eLON\_ZOOM\_STOP: Stopt zoomen.

eLON\_ZOOM\_TELE: Teleobjektiv-Zoom / zoom in.

eLON\_ZOOM\_WIDE: Weitwinkel-Zoom / zoom out.

### 7.3.66 AuxiliaryStructure

#### SNVT\_chlr\_status

| Datentypen                | Beschreibung                    |
|---------------------------|---------------------------------|
| ST_LON_chlr_state [► 538] | Genutzt durch: SNVT_chlr_status |

#### SNVT\_clothes\_w\_c

| Datentypen            | Beschreibung                    |
|-----------------------|---------------------------------|
| ST_LON_action [► 538] | Genutzt durch: SNVT_clothes_w_c |

| Datentypen                              | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_dry [▶ 539]</a>      | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_duration [▶ 539]</a> | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_function [▶ 539]</a> | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_rinse [▶ 539]</a>    | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_spin [▶ 540]</a>     | Genutzt durch: SNVT_clothes_w_c |
| <a href="#">ST_LON_wash [▶ 540]</a>     | Genutzt durch: SNVT_clothes_w_c |

### SNVT\_clothes\_w\_s

| Datentypen                           | Beschreibung                    |
|--------------------------------------|---------------------------------|
| <a href="#">ST_LON_alarm [▶ 540]</a> | Genutzt durch: SNVT_clothes_w_s |

### SNVT\_color\_2

| Datentypen                                     | Beschreibung                |
|--|-----------------------------|
| <a href="#">ST_LON_CIE1931_lumen [▶ 542]</a>   | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_CIE1931_percent [▶ 542]</a> | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_color_value [▶ 543]</a>     | Genutzt durch: SNVT_color_2 |
| <a href="#">ST_LON_RGB [▶ 543]</a>             | Genutzt durch: SNVT_color_2 |

### SNVT\_ctrl\_resp

| Datentypen                            | Beschreibung                  |
|---------------------------------------|-------------------------------|
| <a href="#">ST_LON_range [▶ 543]</a>  | Genutzt durch: SNVT_ctrl_resp |
| <a href="#">ST_LON_sender [▶ 544]</a> | Genutzt durch: SNVT_ctrl_resp |

### SNVT\_dev\_fault

| Datentypen                                | Beschreibung                  |
|---|-------------------------------|
| <a href="#">ST_LON_Dev_type1 [▶ 544]</a>  | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_pump_ctrl1 [▶ 544]</a> | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_valve_pos1 [▶ 545]</a> | Genutzt durch: SNVT_dev_fault |

### SNVT\_dev\_maint

| Datentypen                                | Beschreibung                  |
|---|-------------------------------|
| <a href="#">ST_LON_Dev_type2 [▶ 546]</a>  | Genutzt durch: SNVT_dev_maint |
| <a href="#">ST_LON_pump_ctrl2 [▶ 546]</a> | Genutzt durch: SNVT_dev_maint |
| <a href="#">ST_LON_valve_pos2 [▶ 547]</a> | Genutzt durch: SNVT_dev_maint |

### SNVT\_dev\_status

| Datentypen                                | Beschreibung                   |
|---|--------------------------------|
| <a href="#">ST_LON_Dev_type3 [▶ 547]</a>  | Genutzt durch: SNVT_dev_status |
| <a href="#">ST_LON_pump_ctrl3 [▶ 548]</a> | Genutzt durch: SNVT_dev_status |
| <a href="#">ST_LON_valve_pos3 [▶ 549]</a> | Genutzt durch: SNVT_dev_status |

**SNVT\_ex\_control**

| Datentypen                         | Beschreibung                   |
|------------------------------------|--------------------------------|
| ST_LON_Control_device_addr [▶ 549] | Genutzt durch: SNVT_ex_control |

**SNVT\_file\_req**

| Datentypen                  | Beschreibung                 |
|-----------------------------|------------------------------|
| ST_LON_addrt [▶ 550]        | Genutzt durch: SNVT_file_req |
| ST_LON_dest_address [▶ 550] | Genutzt durch: SNVT_file_req |
| ST_LON_gp [▶ 550]           | Genutzt durch: SNVT_file_req |
| ST_LON_sn [▶ 551]           | Genutzt durch: SNVT_file_req |

**SNVT\_file\_status**

| Datentypen                | Beschreibung  |
|---------------------------|---|
| ST_LON_address [▶ 551]    | Genutzt durch: FB_Write_Address_Table / FB_Read_Address_Table |
| ST_LON_adr [▶ 551]        | Genutzt durch: SNVT_file_status                               |
| ST_LON_descriptor [▶ 552] | Genutzt durch: SNVT_file_status                               |

**SNVT\_lamp\_status**

| Datentypen                    | Beschreibung                    |
|-------------------------------|---------------------------------|
| ST_LON_Alarm_actual [▶ 552]   | Genutzt durch: SNVT_lamp_status |
| ST_LON_alarm_previous [▶ 554] | Genutzt durch: SNVT_lamp_status |

**SNVT\_pos\_ctrl**

| Datentypen            | Beschreibung                 |
|-----------------------|------------------------------|
| ST_LON_abspos [▶ 556] | Genutzt durch: SNVT_pos_ctrl |
| ST_LON_Value [▶ 556]  | Genutzt durch: SNVT_pos_ctrl |

**SNVT\_rac\_ctrl**

| Datentypen               | Beschreibung                                |
|--------------------------|---|
| ST_LON_addr_dest [▶ 556] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| ST_LON_addr_init [▶ 557] | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| ST_LON_addr_talk [▶ 557] | Genutzt durch: SNVT_rac_ctrl                |
| ST_LON_p2m [▶ 557]       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| ST_LON_p2p [▶ 557]       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |

**SNVT\_rac\_req**

| Datentypen                       | Beschreibung |
|----------------------------------|--------------|
| ST_LON_rac_req_addr_dest [▶ 558] |              |
| ST_LON_rac_req_addr_init [▶ 558] |              |

**SNVT\_switch\_2**

| Datentypen             | Beschreibung                 |
|------------------------|------------------------------|
| ST_LON_setting [▶ 559] | Genutzt durch: SNVT_switch_2 |

### SNVT\_time\_zone

| Datentypen                                   | Beschreibung                  |
|--|-------------------------------|
| ST_LON_end_DST [ <a href="#">▶ 559</a> ]     | Genutzt durch: SNVT_time_zone |
| ST_LON_M_end_DST [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |
| ST_LON_M_start_DST [ <a href="#">▶ 560</a> ] | Genutzt durch: SNVT_time_zone |
| ST_LON_start_DST [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |

### 7.3.66.1 SNVT\_chlr\_status

| Datentypen                                  | Beschreibung                    |
|---|---------------------------------|
| ST_LON_chlr_state [ <a href="#">▶ 538</a> ] | Genutzt durch: SNVT_chlr_status |

#### 7.3.66.1.1 ST\_LON\_chlr\_state

Genutzt durch: SNVT\_chlr\_status

```

TYPE ST_LON_chlr_state :
STRUCT
  bIn_alarm      : BOOL;
  bRun_enabled   : BOOL;
  bLocal         : BOOL;
  bLimited       : BOOL;
  bChw_flow      : BOOL;
  bCondw_flow    : BOOL;
END_STRUCT
END_TYPE
    
```

**bIn\_alarm:** Alarm flag (boolean).

**bRun\_enabled:** Run-enabled flag (boolean).

**bLocal:** Locally-controlled flag (boolean).

**bLimited:** Limited-condition flag (boolean). Conditions may exist that prevent reaching the setpoint

**bChw\_flow:** Chiller-water-flow flag (boolean).

**bCondw\_flow:** Condenser-water-flow flag (boolean).

### 7.3.66.2 SNVT\_clothes\_w\_c

| Datentypen                                | Beschreibung                    |
|---|---------------------------------|
| ST_LON_action [ <a href="#">▶ 538</a> ]   | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_dry [ <a href="#">▶ 539</a> ]      | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_duration [ <a href="#">▶ 539</a> ] | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_function [ <a href="#">▶ 539</a> ] | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_rinse [ <a href="#">▶ 539</a> ]    | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_spin [ <a href="#">▶ 540</a> ]     | Genutzt durch: SNVT_clothes_w_c |
| ST_LON_wash [ <a href="#">▶ 540</a> ]     | Genutzt durch: SNVT_clothes_w_c |

#### 7.3.66.2.1 ST\_LON\_action

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_action :
STRUCT
  bPower_on      : BOOL;
  bRun_mode      : BOOL;
    
```

```

byRsrvd2_7 : BYTE;
END_STRUCT
END_TYPE

```

**bPower\_on:**

**bRun\_mode:**

**byRsrvd2\_7:**

### 7.3.66.2.2 ST\_LON\_dry

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_dry :
STRUCT
  byTemp      : BYTE;
  stDuration  : ST_LON_Duration;
END_STRUCT
END_TYPE

```

**byTemp:** Min: 0 / Max: 1.

**stDuration:** (see [ST\\_LON\\_Duration](#) [[▶ 539](#)])

### 7.3.66.2.3 ST\_LON\_duration

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_duration :
STRUCT
  byTime      : BYTE;
  eDryness    : E_LON_discrete_levels_t;
END_STRUCT
END_TYPE

```

**byTime:** Min: 0 / Max: 255

**eDryness:** see [E\\_LON\\_discrete\\_levels\\_t](#) [[▶ 505](#)]

### 7.3.66.2.4 ST\_LON\_function

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_function :
STRUCT
  eProgram    : E_LON_appl_cwp_t;
  stWash      : ST_LON_wash;
  stRinse     : ST_LON_rinse;
  stSpin      : ST_LON_spin;
  stDry       : ST_LON_dry;
END_STRUCT
END_TYPE

```

**eProgram:** (see [E\\_LON\\_appl\\_cwp\\_t](#) [[▶ 495](#)])

**stWash:** (see [ST\\_LON\\_wash](#) [[▶ 540](#)])

**stRinse:** (see [ST\\_LON\\_rinse](#) [[▶ 539](#)])

**stSpin:** (see [ST\\_LON\\_spin](#) [[▶ 540](#)])

**stDry:** (see [ST\\_LON\\_dry](#) [[▶ 539](#)])

### 7.3.66.2.5 ST\_LON\_rinse

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_rinse :
STRUCT
  byTemp      : BYTE;
  byRepeat    : BYTE;
  eOption     : E_LON_appl_rin_t;
END_STRUCT
END_TYPE
    
```

**byTemp:** Min: 0 / Max: 255

**byRepeat:** Min: 0 / Max: 9

**eOption:** see [E\\_LON\\_appl\\_rin\\_t \[► 497\]](#)

### 7.3.66.2.6 ST\_LON\_spin

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_spin :
STRUCT
  uiSpeed     : UINT;
  byTime      : BYTE;
  eHold       : E_LON_boolean_t;
END_STRUCT
END_TYPE
    
```

**uiSpeed:** Min: 0 / Max: 65535

**byTime:** Min: 0 / Max: 255

**eHold:** see [E\\_LON\\_boolean\\_t \[► 497\]](#)

### 7.3.66.2.7 ST\_LON\_wash

Genutzt durch: SNVT\_clothes\_w\_c

```

TYPE ST_LON_wash :
STRUCT
  eLoad_level : E_LON_discrete_levels_t;
  byTemp      : BYTE;
  byTime      : BYTE;
  ePrewash    : E_LON_boolean_t;
END_STRUCT
END_TYPE
    
```

**eLoad\_level:** see [E\\_LON\\_discrete\\_levels\\_t \[► 505\]](#)

**byTemp:** Min: 0 / Max: 255

**byTime:** Min: 0 / Max: 255

**ePrewash:** see [E\\_LON\\_boolean\\_t \[► 497\]](#)

## 7.3.66.3 SNVT\_clothes\_w\_s

| Datentypen                           | Beschreibung                    |
|--------------------------------------|---------------------------------|
| <a href="#">ST_LON_alarm [► 540]</a> | Genutzt durch: SNVT_clothes_w_s |

### 7.3.66.3.1 ST\_LON\_alarm

Genutzt durch: SNVT\_clothes\_w\_s

```

TYPE ST_LON_alarm :
STRUCT
  bAlarm_reset      : BOOL;
  bWar_water_supply : BOOL;
  bWar_drain_slow   : BOOL;
  bWar_door_open    : BOOL;
  bWar_load_unbalanced : BOOL;
  bWar_filter_cleaning : BOOL;
END_STRUCT
    
```

```
bWar_hoses_reversed : BOOL;
bWar_voltage_low    : BOOL;
bWar_power_failure  : BOOL;
bWar_drain_open     : BOOL;
bWar_execute_fail   : BOOL;
bWar_door_locked    : BOOL;
bWar_service        : BOOL;
bWar_rsrvd5         : BOOL;
bWar_rsrvd6         : BOOL;
bWar_rsrvd7         : BOOL;
bErr_motor_stall    : BOOL;
bErr_water_temp     : BOOL;
bErr_pressure       : BOOL;
bErr_overflow       : BOOL;
bErr_water_heat     : BOOL;
bErr_water_leak     : BOOL;
bErr_motor_speed    : BOOL;
bErr_wash_thermistor : BOOL;
bErr_dry_thermistor : BOOL;
bErr_dry_overheat   : BOOL;
bErr_dry_heating    : BOOL;
bErr_dry_fan        : BOOL;
bErr_rsrvd4         : BOOL;
bErr_rsrvd5         : BOOL;
bErr_rsrvd6         : BOOL;
bErr_rsrvd7         : BOOL;
byErr_rsrvd0_7      : BYTE;
byErr_manuf_code    : BYTE;
END_STRUCT
END_TYPE
```

**bAlarm\_reset:**

**bWar\_water\_supply:**

**bWar\_drain\_slow:**

**bWar\_door\_open:**

**bWar\_load\_unbalanced:**

**bWar\_filter\_cleaning:**

**bWar\_hoses\_reversed:**

**bWar\_voltage\_low:**

**bWar\_power\_failure:**

**bWar\_drain\_open:**

**bWar\_execute\_fail:**

**bWar\_door\_locked:**

**bWar\_service:**

**bWar\_rsrvd5:**

**bWar\_rsrvd6:**

**bWar\_rsrvd7:**

**bErr\_motor\_stall:**

**bErr\_water\_temp:**

**bErr\_pressure:**

**bErr\_overflow:**

**bErr\_water\_heat:**

**bErr\_water\_leak:**



- bErr\_motor\_speed:**
- bErr\_wash\_thermistor:**
- bErr\_dry\_thermistor:**
- bErr\_dry\_overheat:**
- bErr\_dry\_heating:**
- bErr\_dry\_fan:**
- bErr\_rsrvd4:**
- bErr\_rsrvd5:**
- bErr\_rsrvd6:**
- bErr\_rsrvd7:**
- byErr\_rsrvd0\_7:**
- byErr\_manuf\_code:** Min: 0 / Max: 255

### 7.3.66.4 SNVT\_color\_2

| Datentypen                     | Beschreibung                |
|--------------------------------|-----------------------------|
| ST_LON_CIE1931_lumen [▶ 542]   | Genutzt durch: SNVT_color_2 |
| ST_LON_CIE1931_percent [▶ 542] | Genutzt durch: SNVT_color_2 |
| ST_LON_color_value [▶ 543]     | Genutzt durch: SNVT_color_2 |
| ST_LON_RGB [▶ 543]             | Genutzt durch: SNVT_color_2 |

#### 7.3.66.4.1 ST\_LON\_CIE1931\_lumen

Genutzt durch: SNVT\_color\_2

```

TYPE ST_LON_CIE1931_lumen :
STRUCT
  rX      : REAL;
  rY      : REAL;
  udiAbsolute_Y : UDINT;
END_STRUCT
END_TYPE
    
```

**rX:** Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value

**rY:** Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value

**udiAbsolute\_Y:** Min: 0 / Max: 6553400 / Invalid: 6553500 / Absolute luminance (lumen). Absolute luminance

#### 7.3.66.4.2 ST\_LON\_CIE1931\_percent

Genutzt durch: SNVT\_color\_2

```

TYPE ST_LON_CIE1931_percent :
STRUCT
  rX      : REAL;
  rY      : REAL;
  rPercent_Y : REAL;
END_STRUCT
END_TYPE
    
```

**rX:** Min: 0.0 / Max: 0.740 / Invalid: 1.275 / CIE 1931 x value (CIE 1931 color space coordinate). CIE 1931 x-axis color value

**rY:** Min: 0.0 / Max: 0.840 / Invalid: 1.275 / CIE 1931 y value (CIE 1931 color space coordinate). CIE 1931 y-axis color value

**rPercent\_Y:** Min: 0.0 / Max: 100.0 / Invalid: 655.35 / Luminance (% of full level). Y output in percent of maximum lumen output of the lamp

### 7.3.66.4.3 ST\_LON\_color\_value

Genutzt durch: SNVT\_color\_2

```
TYPE ST_LON_color_value :
STRUCT
  stCIE1931_lumen      : ST_LON_CIE1931_lumen;
  stCIE1931_percent   : ST_LON_CIE1931_percent;
  stRGB                : ST_LON_RGB;
  uiColor_temperature : UINT;
END_STRUCT
END_TYPE
```

**stCIE1931\_lumen:** CIE 1931 color space with lumen. CIE 1931 color space with Y output in lumen (see [ST\\_LON\\_CIE1931\\_lumen](#) [► 542]).

**stCIE1931\_percent:** CIE 1931 color space with percent. CIE 1931 color space with Y output in percent of maximum lumen output of the lamp (see [ST\\_LON\\_CIE1931\\_percent](#) [► 542]).

**stRGB:** RGB color value (see [ST\\_LON\\_RGB](#) [► 543]).

**uiColor\_temperature:** Min: 2800 / Max: 7500 / Invalid: 12750 / Color temperature (degrees Kelvin).

### 7.3.66.4.4 ST\_LON\_RGB

Genutzt durch: SNVT\_color\_2

```
TYPE ST_LON_RGB :
STRUCT
  byRed   : BYTE;
  byGreen : BYTE;
  byBlue  : BYTE;
END_STRUCT
END_TYPE
```

**byRed:** Min: 0 / Max: 250 / Red component. Red component for RGB color

**byGreen:** Min: 0 / Max: 250 / Green component. Green component for RGB color

**byBlue:** Min: 0 / Max: 250 / Blue component. Blue component for RGB color

### 7.3.66.5 SNVT\_ctrl\_resp

| Datentypen                            | Beschreibung                  |
|---------------------------------------|-------------------------------|
| <a href="#">ST_LON_range</a> [► 543]  | Genutzt durch: SNVT_ctrl_resp |
| <a href="#">ST_LON_sender</a> [► 544] | Genutzt durch: SNVT_ctrl_resp |

#### 7.3.66.5.1 ST\_LON\_range

Genutzt durch: SNVT\_ctrl\_resp

```
TYPE ST_LON_range :
STRUCT
  uiLower : UINT;
  uiUpper : UINT;
END_STRUCT
END_TYPE
```

**uiLower:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender range lower ID (ID number).

**uiUpper:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender range upper ID (ID number).

### 7.3.66.5.2 ST\_LON\_sender

Genutzt durch: SNVT\_ctrl\_resp

```
TYPE ST_LON_sender :
STRUCT
  uiID      : UINT;
  stRange   : ST_LON_range;
END_STRUCT
END_TYPE
```

**uiID:** Min: 1 / Max: 65535 / Invalid: 65535 / Sender ID (ID number).

**stRange:** Sender ID range (lower, upper) (see [ST\\_LON\\_range](#) [▶ 543]).

### 7.3.66.6 SNVT\_dev\_fault

| Datentypen                                | Beschreibung                  |
|---|-------------------------------|
| <a href="#">ST_LON_Dev_type1</a> [▶ 544]  | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_pump_ctrl1</a> [▶ 544] | Genutzt durch: SNVT_dev_fault |
| <a href="#">ST_LON_valve_pos1</a> [▶ 545] | Genutzt durch: SNVT_dev_fault |

#### 7.3.66.6.1 ST\_LON\_Dev\_type1

Genutzt durch: SNVT\_dev\_fault

```
TYPE ST_LON_Dev_type1 :
STRUCT
  stPump_ctrl   : ST_LON_pump_ctrl1;
  stValvePos    : ST_LON_valve_pos1;
END_STRUCT
END_TYPE
```

**stPump\_ctrl:** Pump controller device fault information (see [ST\\_LON\\_pump\\_ctrl1](#) [▶ 544]).

**stValvePos:** Valve positioner device fault information (see [ST\\_LON\\_valve\\_pos1](#) [▶ 545]).

#### 7.3.66.6.2 ST\_LON\_pump\_ctrl1

Genutzt durch: SNVT\_dev\_fault

```
TYPE ST_LON_pump_ctrl1 :
STRUCT
  bSf_voltage_low      : BOOL;
  bSf_voltage_high    : BOOL;
  bSf_phase            : BOOL;
  bSf_no_fluid        : BOOL;
  bSf_press_low       : BOOL;
  bSf_press_high      : BOOL;
  bSf_general_fault    : BOOL;
  bSf_reserved1_7     : BOOL;
  bDf_motor_temp      : BOOL;
  bDf_motor_failure    : BOOL;
  bDf_pump_blocked    : BOOL;
  bDf_elect_temp      : BOOL;
  bDf_elect_failure_nf : BOOL;
  bDf_elect_failure    : BOOL;
  bDf_sensor_failure   : BOOL;
  bDf_general_fault    : BOOL;
  byReserved3_0_7     : BYTE;
END_STRUCT
END_TYPE
```

**bSf\_voltage\_low:** Supply fault - low voltage (boolean). Supply voltage is too low.

**bSf\_voltage\_high:** Supply fault - high voltage (boolean). Supply voltage is too high.

**bSf\_phase:** Supply fault - power phase (boolean). Supply power is missing phase.

**bSf\_no\_fluid:** Supply fault - no fluid (boolean). There is no fluid in the pump.

**bSf\_press\_low:** Supply fault - low pressure (boolean). System pressure is too low

**bSf\_press\_high:** Supply fault - high pressure (boolean). System pressure is too high

**bSf\_general\_fault:** General supply fault.

**bSf\_reserved1\_7:**

**bDf\_motor\_temp:** Device fault - motor temperature (boolean). Motor temperature is too high

**bDf\_motor\_failure:** Device fault - motor fatal failure (boolean). Motor has encountered a fatal failure

**bDf\_pump\_blocked:** Device fault - pump blocked (boolean). Pump is presently blocked

**bDf\_elect\_temp:** Device fault - electronics temperature (boolean). Temperature of the electronic circuitry is too high

**bDf\_elect\_failure\_nf:** Device fault - electronics failure (boolean). Electronic circuitry has encountered a non-fatal failure

**bDf\_elect\_failure:** Device fault - electronics fatal failure (boolean). Electronic circuitry has encountered a fatal failure

**bDf\_sensor\_failure:** Device fault - sensor failure (boolean). Sensor has failed on the device

**bDf\_general\_fault:** General device fault.

**byReserved3\_0\_7:**

### 7.3.66.6.3 ST\_LON\_valve\_pos1

Genutzt durch: SNVT\_dev\_fault

```

TYPE ST_LON_valve_pos1 :
STRUCT
  bDf_valve_blocked          : BOOL;
  bDf_blocked_direction_open : BOOL;
  bDf_blocked_direction_close : BOOL;
  bDf_position_error         : BOOL;
  bDf_stroke_Out_of_range    : BOOL;
  bDf_initialization         : BOOL;
  bDf_vibration_cavitation    : BOOL;
  bDf_ed_too_high            : BOOL;
  byReserved1_0_2            : BYTE;
  bEe_oscillating             : BOOL;
  bEe_valve_too_large         : BOOL;
  bEe_valve_too_small         : BOOL;
  byReserved2_6_7            : BYTE;
  bReserved3_0_7              : BOOL;
  bSf_voltage_Out_of_range    : BOOL;
  bSf_electronic_high_temp    : BOOL;
  bSf_frictional_resistance   : BOOL;
  byReserved4_4_6            : BYTE;
  bGeneral_fault              : BOOL;
END_STRUCT
END_TYPE

```

**bDf\_valve\_blocked:** Device fault - valve blocked. The valve is presently blocked.

**bDf\_blocked\_direction\_open:** Device fault - blocked direction open. The device is blocked while attempting to open.

**bDf\_blocked\_direction\_close:** Device fault - blocked direction close. The device is blocked while attempting to close

**bDf\_position\_error:** Device fault - position error. The valve position is not correct

**bDf\_stroke\_Out\_of\_range:** Device fault - stroke out of range. The valve stroke is out of operating range

**bDf\_initialization:** Device fault - initialization error. There was an error during initialization of the device

**bDf\_vibration\_cavitation:** Device fault - vibration / cavitation. There are excessive vibrations or cavitations detected

**bDf\_ed\_too\_high:** Device fault - ED too high. The ED is too high

**byReserved1\_0\_2:** This field is reserved.. This field is reserved.

**bEe\_oscillating:** Engineering error - oscillating. There is an oscillating error

**bEe\_valve\_too\_large:** Engineering error - valve too big. The valve size is too large

**bEe\_valve\_too\_small:** Engineering error - valve too small. The valve size is too small

**byReserved2\_6\_7:** This field is reserved.. This field is reserved.

**bReserved3\_0\_7:** This field is reserved.. This field is reserved.

**bSf\_voltage\_Out\_of\_range:** Supply fault - voltage out of range. The voltage is out of the specified acceptable range

**bSf\_electronic\_high\_temp:** Supply fault - electronics temperature. The temperature of the electronics is too high

**bSf\_frictional\_resistance:** Supply fault - frictional resistance. Resistance due to friction is detected

**byReserved4\_4\_6:** This field is reserved.. This field is reserved.

**bGeneral\_fault:** General Fault. A General Fault has occurred. Please consult the documentation or contact the valve-controller manufacturer.

### 7.3.66.7 SNVT\_dev\_maint

| Datentypen                                  | Beschreibung                  |
|---|-------------------------------|
| ST_LON_Dev_type2 [ <a href="#">▶ 546</a> ]  | Genutzt durch: SNVT_dev_maint |
| ST_LON_pump_ctrl2 [ <a href="#">▶ 546</a> ] | Genutzt durch: SNVT_dev_maint |
| ST_LON_valve_pos2 [ <a href="#">▶ 547</a> ] | Genutzt durch: SNVT_dev_maint |

#### 7.3.66.7.1 ST\_LON\_Dev\_type2

Genutzt durch: SNVT\_dev\_maint

```

TYPE ST_LON_Dev_type2 :
STRUCT
  stPump_ctrl1 : ST_LON_pump_ctrl2;
  stValvePos   : ST_LON_valve_pos2;
END_STRUCT
END_TYPE
    
```

**stPump\_ctrl1:** Pump controller device maintenance state (see [ST\\_LON\\_pump\\_ctrl2](#) [[▶ 546](#)]).

**stValvePos:** Valve positioner device maintenance information (see [ST\\_LON\\_valve\\_pos2](#) [[▶ 547](#)]).

#### 7.3.66.7.2 ST\_LON\_pump\_ctrl2

Genutzt durch: SNVT\_dev\_maint

```

TYPE ST_LON_pump_ctrl2 :
STRUCT
  bService_required : BOOL;
  bBearings_change  : BOOL;
  bBearings_lubricate : BOOL;
  bShaftseal_change : BOOL;
  byReserved1_4_7    : BYTE;
  byReserved2_0_7    : BYTE;
  byReserved3_0_7    : BYTE;
END_STRUCT
END_TYPE
    
```

**bService\_required:** Service required (boolean) . Service/maintenance is required

**bBearings\_change:** Change bearings (boolean) . Bearings need to be replaced

**bBearings\_lubricate:** Lubricate bearings (boolean) . Bearings need to be greased

**bShaftseal\_change:** Change shaft seal (boolean) . Seal on the shaft needs to be replaced

**byReserved1\_4\_7:** Reserve

**byReserved2\_0\_7:** Reserve

**byReserved3\_0\_7:** Reserve

### 7.3.66.7.3 ST\_LON\_valve\_pos2

Genutzt durch: SNVT\_dev\_maint

```
TYPE ST_LON_valve_pos2 :
STRUCT
  bMotor_maint      : BOOL;
  bPacking_change   : BOOL;
  bElectronics_check : BOOL;
  bPositioning_check : BOOL;
  bLubrication_check : BOOL;
  bReturn_check     : BOOL;
  battery_check     : BOOL;
  bReserved1_7      : BOOL;
  byReserved2_0_7   : BYTE;
  byReserved3_0_6   : BYTE;
  bGeneral_maint    : BOOL;
END_STRUCT
END_TYPE
```

**bMotor\_maint:** Motor Maintenance. The motor requires servicing

**bPacking\_change:** Packing Change. The packing needs to be controlled or changed

**bElectronics\_check:** Check Electronics. The electronics need to be checked (temperature too high)

**bPositioning\_check:** Check Position. The positioning needs to be checked (mechanical or electronic)

**bLubrication\_check:** Check Lubrication. The lubrication need to be checked

**bReturn\_check:** Check Spring-Return Function. The spring-return function needs to be checked

**battery\_check:** Check battery. The battery needs to be checked

**bReserved1\_7:** This field is reserved.. This field is reserved.

**byReserved2\_0\_7:** This field is reserved.. This field is reserved.

**byReserved3\_0\_6:** This field is reserved.. This field is reserved.

**bGeneral\_maint:** General Maintenance. General Maintenance needs to be performed. Please consult the documentation or your Maintenance Department.

### 7.3.66.8 SNVT\_dev\_status

| Datentypen                                  | Beschreibung                   |
|---|--------------------------------|
| ST_LON_Dev_type3 [ <a href="#">▶ 547</a> ]  | Genutzt durch: SNVT_dev_status |
| ST_LON_pump_ctrl3 [ <a href="#">▶ 548</a> ] | Genutzt durch: SNVT_dev_status |
| ST_LON_valve_pos3 [ <a href="#">▶ 549</a> ] | Genutzt durch: SNVT_dev_status |

#### 7.3.66.8.1 ST\_LON\_Dev\_type3

Genutzt durch: SNVT\_dev\_status

```
TYPE ST_LON_Dev_type3 :
STRUCT
  stPump_ctrl : ST_LON_pump_ctrl3;
```

```

    stValvePos      : ST_LON_valve_pos3;
END_STRUCT
END_TYPE

```

**stPump\_ctrl:** Pump controller device status (see [ST\\_LON\\_pump\\_ctrl3](#) [► 548]).

**stValvePos:** Valve positioner device status (see [ST\\_LON\\_valve\\_pos3](#) [► 549]).

### 7.3.66.8.2 ST\_LON\_pump\_ctrl3

Genutzt durch: SNVT\_dev\_status

```

TYPE ST_LON_pump_ctrl3 :
STRUCT
    bDevice_fault      : BOOL;
    bSupply_fault      : BOOL;
    bReserved1_2       : BOOL;
    bSpeed_low         : BOOL;
    bSpeed_high        : BOOL;
    bReserved1_5       : BOOL;
    bSetpt_Out_of_range : BOOL;
    bReserved1_7       : BOOL;
    bLocal_control     : BOOL;
    bReserved2_1       : BOOL;
    bRunning           : BOOL;
    bReserved2_3       : BOOL;
    bRemote_press      : BOOL;
    bRemote_flow       : BOOL;
    bRemote_temp       : BOOL;
    bReserved2_7       : BOOL;
    byReserved3_0_7    : BYTE;
END_STRUCT
END_TYPE

```

**bDevice\_fault:** Pump controller fault (boolean). See SNVT\_pump\_fault network variable declaration on device

**bSupply\_fault:** Supply fault (boolean). No electrical power, no fluid in pump, etc. See SNVT\_pump\_fault network variable declaration on device.

**bReserved1\_2:**

**bSpeed\_low:** Low-speed limit of pump (boolean). Pump is running at the lowest possible speed, therefore the requested performance is not possible.

**bSpeed\_high:** High-speed limit of pump (boolean). Pump is running at the highest possible speed, therefore the requested performance is not possible.

**bReserved1\_5:**

**bSetpt\_Out\_of\_range:** Setpoint out of range (boolean). Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

**bReserved1\_7:**

**bLocal\_control:** Locally controlled pump (boolean). Pump is locally operated (hardware override)

**bReserved2\_1:**

**bRunning:** Running pump (boolean). Pump is presently running

**bReserved2\_3:**

**bRemote\_press:** Remote pressure sensor (boolean). Pump controller is using a remote pressure sensor

**bRemote\_flow:** Remote flow sensor (boolean). Pump controller is using a remote flow sensor

**bRemote\_temp:** Remote temperature sensor (boolean). Pump controller is using a remote temperature sensor

**bReserved2\_7:**

**byReserved3\_0\_7:**

### 7.3.66.8.3 ST\_LON\_valve\_pos3

Genutzt durch: SNVT\_dev\_status

```

TYPE ST_LON_valve_pos3 :
STRUCT
  bRunning          : BOOL;
  bAdapting         : BOOL;
  bInitializing     : BOOL;
  bLocal_control    : BOOL;
  bSetpt_Out_of_range : BOOL;
  bRemote_ctrl_signal : BOOL;
  byReserved1_6_7   : BYTE;
  bHw_emergency     : BOOL;
  bSw_emergency     : BOOL;
  byReserved2_2_7   : BYTE;
  byReserved3_0_7   : BYTE;
END_STRUCT
END_TYPE
    
```

**bRunning:** Valve Running. Valve is presently being positioned.

**bAdapting:** Adapting. Valve is presently adapting.

**bInitializing:** Initializing. Valve is presently initializing.

**bLocal\_control:** Local Control. The valve operation is being locally controlled.

**bSetpt\_Out\_of\_range:** Setpoint out of range. Chosen override setpoint value is lower than the manufacturer-defined low-setpoint limit or higher than the manufacturer-defined high-setpoint limit.

**bRemote\_ctrl\_signal:** Remote Control Signal. The remote-control signal is active.

**byReserved1\_6\_7:** This field is reserved.. This field is reserved.

**bHw\_emergency:** Hardware Emergency. The hardware-emergency state is active

**bSw\_emergency:** Software Emergency. The software-emergency state is active

**byReserved2\_2\_7:** This field is reserved.. This field is reserved.

**byReserved3\_0\_7:** This field is reserved.. This field is reserved.

### 7.3.66.9 SNVT\_ex\_control

| Datentypen   | Beschreibung                   |
|--|--------------------------------|
| ST_LON_Control_device_addr [ <a href="#">▶ 549</a> ] | Genutzt durch: SNVT_ex_control |

#### 7.3.66.9.1 ST\_LON\_Control\_device\_addr

Genutzt durch: SNVT\_ex\_control

```

TYPE ST_LON_Control_device_addr :
STRUCT
  arrDomain_id      : ARRAY [0..5] OF BYTE;
  byDomain_length   : BYTE;
  bySubnet          : BYTE;
  byNode            : BYTE;
END_STRUCT
END_TYPE
    
```

**arrDomain\_id:** Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID

**byDomain\_length:** Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.

**bySubnet:** Min: 1 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

**byNode:** Min: 1 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.



### 7.3.66.10 SNVT\_file\_req

| Datentypen                         | Beschreibung                 |
|------------------------------------|------------------------------|
| <u>ST_LON_addrt</u> [▶ 550]        | Genutzt durch: SNVT_file_req |
| <u>ST_LON_dest_address</u> [▶ 550] | Genutzt durch: SNVT_file_req |
| <u>ST_LON_gp</u> [▶ 550]           | Genutzt durch: SNVT_file_req |
| <u>ST_LON_sn</u> [▶ 551]           | Genutzt durch: SNVT_file_req |

#### 7.3.66.10.1 ST\_LON\_addrt

Genutzt durch: SNVT\_file\_req

```
TYPE ST_LON_addrt :
STRUCT
  byType   : BYTE;
  uiIndex  : UINT;
END_STRUCT
END_TYPE
```

**byType:** Min: 0 / Max: 33 / Address type (8-bit unsigned value). The address-table address type is 33 (0x21).

**uiIndex:** Min: 0 / Max: 65535 / Address table index (16-bit unsigned value).

#### 7.3.66.10.2 ST\_LON\_dest\_address

Genutzt durch: SNVT\_file\_req

```
TYPE ST_LON_dest_address :
STRUCT
  stAddrnt : ST_LON_addrt;
  stSn     : ST_LON_sn;
  stGp    : ST_LON_gp;
END_STRUCT
END_TYPE
```

**stAddrnt:** Address table entry. ANSI/CEA-709.1 address in device's internal address table entry (see ST\_LON\_addrt [▶ 550]).

**stSn:** Subnet-node address (LonWorks subnet-node address) (see ST\_LON\_sn [▶ 551]).

**stGp:** Group address (LonWorks group address) (see ST\_LON\_gp [▶ 550]).

#### 7.3.66.10.3 ST\_LON\_gp

Genutzt durch: SNVT\_file\_req

```
TYPE ST_LON_gp :
STRUCT
  byType      : BYTE;
  bySize      : BYTE;
  bDomain     : BOOL;
  byUnused    : BYTE;
  byRetry     : BYTE;
  byTx_timer  : BYTE;
  byGroup     : BYTE;
END_STRUCT
END_TYPE
```

**byType:** Min: 0 / Max: 1 / Address type (boolean). The group address type is 1.

**bySize:** Min: 0 / Max: 65 / Size (LonWorks group size). An acknowledged group can have from 0-64 addressees, plus the sender.

**bDomain:** Min: 0 / Max: 1 / Domain (LonWorks domain index).

**byUnused:** Unused field. This field is reserved.

**byRetry:** Min: 0 / Max: 15 / Retry count (number of retries).

**byTx\_timer:** Min: 0 / Max: 15 / Transaction timer (timer code value).

**byGroup:** Min: 0 / Max: 255 / Group. There can be 256 groups (0-255) in a domain.

### 7.3.66.10.4 ST\_LON\_sn

Genutzt durch: SNVT\_file\_req

```

TYPE ST_LON_sn :
STRUCT
  byType      : BYTE;
  bDomain     : BOOL;
  byNode      : BYTE;
  byRetry     : BYTE;
  byTx_timer  : BYTE;
  bySubnet    : BYTE;
END_STRUCT
END_TYPE
    
```

**byType:** Min: 1 / Max: 1 / Address type (8-bit unsigned value). The subnet-node address type is 1.

**bDomain:** Domain (LonWorks domain index).

**byNode:** Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

**byRetry:** Min: 0 / Max: 15 / Retry count (number of retries).

**byTx\_timer:** Min: 0 / Max: 15 / Transaction timer (timer code value).

**bySubnet:** Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

### 7.3.66.11 SNVT\_file\_status

| Datentypen                | Beschreibung  |
|---------------------------|---|
| ST_LON_address [► 551]    | Genutzt durch: FB_Write_Address_Table / FB_Read_Address_Table |
| ST_LON_adr [► 551]        | Genutzt durch: SNVT_file_status                               |
| ST_LON_descriptor [► 552] | Genutzt durch: SNVT_file_status                               |

#### 7.3.66.11.1 ST\_LON\_address

Genutzt durch: FB\_Write\_Address\_Table / FB\_Read\_Address\_Table

```

TYPE ST_LON_address :
STRUCT
  arrDomain_id : ARRAY [0..5] OF BYTE;
  byDomain_length : BYTE;
  bySubnet      : BYTE;
  byNode        : BYTE;
END_STRUCT
END_TYPE
    
```

**arrDomain\_id:** Domain ID (array of 6 bytes). ANSI/CEA-709.1 domain ID

**byDomain\_length:** Min: 0 / Max: 6 / Domain length (ANSI/CEA-709.1 domain length). Valid domain lengths are 0, 1, 3, and 6.

**bySubnet:** Min: 0 / Max: 255 / Subnet (subnet number). There can be 255 subnets (1-255) in a domain.

**byNode:** Min: 0 / Max: 127 / Node (node number). There can be 127 nodes (1-127) in a subnet.

#### 7.3.66.11.2 ST\_LON\_adr

Genutzt durch: SNVT\_file\_status

```

TYPE ST_LON_adr :
STRUCT
  stDescriptor : ST_LON_descriptor;
  stAddress    : ST_LON_address;
END_STRUCT
END_TYPE
    
```

**stDescriptor:** Descriptor (siehe [ST\\_LON\\_descriptor](#) [▶ 552]).

**stAddress:** Address (siehe [ST\\_LON\\_address](#) [▶ 551]).

### 7.3.66.11.3 ST\_LON\_descriptor

Genutzt durch: SNVT\_file\_status

```

TYPE ST_LON_descriptor :
STRUCT
  sFile_info : STRING(16);
  udiSize    : UDINT;
  uiType     : UINT;
END_STRUCT
END_TYPE
    
```

**sFile\_info:** File info (array of 16 characters)

**udiSize:** Min: 0 / Max: 2147483647 / Size (bytes)

**uiType:** Min: 0 / Max: 65535 / Type

### 7.3.66.12 SNVT\_lamp\_status

| Datentypen                                    | Beschreibung                    |
|---|---------------------------------|
| <a href="#">ST_LON_Alarm_actual</a> [▶ 552]   | Genutzt durch: SNVT_lamp_status |
| <a href="#">ST_LON_alarm_previous</a> [▶ 554] | Genutzt durch: SNVT_lamp_status |

#### 7.3.66.12.1 ST\_LON\_Alarm\_actual

Genutzt durch: SNVT\_lamp\_status

```

TYPE ST_LON_Alarm_actual :
STRUCT
  bLamp_current_high : BOOL;
  bLamp_current_low  : BOOL;
  bMain_current_high : BOOL;
  bMain_current_low  : BOOL;
  bLamp_voltage_high : BOOL;
  bLamp_voltage_low  : BOOL;
  bMain_voltage_high : BOOL;
  bMain_voltage_low  : BOOL;
  bPowerfactor_low   : BOOL;
  bOLC_temp_high     : BOOL;
  bPower_high        : BOOL;
  bPower_low         : BOOL;
  bRelay_failure     : BOOL;
  bCap_failure       : BOOL;
  bLamp_failure      : BOOL;
  bBallast_failure   : BOOL;
  bInter_com_failure : BOOL;
  bExter_com_failure : BOOL;
  bMain_volt_below_spec : BOOL;
  bLamp_restart_count : BOOL;
  bFading_ready      : BOOL;
  bBallast_temp_high : BOOL;
  bDigi_in_A         : BOOL;
  bDigi_in_B         : BOOL;
  bBit_25_res        : BOOL;
  bBit_26_res        : BOOL;
  bBit_27_res        : BOOL;
  bBit_28_res        : BOOL;
  bBit_29_res        : BOOL;
  bBit_30_res        : BOOL;
  bBit_31_res        : BOOL;
END_STRUCT
    
```

```
bBit_32_res      : BOOL;  
bBit_33_res      : BOOL;  
bBit_34_res      : BOOL;  
bBit_35_res      : BOOL;  
bBit_36_res      : BOOL;  
bBit_37_res      : BOOL;  
bBit_38_res      : BOOL;  
bBit_39_res      : BOOL;  
bBit_40_res      : BOOL;  
END_STRUCT  
END_TYPE
```

**bLamp\_current\_high:** Lamp current too high.

**bLamp\_current\_low:** Lamp current too low.

**bMain\_current\_high:** Main current too high.

**bMain\_current\_low:** Main current too low.

**bLamp\_voltage\_high:** Lamp voltage too high.

**bLamp\_voltage\_low:** Lamp voltage too low.

**bMain\_voltage\_high:** Main voltage too high.

**bMain\_voltage\_low:** Main voltage too low.

**bPowerfactor\_low:** Powerfactor too low.

**bOLC\_temp\_high:** Temperature too high.

**bPower\_high:** Power too high.

**bPower\_low:** Power too low.

**bRelay\_failure:** Relay failure.

**bCap\_failure:** Capacitor failure.

**bLamp\_failure:** Lamp failure.

**bBallast\_failure:** Ballast failure.

**bInter\_com\_failure:** Internal communication failure.

**bExter\_com\_failure:** External communication failure.

**bMain\_volt\_below\_spec:** Main voltage below performance specification.

**bLamp\_restart\_count:** Lamp restart retry counter / cycling failure.

**bFading\_ready:** Fading ready.

**bBallast\_temp\_high:** Ballast temperature too high.

**bDigi\_in\_A:** digital input A active.

**bDigi\_in\_B:** digital input B active.

**bBit\_25\_res:**

**bBit\_26\_res:**

**bBit\_27\_res:**

**bBit\_28\_res:**

**bBit\_29\_res:**

**bBit\_30\_res:**

**bBit\_31\_res:**

**bBit\_32\_res:**

**bBit\_33\_res:**

**bBit\_34\_res:**

**bBit\_35\_res:**

**bBit\_36\_res:**

**bBit\_37\_res:**

**bBit\_38\_res:**

**bBit\_39\_res:**

**bBit\_40\_res:**

### 7.3.66.12.2 ST\_LON\_alarm\_previous

Genutzt durch: SNVT\_lamp\_status

```

TYPE ST_LON_alarm_previous :
STRUCT
  bLamp_current_high      : BOOL;
  bLamp_current_low      : BOOL;
  bMain_current_high     : BOOL;
  bMain_current_low     : BOOL;
  bLamp_voltage_high     : BOOL;
  bLamp_voltage_low     : BOOL;
  bMain_voltage_high    : BOOL;
  bMain_voltage_low     : BOOL;
  bPowerfactor_low      : BOOL;
  bOLC_temp_high        : BOOL;
  bPower_high           : BOOL;
  bPower_low            : BOOL;
  bRelay_failure        : BOOL;
  bCap_failure          : BOOL;
  bLamp_failure         : BOOL;
  bBallast_failure      : BOOL;
  bInter_com_failure    : BOOL;
  bExter_com_failure    : BOOL;
  bMain_volt_below_spec : BOOL;
  bLamp_restart_count   : BOOL;
  bFading_ready         : BOOL;
  bBallast_temp_high    : BOOL;
  bDigi_in_A            : BOOL;
  bDigi_in_B            : BOOL;
  bBit_25_res            : BOOL;
  bBit_26_res            : BOOL;
  bBit_27_res            : BOOL;
  bBit_28_res            : BOOL;
  bBit_29_res            : BOOL;
  bBit_30_res            : BOOL;
  bBit_31_res            : BOOL;
  bBit_32_res            : BOOL;
  bBit_33_res            : BOOL;
  bBit_34_res            : BOOL;
  bBit_35_res            : BOOL;
  bBit_36_res            : BOOL;
  bBit_37_res            : BOOL;
  bBit_38_res            : BOOL;
  bBit_39_res            : BOOL;
  bBit_40_res            : BOOL;
END_STRUCT
END_TYPE

```

**bLamp\_current\_high:** Lamp current too high.

**bLamp\_current\_low:** Lamp current too low.

**bMain\_current\_high:** Main current too high.

**bMain\_current\_low:** Main current too low.

**bLamp\_voltage\_high:** Lamp voltage too high.

**bLamp\_voltage\_low:** Lamp voltage too low.

**bMain\_voltage\_high:** Main voltage too high.

**bMain\_voltage\_low:** Main voltage too low.

**bPowerfactor\_low:** Powerfactor too low.

**bOLC\_temp\_high:** Temperature too high.

**bPower\_high:** Power too high.

**bPower\_low:** Power too low.

**bRelay\_failure:** Relay failure.

**bCap\_failure:** Capacitor failure.

**bLamp\_failure:** Lamp failure.

**bBallast\_failure:** Ballast failure.

**bInter\_com\_failure:** Internal communication failure.

**bExter\_com\_failure:** External communication failure.

**bMain\_volt\_below\_spec:** Main voltage below performance specification.

**bLamp\_restart\_count:** Lamp restart retry counter / cycling failure.

**bFading\_ready:** Fading ready.

**bBallast\_temp\_high:** Ballast temperature too high.

**bDigi\_in\_A:** digital input A active.

**bDigi\_in\_B:** digital input B active.

**bBit\_25\_res:**

**bBit\_26\_res:**

**bBit\_27\_res:**

**bBit\_28\_res:**

**bBit\_29\_res:**

**bBit\_30\_res:**

**bBit\_31\_res:**

**bBit\_32\_res:**

**bBit\_33\_res:**

**bBit\_34\_res:**

**bBit\_35\_res:**

**bBit\_36\_res:**

**bBit\_37\_res:**

**bBit\_38\_res:**

**bBit\_39\_res:**

**bBit\_40\_res:**

### 7.3.66.13 SNVT\_pos\_ctrl

| Datentypen                            | Beschreibung                 |
|---------------------------------------|------------------------------|
| <a href="#">ST_LON_abspos [► 556]</a> | Genutzt durch: SNVT_pos_ctrl |
| <a href="#">ST_LON_Value [► 556]</a>  | Genutzt durch: SNVT_pos_ctrl |

#### 7.3.66.13.1 ST\_LON\_abspos

Genutzt durch: SNVT\_pos\_ctrl

```
TYPE ST_LON_abspos :
STRUCT
  rPan    : REAL;
  rTilt   : REAL;
  rZoom   : REAL;
END_STRUCT
END_TYPE
```

**rPan:** Min: -359.98 / Max: 360 / Pan position.

**rTilt:** Min: -359.98 / Max: 360 / Tilt position.

**rZoom:** Min: -163.84 / Max: 163.835 / Zoom position.

#### 7.3.66.13.2 ST\_LON\_Value

Genutzt durch: SNVT\_pos\_ctrl

```
TYPE ST_LON_Value :
STRUCT
  byNumber : BYTE;
  stAbspos : ST_LON_abspos;
END_STRUCT
END_TYPE
```

**byNumber:** Min: 0 / Max: 255 / Action number (action number).

**stAbspos:** Function absolute values (pan, tilt, zoom) (see [ST\\_LON\\_abspos \[► 556\]](#)).

### 7.3.66.14 SNVT\_rac\_ctrl

| Datentypen                               | Beschreibung                                |
|--|---|
| <a href="#">ST_LON_addr_dest [► 556]</a> | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_addr_init [► 557]</a> | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_addr_talk [► 557]</a> | Genutzt durch: SNVT_rac_ctrl                |
| <a href="#">ST_LON_p2m [► 557]</a>       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |
| <a href="#">ST_LON_p2p [► 557]</a>       | Genutzt durch: SNVT_rac_req / SNVT_rac_ctrl |

#### 7.3.66.14.1 ST\_LON\_addr\_dest

Genutzt durch: SNVT\_rac\_req / SNVT\_rac\_ctrl

```
TYPE ST_LON_addr_dest :
STRUCT
  stP2p : ST_LON_p2p;
  stP2m : ST_LON_p2m;
END_STRUCT
END_TYPE
```

**stP2p:** (siehe [ST\\_LON\\_p2p \[► 557\]](#))

**stP2m:** (siehe [ST\\_LON\\_p2m \[► 557\]](#))

### 7.3.66.14.2 ST\_LON\_addr\_init

Genutzt durch: SNVT\_rac\_req / SNVT\_rac\_ctrl

```

TYPE ST_LON_addr_init :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id       : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE

```

**byUnit\_id:** Min: 0 / Max: 15

**byLocation:** Min: 0 / Max: 15

**byCar\_id:** Min: 0 / Max: 31

**byReserved:**

**eAudio\_sensor\_type:** see [E\\_LON\\_rail\\_audio\\_sensor\\_type\\_t](#) [► 520]

### 7.3.66.14.3 ST\_LON\_addr\_talk

Genutzt durch: SNVT\_rac\_ctrl

```

TYPE ST_LON_addr_talk :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id       : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE

```

**byUnit\_id:** Min: 0 / Max: 15

**byLocation:** Min: 0 / Max: 15

**byCar\_id:** Min: 0 / Max: 31

**byReserved:**

**eAudio\_sensor\_type:** see [E\\_LON\\_rail\\_audio\\_sensor\\_type\\_t](#) [► 520]

### 7.3.66.14.4 ST\_LON\_p2m

Genutzt durch: SNVT\_rac\_req / SNVT\_rac\_ctrl

```

TYPE ST_LON_p2m :
STRUCT
  byMask_unit    : BYTE;
  arrMask_car    : ARRAY [0..3] OF BYTE;
  arrMask_location : ARRAY [0..1] OF BYTE;
  arrMask_audio  : ARRAY [0..2] OF BYTE;
END_STRUCT
END_TYPE

```

**byMask\_unit:** Min: 0 / Max: 255

**arrMask\_car:** unsigned char [4] ??

**arrMask\_location:** unsigned char [2] ??

**arrMask\_audio:** unsigned char [3] ??

### 7.3.66.14.5 ST\_LON\_p2p

Genutzt durch: SNVT\_rac\_req / SNVT\_rac\_ctrl



```

TYPE ST_LON_p2p :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id       : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
    
```

**byUnit\_id:** Min: 0 / Max: 15

**byLocation:** Min: 0 / Max: 15

**byCar\_id:** Min: 0 / Max: 31

**byReserved:**

**eAudio\_sensor\_type:** see [E\\_LON\\_rail\\_audio\\_sensor\\_type\\_t \[► 520\]](#)

### 7.3.66.15 SNVT\_rac\_req

| Datentypen                                       | Beschreibung |
|--|--------------|
| ST_LON_rac_req_addr_dest <a href="#">[► 558]</a> |              |
| ST_LON_rac_req_addr_init <a href="#">[► 558]</a> |              |

#### 7.3.66.15.1 ST\_LON\_rac\_req\_addr\_dest

```

TYPE ST_LON_rac_req_addr_dest :
STRUCT
  stP2p : ST_LON_p2p;
  stP2m : ST_LON_p2m;
END_STRUCT
END_TYPE
    
```

**stP2p:** (see [ST\\_LON\\_p2p \[► 557\]](#))

**stP2m:** (see [ST\\_LON\\_p2m \[► 557\]](#))

#### 7.3.66.15.2 ST\_LON\_rac\_req\_addr\_init

```

TYPE ST_LON_rac_req_addr_init :
STRUCT
  byUnit_id      : BYTE;
  byLocation     : BYTE;
  byCar_id       : BYTE;
  byReserved     : BYTE;
  eAudio_sensor_type : E_LON_rail_audio_sensor_type_t;
END_STRUCT
END_TYPE
    
```

**byUnit\_id:**

**byLocation:**

**byCar\_id:**

**byReserved:**

**eAudio\_sensor\_type:** see [E\\_LON\\_rail\\_audio\\_sensor\\_type\\_t \[► 520\]](#)

### 7.3.66.16 SNVT\_switch\_2

| Datentypen                             | Beschreibung                 |
|--|------------------------------|
| ST_LON_setting <a href="#">[► 559]</a> | Genutzt durch: SNVT_switch_2 |

### 7.3.66.16.1 ST\_LON\_setting

Genutzt durch: SNVT\_switch\_2

```

TYPE ST_LON_setting :
STRUCT
  rValue      : REAL;
  rChange     : REAL;
  byDelay     : BYTE;
  byGroup_number : BYTE;
  rMultiplier : REAL;
  iAngle      : INT;
  siFan_level : SINT;
  byButton_number : BYTE;
END_STRUCT
END_TYPE
    
```

**rValue:** Min: 0 / Max: 100 / Value. Percent of full level when state is on. Reports last level for outputs when state is off.

**rChange:** Min: 0 / Max: 100 / Percent change. Percent change to level.

**byDelay:** Min: 0 / Max: 255 / On or off delay (seconds) (seconds). Time delay before changing state to on or off.

**byGroup\_number:** Min: 0 / Max: 63 / Group number. Group number that is enabled or disabled by the SW\_ENABLE\_GROUP and SW\_DISABLE\_GROUP states in the state field; if 0, all groups are enabled or disabled.

**rMultiplier:** Min: 0 / Max: 2.54 / Factor (percent). Multiplier for the level

**iAngle:** Min: -180 / Max: 180 / Rotation angle (degrees). Rotation angle for devices that support a rotation setting such as blinds.

**siFan\_level:** Min: -100 / Max: 100 / Fan level. Percent of full level fan speed when state is on. Reports last fan speed for outputs when state is off. Positive values represent the down direction, and negative values represent the up direction.

**byButton\_number:** Min: 0 / Max: 255 / Button number. The button number to activate when the state field is set to SW\_SET\_BUTTON, no invalid value

### 7.3.66.17 SNVT\_time\_zone

| Datentypen                                   | Beschreibung                  |
|--|-------------------------------|
| ST_LON_end_DST [ <a href="#">▶ 559</a> ]     | Genutzt durch: SNVT_time_zone |
| ST_LON_M_end_DST [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |
| ST_LON_M_start_DST [ <a href="#">▶ 560</a> ] | Genutzt durch: SNVT_time_zone |
| ST_LON_start_DST [ <a href="#">▶ 560</a> ]   | Genutzt durch: SNVT_time_zone |

### 7.3.66.17.1 ST\_LON\_end\_DST

Genutzt durch: SNVT\_time\_zone

```

TYPE ST_LON_end_DST :
STRUCT
  uiG_day_of_end_DST : UINT;
  uiJ_day_of_end_DST : UINT;
  stM_end_DST       : ST_LON_M_end_DST;
END_STRUCT
END_TYPE
    
```

**uiG\_day\_of\_end\_DST:** Min: 0 / Max: 365 / Gregorian calendar day of end DST (days).

**uiJ\_day\_of\_end\_DST:** Min: 1 / Max: 365 / Julian calendar day of end DST (days).

**stM\_end\_DST:** Meu calendar day of end DST (month, week, dateday) (see [ST\\_LON\\_M\\_end\\_DST \[\[▶ 560\]\(#\)\]](#)).

### 7.3.66.17.2 ST\_LON\_M\_end\_DST

Genutzt durch: SNVT\_time\_zone

```
TYPE ST_LON_M_end_DST :
STRUCT
  byMonth_of_end_DST   : BYTE;
  byWeek_of_end_DST   : BYTE;
  eDateday_of_end_DST : E_LON_days_of_week_t;
END_STRUCT
END_TYPE
```

**byMonth\_of\_end\_DST:** Min: 1 / Max: 12 / Month of end DST (months).

**byWeek\_of\_end\_DST:** Min: 1 / Max: 5 / Week of end DST (weeks).

**eDateday\_of\_end\_DST:** Day of week (day names, see [E\\_LON\\_days\\_of\\_week\\_t \[► 502\]](#)).

### 7.3.66.17.3 ST\_LON\_M\_start\_DST

Genutzt durch: SNVT\_time\_zone

```
TYPE ST_LON_M_start_DST :
STRUCT
  byMonth_of_start_DST : BYTE;
  byWeek_of_start_DST  : BYTE;
  eDateday_of_start_DST : E_LON_days_of_week_t;
END_STRUCT
END_TYPE
```

**byMonth\_of\_start\_DST:** Min: 1 / Max: 12 / Month of start DST (months).

**byWeek\_of\_start\_DST:** Min: 1 / Max: 5 / Week of start DST (weeks).

**eDateday\_of\_start\_DST:** Day of week (day names, see [E\\_LON\\_days\\_of\\_week\\_t \[► 502\]](#)).

### 7.3.66.17.4 ST\_LON\_start\_DST

Genutzt durch: SNVT\_time\_zone

```
TYPE ST_LON_start_DST :
STRUCT
  uiG_day_of_start_DST : UINT;
  uiJ_day_of_start_DST : UINT;
  stM_start_DST        : ST_LON_M_start_DST;
END_STRUCT
END_TYPE
```

**uiG\_day\_of\_start\_DST:** Min: 0 / Max: 365 / Gregorian calendar day of start DST (days).

**uiJ\_day\_of\_start\_DST:** Min: 1 / Max: 365 / Julian calendar day of start DST (days).

**stM\_start\_DST:** Meu calendar day of start DST (month, week, dateday) (see [ST\\_LON\\_M\\_start\\_DST \[► 560\]](#)).

## 7.3.67 ST\_LON\_Parameter\_IN\_36B

Prozessabbild der Eingänge.

Diese Variable muss im System Manager mit der KL6401 verknüpft werden.

```
TYPE ST_LON_Parameter_IN_36B :
STRUCT
  wParameterStatus : WORD;
  wDummy           : WORD;
  stParameterReadValue : ST_LON_ParameterInterface;
  byParameterType   : BYTE;
  byLONStatus       : BYTE;
END_STRUCT
END_TYPE
```

**wParameterStatus:** Status Word.

**wDummy:** Dummy für CX9000.

**stParameterReadValue:** Datenstruktur (siehe [ST\\_LON\\_ParameterInterface](#) [► 562]).

**byParameterType:** Parametertyp.

**byLONStatus:** LON Status.

### 7.3.68 ST\_LON\_Parameter\_OUT\_36B

Prozessabbild der Ausgänge

Diese Variable muss im System Manager mit der KL6401 verknüpft werden.

```
TYPE ST_LON_Parameter_OUT_36B :
STRUCT
  wParameterControl : WORD;
  wDummy           : WORD;
  stParametervalue : ST_LON_ParameterInterface;
  byCMD            : BYTE;
  byIdx            : BYTE;
END_STRUCT
END_TYPE
```

**wParameterControl:** Control Word.

**wDummy:** Dummy für CX9000.

**stParametervalue:** Datenstruktur (siehe [ST\\_LON\\_ParameterInterface](#) [► 562]).

**byCMD:** LON Kommando.

**byIdx:** LON Index.

### 7.3.69 ST\_ExplicitMessage

Explicit Message

```
TYPE ST_ExplicitMessage :
STRUCT
  byEcpM_1 : BYTE;
  byEcpM_2 : BYTE;
  byLen     : BYTE;
  byAddressingTyp : BYTE;
  byRetry   : BYTE;
  byRptTimer : BYTE;
  byTxTimer : BYTE;
  byRcvTimer : BYTE;
  bySubNet   : BYTE;
  byDestSubNet : BYTE;
  byNode     : BYTE;
  arrNeuronId : ARRAY[0..7] OF BYTE;
  byTyp       : BYTE;
  arrData     : ARRAY [0..47] OF BYTE;
END_STRUCT
END_TYPE
```

**byEcpM\_1:** Ecp M1

**byEcpM\_2:** Ecp M2

**byLen:** Länge

**byAddressingTyp:** Adresstyp

**byRetry:** Retry

**byRptTimer:** Rpt Timer

**byTxTimer:** Tx Timer

**byRcvTimer:** Rcv Timer

**bySubNet:** Sub Net

**byDestSubNet:** Ziel Sub Net

**byNode:** Node

**arrNeuronId:** Neuron Adresse

**byTyp:** Typ

**arrData:** 48 Datenbytes

### 7.3.70 ST\_LON\_Communication

Verbindung zwischen [FB\\_LON\\_KL6401 \[► 68\]](#) und den Sende-/Empfangsbausteinen.

```

TYPE ST_LON_Communication :
STRUCT
  arrWriteLONdata      : ARRAY[0..iLONBufferSize] OF ST_LON_WriteData;
  bWrite               : BOOL;
  bWriteBusy          : BOOL;
  bReadBusy           : BOOL;
  arrParameterReadValue : ARRAY[1..32] OF BYTE;
  wNV_Index           : WORD;
  eParameterDataType  : E_LON_Parameter_Datatypes;
  bWriteLONdataToTable : BOOL;
  bTerminalOk         : BOOL;
  byActBuffer          : BYTE;
  rActBuffer           : REAL;
  rMaxBuffer           : REAL;
END_STRUCT
END_TYPE
    
```

**arrWriteLONdata:** Sendepuffer.

**bWrite:** Daten werden gesendet.

**bWriteBusy:** Write-OnChange aktiv.

**bReadBusy:** Read-OnChange aktiv.

**arrParameterReadValue:** 32 Datenbytes.

**wNV\_Index:** NV-Index.

**eParameterDataType:** Datentyp der LON Variable (SNVT), see [E\\_LON\\_Parameter\\_Datatypes \[► 485\]](#).

**bWriteLONdataToTable:** Schreiben der Daten in die Tabelle ist aktiv.

**bTerminalOk:** Die Initialisierung der KL6401 wurde erfolgreich abgeschlossen.

**byActBuffer:** Anzahl der Aufträge im Sendepuffer.

**rActBuffer:** Aktuelle Auslastung des Sendepuffers in Prozent.

**rMaxBuffer:** Maximale Auslastung des Sendepuffers in Prozent. Der Wert kann mit der Eingangsvariable *bResetMaxBuffer* gelöscht werden.

### 7.3.71 ST\_LON\_ParameterInterface

LON Parameterinterface.

```

TYPE ST_LON_ParameterInterface :
STRUCT
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE
    
```

**arrParameterInterface:** 8 word data.

### 7.3.72 ST\_LON\_WriteData

Struktur des Sendepuffer

```
TYPE ST_LON_WriteData :
STRUCT
  wNVIndex      : WORD;
  udiSrcAddrWriteValue : UDINT;
  uiLenWriteValue : UINT;
  udiAdrBusy    : UDINT;
  udiAdrErrorKL : UDINT;
END_STRUCT
END_TYPE
```

**wNVIndex:** NV-Index

**udiSrcAddrWriteValue:** Adresse des zu sendenden Wertes

**uiLenWriteValue:** Länge des zu sendenden Wertes

**udiAdrBusy:** Adresse des Ausganges "bBusy"

**udiAdrErrorKL:** Adresse des Ausganges "dwErrorKL"

### 7.3.73 ST\_Prm

Struktur für Konfiguration

```
TYPE ST_Prm :
STRUCT
  byCMD      : BYTE;
  byIDX      : BYTE;
  wControl   : WORD;
  wStatus    : WORD;
  wParameterControl : WORD;
  wError     : WORD;
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE
```

**byCMD:** LON Command

**byIDX:** LON Index

**wControl:** Control Word

**wStatus:** Status Word

**wParameterControl:** Control Word

**wError:** Fehlerinformation

**arrParameterInterface:** 8 Data Word

### 7.3.74 ST\_KL6401

Struktur für die Konfiguration.

```
TYPE ST_KL6401 :
STRUCT
  wStatus      : WORD;
  wControl     : WORD;
  dwPointer_IN : DWORD;
  dwPointer_OUT : DWORD;
  arrParameterInterface : ARRAY[1..8] OF DWORD;
END_STRUCT
END_TYPE
```

**wStatus:** Status Word.

**wControl:** Control Word.

**dwPointer\_IN:** Pointer In.

**dwPointer\_OUT:** Pointer Out.

**arrParameterInterface:** 8 Word Daten.

### 7.3.75 ST\_LON\_AddressTable

Genutzt durch: FB\_Write\_Address\_Table / FB\_Read\_Address\_Table

```

TYPE ST_LON_AddressTable :
STRUCT
  bType      : BOOL;
  usiNode    : USINT;
  bDomain    : BOOL;
  usiMember  : USINT;
  usiRPT_Timer : USINT;
  usiRetry   : USINT;
  usiRCV_Timer : USINT;
  usiTx_Timer : USINT;
  byGroup   : BYTE;
END_STRUCT
END_TYPE

```

**bType:** Type

**usiNode:** Node

**bDomain:** Domain

**usiMember:** Member

**usiRPT\_Timer:** RPT Timer

**usiRetry:** Retry

**usiRCV\_Timer:** Rcv timer

**usiTx\_Timer:** Tx timer

**byGroup:** group

### 7.3.76 ST\_LON\_ConfigTable

Genutzt durch: FB\_Write\_Config\_Table / FB\_Read\_Config\_Table

```

TYPE ST_LON_ConfigTable :
STRUCT
  bPriority    : BOOL;
  bDirection  : BOOL;
  wSelector   : WORD;
  bTurnaround : BOOL;
  usiService  : USINT;
  bAuth       : BOOL;
  usiAddrIndex : USINT;
END_STRUCT
END_TYPE

```

**bPriority:** priority

**bDirection:** direction

**wSelector:** selector

**bTurnaround:** turn around

**usiService:** service

**bAuth:** auth

**usiAddrIndex:** Address index

### 7.3.77 ST\_LON\_DomainTable

Genutzt durch: FB\_Write\_Domain\_Table / FB\_Read\_Domain\_Table

```

TYPE ST_LON_DomainTable :
STRUCT
  arrDomainID      : ARRAY[0..5] OF BYTE;
  bySubNet         : BYTE;
  bCloneDomainBit : BOOL;
  byNode          : BYTE;
  byLen           : BYTE;
  arrKey          : ARRAY[0..5] OF BYTE := 16#FF;
END_STRUCT
END_TYPE

```

**arrDomainID:** domain id

**bySubNet:** sub net

**bCloneDomainBit:** Clone domain bit

**byNode:** node

**byLen:** Len

**arrKey:** key

### 7.3.78 ST\_LON\_SNVT\_alarm

Genutzt durch: SNVT\_alarm

```

TYPE ST_LON_SNVT_alarm :
STRUCT
  arrLocation      : ARRAY[0..5] OF BYTE;
  uiObject_Id     : UINT;
  eAlarm_type     : E_LON_alarm_type_t;
  ePriority_level  : E_LON_priority_level_t;
  uiIndex_To_SNVT : WORD;
  arrValue        : ARRAY [0..3] OF BYTE;
  uiYear          : UINT;
  uiMonth         : UINT;
  uiDay          : UINT;
  uiHour         : UINT;
  uiMinute       : UINT;
  uiSecond       : UINT;
  uiMilliseconds  : UINT;
  arrAlarm_limit  : ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE

```

**arrLocation:** Location (array of 6 bytes). Location code for the node

**uiObject\_Id:** Object ID (object index). ID of object within node

**eAlarm\_type:** Alarm type (alarm type names, see [E\\_LON\\_alarm\\_type\\_t](#) [▶ 493]).

**ePriority\_level:** Priority level (priority level names, see [E\\_LON\\_priority\\_level\\_t](#) [▶ 519]).

**uiIndex\_To\_SNVT:** Index of NV (index of NV causing alarm).

**arrValue:** Value (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

**uiYear:** Year (years). Zero (0) means year not specified.

**uiMonth:** Month (months). Zero (0) means month not specified.

**uiDay:** Day (days). Zero (0) means day not specified.

**uiHour:** Hour (hours). This field uses a 24-hour value.

**uiMinute:** Minute (minutes).

**uiSecond:** Second (seconds).



**uiMilliseconds:** Millisecond (milliseconds).

**arrAlarm\_limit:** Alarm limit (array of 4 bytes). The type of this field is dependent on the NV causing the alarm condition.

### 7.3.79 ST\_LON\_SNVT\_alarm\_2

Genutzt durch: SNVT\_alarm\_2

```
TYPE ST_LON_SNVT_alarm_2 :
STRUCT
  eAlarm_type      : E_LON_alarm_type_t;
  ePriority_level   : E_LON_priority_level_t;
  udiAlarm_time    : UDINT;
  iMilliseconds    : INT;
  bySequence_number : BYTE;
  sDescription     : STRING(22);
END_STRUCT
END_TYPE
```

**eAlarm\_type:** Alarm type (alarm type names, see [E\\_LON\\_alarm\\_type\\_t \[▶ 493\]](#)). Alarm condition reported by this update

**ePriority\_level:** Priority level (priority level names, see [E\\_LON\\_priority\\_level\\_t \[▶ 519\]](#)). Priority level of the alarm reported by this update

**udiAlarm\_time:** Alarm time (seconds). Alarm time in seconds since 2000-01-01T00:00:00Z (the 0 hour of 1 January 2000, Coordinated Universal Time)

**iMilliseconds:** Milliseconds (milliseconds). Alarm time in milliseconds since the second specified by the alarm\_time field

**bySequence\_number:** Sequence number(count). Sequence number for this update. Incremented by one for each update from an alarm source. Wraps to zero after reaching 255. An alarm receiver can use the sequence number to detect missed alarm messages.

**sDescription:** Description (array of 22 characters). Alarm description with NUL terminator. The terminator is not required if the description requires 22 characters. May include a reference to a language string, delimited by a 0x80 value.

### 7.3.80 ST\_LON\_SNVT\_chlr\_status

Genutzt durch: SNVT\_chlr\_status

```
TYPE ST_LON_SNVT_chlr_status :
STRUCT
  eChlr_run_mode   : E_LON_chiller_t;
  echlr_op_mode    : E_LON_hvac_t;
  stChlr_state     : ST_LON_chlr_state;
END_STRUCT
END_TYPE
```

**eChlr\_run\_mode:** Chiller run mode (chiller run mode names) (see [E\\_LON\\_chiller\\_t \[▶ 498\]](#)).

**echlr\_op\_mode:** Chiller operating mode (HVAC mode names) (see [E\\_LON\\_hvac\\_t \[▶ 514\]](#)).

**stChlr\_state:** Chiller state flags (alarm, enabled, local, limited, chiller water flow, condenser water flow) (see [ST\\_LON\\_chlr\\_state \[▶ 538\]](#)).

### 7.3.81 ST\_LON\_SNVT\_clothes\_w\_a

Genutzt durch: SNVT\_clothes\_w\_a

```
TYPE ST_LON_SNVT_clothes_w_a :
STRUCT
  bAlarm_reset     : BOOL;
  bWar_water_supply : BOOL;
  bWar_drain_slow  : BOOL;
  bWar_door_open   : BOOL;
END_STRUCT
```

```
bWar_load_unbalanced : BOOL;
bWar_filter_cleaning : BOOL;
bWar_hoses_reversed  : BOOL;
bWar_voltage_low     : BOOL;
bWar_power_failure   : BOOL;
bWar_drain_open      : BOOL;
bWar_execute_fail    : BOOL;
bWar_door_locked     : BOOL;
bWar_service         : BOOL;
bRsrvd5              : BOOL;
bRsrvd6              : BOOL;
bRsrvd7              : BOOL;
bErr_motor_stall     : BOOL;
bErr_water_temp      : BOOL;
bErr_pressure        : BOOL;
bErr_overflow        : BOOL;
bErr_water_heat      : BOOL;
bErr_water_leak      : BOOL;
bErr_motor_speed     : BOOL;
bErr_wash_thermistor : BOOL;
bErr_dry_thermistor  : BOOL;
bErr_dry_overheat    : BOOL;
bErr_dry_heating     : BOOL;
bErr_dry_fan         : BOOL;
bErr_rsrvd4          : BOOL;
bErr_rsrvd5          : BOOL;
bErr_rsrvd6          : BOOL;
bErr_rsrvd7          : BOOL;
byErr_rsrvd0_7       : BYTE;
byManuf_code         : BYTE;
END_STRUCT
END_TYPE
```

**bAlarm\_reset:**

**bWar\_water\_supply:**

**bWar\_drain\_slow:**

**bWar\_door\_open:**

**bWar\_load\_unbalanced:**

**bWar\_filter\_cleaning:**

**bWar\_hoses\_reversed:**

**bWar\_voltage\_low:**

**bWar\_power\_failure:**

**bWar\_drain\_open:**

**bWar\_execute\_fail:**

**bWar\_door\_locked:**

**bWar\_service:**

**bRsrvd5:**

**bRsrvd6:**

**bRsrvd7:**

**bErr\_motor\_stall:**

**bErr\_water\_temp:**

**bErr\_pressure:**

**bErr\_overflow:**

**bErr\_water\_heat:**

**bErr\_water\_leak:**

**bErr\_motor\_speed:**

**bErr\_wash\_thermistor:**

**bErr\_dry\_thermistor:**

**bErr\_dry\_overheat:**

**bErr\_dry\_heating:**

**bErr\_dry\_fan:**

**bErr\_rsrvd4:**

**bErr\_rsrvd5:**

**bErr\_rsrvd6:**

**bErr\_rsrvd7:**

**byErr\_rsrvd0\_7:**

**byManuf\_code:** Min: 0 / Max: 255

### 7.3.82 ST\_LON\_SNVT\_clothes\_w\_c

Genutzt durch: SNVT\_clothes\_w\_c

```
TYPE ST_LON_SNVT_clothes_w_c :
STRUCT
  eCycle      : E_LON_appl_cwc_t;
  eSubcycle   : E_LON_appl_cws_t;
  byRervd     : BYTE;
  stAction    : ST_LON_action;
  stFunction  : ST_LON_function;
  uiTime_remaining : UINT;
END_STRUCT
END_TYPE
```

**eCycle:** (siehe [E\\_LON\\_appl\\_cwc\\_t](#) [▶ 495])

**eSubcycle:** (siehe [E\\_LON\\_appl\\_cws\\_t](#) [▶ 496])

**byRervd:**

**stAction:** (siehe [ST\\_LON\\_action](#) [▶ 538])

**stFunction:** (siehe [ST\\_LON\\_function](#) [▶ 539])

**uiTime\_remaining:** Min: 0 / Max: 65535.

### 7.3.83 ST\_LON\_SNVT\_clothes\_w\_m

Genutzt durch: SNVT\_clothes\_w\_m

```
TYPE ST_LON_SNVT_clothes_w_m :
STRUCT
  bDoor_ajar  : BOOL;
  bDrain_on   : BOOL;
  byReserved  : BYTE;
END_STRUCT
END_TYPE
```

**bDoor\_ajar:** Door/Lid Ajar. The door/lid of the washer is not fully closed.

**bDrain\_on:** Drain On. The drain is on

**byReserved:** Reserve

### 7.3.84 ST\_LON\_SNVT\_clothes\_w\_s

Genutzt durch: SNVT\_clothes\_w\_s

```

TYPE ST_LON_SNVT_clothes_w_s :
STRUCT
  eCycle      : E_LON_appl_cwc_t;
  eSubcycle   : E_LON_appl_cws_t;
  stWasher_command_data : ST_LON_SNVT_clothes_w_c;
  uiTime_remaining : UINT;
  stAlarm     : ST_LON_alarm;
END_STRUCT
END_TYPE

```

**eCycle:** (siehe [E\\_LON\\_appl\\_cwc\\_t](#) [▶ 495])

**eSubcycle:** (siehe [E\\_LON\\_appl\\_cws\\_t](#) [▶ 496])

**stWasher\_command\_data:** (siehe [ST\\_LON\\_SNVT\\_clothes\\_w\\_c](#) [▶ 568])

**uiTime\_remaining:** Min: 0 / Max: 65535.

**stAlarm:** (siehe [ST\\_LON\\_alarm](#) [▶ 540])

### 7.3.85 ST\_LON\_SNVT\_color

Genutzt durch: SNVT\_color

```

TYPE ST_LON_SNVT_color :
STRUCT
  rL_star : REAL;
  rA_star : REAL;
  rB_star : REAL;
END_STRUCT
END_TYPE

```

**rL\_star:** Min: 0 / Max: 100.0 / L\*

**rA\_star:** Min: -200.0 / Max: 200.0 / a\*

**rB\_star:** Min: -200.0 / Max: 200.0 / b\*

### 7.3.86 ST\_LON\_SNVT\_color\_2

Genutzt durch: SNVT\_color\_2

```

TYPE ST_LON_SNVT_color_2 :
STRUCT
  eEncoding      : E_LON_color_encoding_t;
  stColor_value  : ST_LON_color_value;
END_STRUCT
END_TYPE

```

**eEncoding:** Color encoding. Color encoding specified by the color\_value union; additional encodings may be added (see [E\\_LON\\_color\\_encoding\\_t](#) [▶ 498]).

**stColor\_value:** Color value. Color value encoded as specified by the encoding field (see [ST\\_LON\\_color\\_value](#) [▶ 543]).

### 7.3.87 ST\_LON\_SNVT\_ctrl\_req

Genutzt durch: SNVT\_ctrl\_req

```

TYPE ST_LON_SNVT_ctrl_req :
STRUCT
  uiReceiver_id : UINT;
  uiSender_id   : UINT;
  bySender_prio : BYTE;
END_STRUCT
END_TYPE

```

**uiReceiver\_id:** Min: 1 / Max: 65535 / Invalid / 0 / Receiver ID (ID number)

**uiSender\_id:** Min: 1 / Max: 65535 / Invalid / 65535 / Sender ID (ID number)

**bySender\_prio:** Min: 0 / Max: 200 / Sender priority (priority value)

### 7.3.88 ST\_LON\_SNVT\_ctrl\_resp

Genutzt durch: SNVT\_ctrl\_resp

```
TYPE ST_LON_SNVT_ctrl_resp :
STRUCT
  eStatus      : E_LON_control_resp_t;
  stSender     : ST_LON_sender;
  uiController_id : UINT;
END_STRUCT
END_TYPE
```

**eStatus:** Control response type (control response type names) (see [E\\_LON\\_control\\_resp\\_t \[▶ 499\]](#)).

**stSender:** Sender ID (see [ST\\_LON\\_sender \[▶ 544\]](#)).

**uiController\_id:** Min: 1 / Max: 65535 / Invalid: 65535 / Controller ID (ID number).

### 7.3.89 ST\_LON\_SNVT\_currency

Genutzt durch: SNVT\_currency

```
TYPE ST_LON_SNVT_currency :
STRUCT
  eCurrency      : E_LON_currency_t;
  siPower_of_10 : SINT;
  diValue       : DINT;
END_STRUCT
END_TYPE
```

**eCurrency:** Currency (currency names, see [E\\_LON\\_currency\\_t \[▶ 499\]](#))

**siPower\_of\_10:** Min: -128 / Max: 127 / Magnitude (power of 10)

**diValue:** Min: -2147483648 / Max: 2147483647 / Value (currency value). Credit is positive, debit is negative.

### 7.3.90 ST\_LON\_SNVT\_date\_event

Genutzt durch: SNVT\_date\_event

```
TYPE ST_LON_SNVT_date_event :
STRUCT
  iDays_to_active      : INT;
  iDays_to_inactive   : INT;
  sName                : STRING(22);
END_STRUCT
END_TYPE
```

**iDays\_to\_active:** Min: -32768 / Max: 32767 / Invalid: 32767 / Days to active (days). Number of days until this schedule will be active. Positive if a schedule is inactive; zero or negative if a schedule is active.

**iDays\_to\_inactive:** Min: -32768 / Max: 32767 / Invalid: -32768 / Days to inactive (days). Number of days until this schedule will be inactive. Positive if a schedule is active; zero or negative if a schedule is inactive.

**sName:** 22 characters / Schedule name (array of 22 characters). Nul-terminated schedule name. The nul terminator is not required if the name is 22 characters.

### 7.3.91 ST\_LON\_SNVT\_dev\_fault

Genutzt durch: SNVT\_dev\_fault

```

TYPE ST_LON_SNVT_dev_fault :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type1;
END_STRUCT
END_TYPE

```

**eDevice\_select:** Device selection (device selection names). Determines the interpretation of the network-variable content (see [E\\_LON\\_device\\_select\\_t](#) [► 505]).

**stDev\_type:** Union of device fault structures for various devices (see [ST\\_LON\\_Dev\\_type1](#) [► 544]).

### 7.3.92 ST\_LON\_SNVT\_dev\_maint

Genutzt durch: SNVT\_dev\_maint

```

TYPE ST_LON_SNVT_dev_maint :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type2;
END_STRUCT
END_TYPE

```

**eDevice\_select:** Device selection (device selection names). Determines the interpretation of the network-variable content (see [E\\_LON\\_device\\_select\\_t](#) [► 505]).

**stDev\_type:** Union of device maintenance state structures for various devices (see [ST\\_LON\\_Dev\\_type2](#) [► 546]).

### 7.3.93 ST\_LON\_SNVT\_dev\_status

Genutzt durch: SNVT\_dev\_status

```

TYPE ST_LON_SNVT_dev_status :
STRUCT
  eDevice_select : E_LON_device_select_t;
  stDev_type     : ST_LON_Dev_type3;
END_STRUCT
END_TYPE

```

**eDevice\_select:** Device selection (device selection names). Determines the interpretation of the network-variable content (see [E\\_LON\\_device\\_select\\_t](#) [► 505]).

**stDev\_type:** Union of device status for various devices (see [ST\\_LON\\_Dev\\_type3](#) [► 547]).

### 7.3.94 ST\_LON\_SNVT\_earth\_pos

Genutzt durch: SNVT\_earth\_pos

```

TYPE ST_LON_SNVT_earth_pos :
STRUCT
  bLatitude_direction : BOOL;
  bLongitude_direction : BOOL;
  byLatitude_deg      : BYTE;
  rLatitude_min       : REAL;
  byLongitude_deg     : BYTE;
  rLongitude_min      : REAL;
  rHeight_above_sea  : REAL;
END_STRUCT
END_TYPE

```

**bLatitude\_direction:** FALSE = South latitude, TRUE = North latitude

**bLongitude\_direction:** FALSE = East longitude, TRUE = West longitude

**byLatitude\_deg:** Min: 0 / Max: 90 / Invalid: 255 / Latitude degrees (degrees).

**rLatitude\_min:** Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Latitude minutes (minutes).

**byLongitude\_deg:** Min: 0 / Max: 180 / Invalid: 255 / Longitude degrees (degrees).

**rLongitude\_min:** Min: 0.0 / Max: 59.999 / Invalid: 65.535/ Longitude minutes (minutes).

**rHeight\_above\_sea:** Min: -3.40E+38 / Max: 3.40E+38 / Height above sea level (meters).

### 7.3.95 ST\_LON\_SNVT\_elapsed\_tm

Genutzt durch: SNVT\_elapsed\_tm

```
TYPE ST_LON_SNVT_elapsed_tm :
STRUCT
  uiDay      : UINT;
  uiHour     : UINT;
  uiMinute   : UINT;
  uiSecond   : UINT;
  uiMillisecond : UINT;
END_STRUCT
END_TYPE
```

**uiDay:** Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

**uiHour:** Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

**uiMinute:** Min: 0 / Max: 59 / Minutes (minutes).

**uiSecond:** Min: 0 / Max: 59 / Seconds (seconds).

**uiMillisecond:** Min: 0 / Max: 999 / Milliseconds (milliseconds).

### 7.3.96 ST\_LON\_SNVT\_ent\_status

Genutzt durch: SNVT\_ent\_status

```
TYPE ST_LON_SNVT_ent_status :
STRUCT
  bUnlocked      : BOOL;
  bLocked        : BOOL;
  bSecurity_locked : BOOL;
  bClosed        : BOOL;
  bOpen          : BOOL;
  bIn_alarm      : BOOL;
  bIn_error_cond : BOOL;
  bOpen_pre_alarm : BOOL;
  bOpen_alarm    : BOOL;
  bService_alarm : BOOL;
  bTamper        : BOOL;
  bEntry_req     : BOOL;
  bExit_req      : BOOL;
  bKey_req       : BOOL;
  bSafety_ext_req : BOOL;
  bEmergency_req : BOOL;
  bUnable_lock   : BOOL;
  bUnable_unlock : BOOL;
  bStuck         : BOOL;
  bForced_open   : BOOL;
  bForced_close  : BOOL;
  bOpening       : BOOL;
  bClosing       : BOOL;
  bMoving        : BOOL;
  bStopped       : BOOL;
  bSafety_alarm  : BOOL;
  bUnknown_state : BOOL;
  eCmd_fb       : E_LON_ent_opmode_cmd_t;
END_STRUCT
END_TYPE
```

**bUnlocked:** Unlocked device (boolean). Device is in unlocked position.

**bLocked:** Locked device (boolean). Device is in locked position.

**bSecurity\_locked:** Security locked (boolean). Device is in a security-driven locked position.

**bClosed:** Closed device (boolean). Device is in a closed position

**bOpen:** Open device (boolean). Device is in an open position.

**bln\_alarm:** In alarm state (boolean). The device is in the alarm state.

**bln\_error\_cond:** In error condition (boolean). Device has an error condition

**bOpen\_pre\_alarm:** Open device, pre-alarm (boolean). Device is open, and in warning state.

**bOpen\_alarm:** Open Device, alarm state (boolean). Device is open, and in not-closed alarm state.

**bService\_alarm:** Service alarm (boolean). Device needs service.

**bTamper:** Tamper mode (boolean). Device has detected tamper.

**bEntry\_req:** Entry request pending (boolean). Device has a pending entry request.

**bExit\_req:** Exit request pending (boolean). Device has a pending exit request.

**bKey\_req:** Key request pending (boolean). Device has a pending key request.

**bSafety\_ext\_req:** Safety-exit request pending (boolean). Device has a pending safety-exit request.

**bEmergency\_req:** Emergency-exit request pending (boolean). Device has a pending emergency-exit request.

**bUnable\_lock:** Unable to lock (boolean). Device is unable to close and/or lock

**bUnable\_unlock:** Unable to unlock (boolean). Device is unable to open and/or unlock

**bStuck:** Device is stuck (boolean). Device is unable to move.

**bForced\_open:** Forced-open Device (boolean). Device is/was forced to go to an open position.

**bForced\_close:** Forced-closed Device (boolean). Device is/was forced to go to a closed position.

**bOpening:** Device is opening (boolean). Device is currently opening from a closed position.

**bClosing:** Device is closing (boolean). Device is currently closing from an open position.

**bMoving:** Device is in motion (boolean). Device is currently changing position.

**bStopped:** Device Stopped (boolean). The device is stopped and can be moved manually.

**bSafety\_alarm:** Safety-alarm (boolean). Device is in a safety-alarm state.

**bUnknown\_state:** Unknown state (boolean). The state of the device is currently unknown

**eCmd\_fb:** Command feedback (entry command names, see [E\\_LON\\_ent\\_opmode\\_cmd\\_t \[► 506\]](#)). Feedback of requested-operation-mode of device

### 7.3.97 ST\_LON\_SNVT\_environment

Genutzt durch: SNVT\_environment

```

TYPE ST_LON_SNVT_environment :
STRUCT
  uiLampCurrent      : UINT;
  uiLampVoltage      : UINT;
  uiSupplyVoltage    : UINT;
  uiSupplyCurrent    : UINT;
  rBallastTemp       : REAL;
  rPower             : REAL;
  rPowerFactor       : REAL;
  udiRunHours        : UDINT;
  lrEnergy           : LREAL;
END_STRUCT
END_TYPE

```

**uiLampCurrent:** Min: 0 / Max: 65534 / Invalid: 65535 / Lamp current (milliAmperes). This is the current the lamp consumes.

**uiLampVoltage:** Min: 0 / Max: 65535/ Lamp Voltage (Volts). This is the lamp voltage.

**uiSupplyVoltage:** Min: 0 / Max: 65535 / Supply Voltage (Volts). This is the luminaire supply voltage.



**uiSupplyCurrent:** Min: 0 / Max: 65534 / Invalid: 65535 / Supply Current (milliAmperes). This is the luminaire supply current.

**rBallastTemp:** Min: -273.17 / Max: 327.67 / Ballast temperature (degrees Celsius). This is the temperature at the ballast.

**rPower:** Min: 0 / Max: 6553.5 / Power (Watts). The value shows the at this moment consumed power of the ballast and the luminaire.

**rPowerFactor:** Min: -1 / Max: 1 / Power factor. This is the luminaire power-factor.

**udiRunHours:** Min: 0 / Max: 4294967294 / Run Hours (hours). This are the run hours since the last maintenance.

**IrEnergy:** Min: -214748364.8 / Max: 214748364.7 / Energy (kiloWatt-hours). This is the energy the luminair has consumt since the last maintenance.

### 7.3.98 ST\_LON\_SNVT\_ex\_control

Genutzt durch: SNVT\_ex\_control

```
TYPE ST_LON_SNVT_ex_control :
STRUCT
  eControl_status      : E_LON_ex_control_t;
  stControl_device_addr : ST_LON_Control_device_addr;
END_STRUCT
END_TYPE
```

**eControl\_status:** Control type (control type names) (see [E\\_LON\\_ex\\_control\\_t \[► 508\]](#)).

**stControl\_device\_addr:** Control device address (LonWorks subnet-node address) (see [ST\\_LON\\_Control\\_device\\_addr \[► 549\]](#)).

### 7.3.99 ST\_LON\_SNVT\_file\_pos

Genutzt durch: SNVT\_file\_pos

```
TYPE ST_LON_SNVT_file_pos :
STRUCT
  diRw_ptr      : DINT;
  uiRw_length   : UINT;
END_STRUCT
END_TYPE
```

**diRw\_ptr:** Min: 0 / Max: 2147483647 / Read/Write pointer (file byte address).

**uiRw\_length:** Min: 0 / Max: 65535 / Read/Write length (number of bytes).

### 7.3.100 ST\_LON\_SNVT\_file\_req

Genutzt durch: SNVT\_file\_req

```
TYPE ST_LON_SNVT_file_req :
STRUCT
  eRequest      : E_LON_file_request_t;
  uiIndex       : UINT;
  uiReceive_timeout : UINT;
  stDest_address : ST_LON_dest_address;
  byAuth_on    : BYTE;
  byPrio_on    : BYTE;
END_STRUCT
END_TYPE
```

**eRequest:** Request (file request names) (see [E\\_LON\\_file\\_request\\_t \[► 508\]](#)).

**uiIndex:** Min: 0 / Max: 65535 / Index (file index).

**uiReceive\_timeout:** Min: 0 / Max: 65535 / Receive timeout (milliseconds).

**stDest\_address:** Destination address (LonWorks address) (see [ST\\_LON\\_dest\\_address \[► 550\]](#)).

**byAuth\_on:** Min: 0 / Max: 1 / Authentication on (boolean). This field specifies whether the message requires authentication.

**byPrio\_on:** Min: 0 / Max: 1 / Priority on (boolean). This field specifies whether the message is to be sent with priority.

### 7.3.101 ST\_LON\_SNVT\_file\_status

Genutzt durch: SNVT\_file\_status

```
TYPE ST_LON_SNVT_file_status :
STRUCT
  eStatus      : E_LON_file_status_t;
  uiNumber_of_files : UINT;
  uiSelected_file  : UINT;
END_STRUCT
END_TYPE
```

**eStatus:** Status (file status names, see [E\\_LON\\_file\\_status\\_t \[► 508\]](#) ).

**uiNumber\_of\_files:** Min: 0 / Max: 65535 / Number of files (count).

**uiSelected\_file:** Min: 0 / Max: 65535 / Selected file (file index).

### 7.3.102 ST\_LON\_SNVT\_geo\_loc

Genutzt durch: SNVT\_geo\_loc

```
TYPE ST_LON_SNVT_geo_loc :
STRUCT
  lrLongitude  : LREAL;
  lrLatitude   : LREAL;
  rElevation   : REAL;
  sName        : STRING(19);
END_STRUCT
END_TYPE
```

**lrLongitude:** Min: -180.0 / Max: 180.0 / Longitude. Longitude is given as an angular measurement ranging from 0° at the prime meridian to +180° eastward and -180° westward

**lrLatitude:** Min: -90.0 / Max: 90.0 / Latitude. Latitude is given as an angular measurement ranging from 0° at the equator to +90° northward and -90° southward

**rElevation:** Min: -3.40E+51 / Max: 3..40+51 / Elevation (meters).

**sName:**

### 7.3.103 ST\_LON\_SNVT\_hvac\_overid

Genutzt durch: SNVT\_hvac\_overid

```
TYPE ST_LON_SNVT_hvac_overid :
STRUCT
  eState      : E_LON_hvac_overid_t;
  rPercent    : REAL;
  uiFlow      : UINT;
END_STRUCT
END_TYPE
```

**eState:** HVAC override state (override state names, see [E\\_LON\\_hvac\\_overid\\_t \[► 512\]](#)).

**rPercent:** Min: -163,84 / Max: 163,835 / Percent (% of full scale). Position or flow override value.

**uiFlow:** Min: 0 / Max: 65535 / Flow (liters/second).

### 7.3.104 ST\_LON\_SNVT\_hvac\_satsts

Genutzt durch: SNVT\_hvac\_satsts

```

TYPE ST_LON_SNVT_hvac_satsts :
STRUCT
  bPri_heat      : BOOL;
  bSec_heat      : BOOL;
  bPri_cool      : BOOL;
  bSec_cool      : BOOL;
  bPri_duct_starved : BOOL;
  bSec_duct_starved : BOOL;
  bReserved1     : BOOL;
  bReserved2     : BOOL;
  byReserved1    : BYTE;
  byManufacturer_defined : BYTE;
END_STRUCT
END_TYPE
    
```

**bPri\_heat:** Primary heating saturation status (boolean). A value of 0 indicates primary heating is not saturated. A value of 1 indicates primary heating is saturated.

**bSec\_heat:** Secondary heating saturation status (boolean). A value of 0 indicates secondary heating is not saturated. A value of 1 indicates secondary heating is saturated.

**bPri\_cool:** Primary cooling saturation status (boolean). A value of 0 indicates primary cooling is not saturated. A value of 1 indicates primary cooling is saturated.

**bSec\_cool:** Secondary cooling saturation status (boolean). A value of 0 indicates secondary cooling is not saturated. A value of 1 indicates secondary cooling is saturated.

**bPri\_duct\_starved:** Primary duct saturation status (boolean). A value of 0 indicates primary duct is not saturated (starved). A value of 1 indicates primary duct is saturated (starved).

**bSec\_duct\_starved:** Secondary duct saturation status (boolean). A value of 0 indicates secondary duct is not saturated (starved). A value of 1 indicates secondary duct is saturated (starved).

**bReserved1:**

**bReserved2:**

**byReserved1:** Min: 0 / Max: 15

**byManufacturer\_defined:** Min: 0 / Max: 15 / Manufacturer defined (boolean). Four manufacturer-defined bits -- please see product documentation for proper interpretation of these bits

### 7.3.105 ST\_LON\_SNVT\_hvac\_status

Genutzt durch: SNVT\_hvac\_status

```

TYPE ST_LON_SNVT_hvac_status :
STRUCT
  eMode          : E_LON_hvac_t;
  rHeat_output_primary : REAL;
  rHeat_output_secondary : REAL;
  rCool_output    : REAL;
  rEcon_output    : REAL;
  rFan_output     : REAL;
  byIn_alarm      : BYTE;
END_STRUCT
END_TYPE
    
```

**eMode:** HVAC status mode (HVAC mode names, see [E\\_LON\\_hvac\\_t \[► 514\]](#)).

**rHeat\_output\_primary:** Min: -163.8400 / Max: 163.8350 / Primary heat output (% of full scale).

**rHeat\_output\_secondary:** Min: -163.8400 / Max: 163.8350 / Secondary heat output (% of full scale).

**rCool\_output:** Min: -163.8400 / Max: 163.8350 / Cooling output (% of full scale).

**rEcon\_output:** Min: -163.8400 / Max: 163.8350 / Economizer output (% of full scale).

**rFan\_output:** Min: -163.8400 / Max: 163.8350 / Fan output (% of full scale).

**byIn\_alarm:** Min: 0 / Max: 255 / in\_alarm

### 7.3.106 ST\_LON\_SNVT\_lamp\_status

Genutzt durch: SNVT\_lamp\_status

```
TYPE ST_LON_SNVT_lamp_status :
STRUCT
    stTime_actual      : TIMESTRUCT;
    stAlarm_actual    : ST_LON_Alarm_actual;
    stTime_previous   : TIMESTRUCT;
    stAlarm_previous  : ST_LON_alarm_previous;
END_STRUCT
END_TYPE
```

**stTime\_actual:** Actual alarm message. This is the time stamp for the actual alarm message (see [TIMESTRUCT](#)).

**stAlarm\_actual:** (see [ST\\_LON\\_Alarm\\_actual](#) [► 552])

**stTime\_previous:** Time Stamp Old. This is the time stamp for an old alarm message (see [TIMESTRUCT](#)).

**stAlarm\_previous:** (see [ST\\_LON\\_alarm\\_previous](#) [► 554])

### 7.3.107 ST\_LON\_SNVT\_log\_fx\_request

Genutzt durch: SNVT\_log\_fx\_request

```
TYPE ST_LON_SNVT_log_fx_request :
STRUCT
    uiRequested_log   : UINT;
    udiRecord_count  : UDINT;
    stStart_time     : TIMESTRUCT;
    stEnd_time       : TIMESTRUCT;
END_STRUCT
END_TYPE
```

**uiRequested\_log:** Min: 1 / Max: 65535 / Requested log number. The log number of the data log to be transferred. Logs are numbered from 1 to number\_of\_logs.

**udiRecord\_count:** Min: 0 / Max: 4294967295 / Record count. The maximum number of log records to be transferred.

**stStart\_time:** Start time. (seconds). Timestamp of first record to be transferred. If no records exist with this timestamp, the first record with a timestamp after this timestamp is the starting record.

**stEnd\_time:** End time. (seconds). Timestamp of last record to be transferred. If no records exist with this timestamp, the last record with a timestamp before this timestamp is the ending record.

### 7.3.108 ST\_LON\_SNVT\_log\_fx\_status

Genutzt durch: SNVT\_log\_fx\_status

```
TYPE ST_LON_SNVT_log_fx_status :
STRUCT
    byRequestor_subnet : BYTE;
    byRequestor_node   : BYTE;
    uiLog_number       : UINT;
    rComplete          : REAL;
END_STRUCT
END_TYPE
```

**byRequestor\_subnet:** Min: 1 / Max: 255 / Requestor subnet ID. Subnet ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

**byRequestor\_node:** Min: 1 / Max: 255 / Requestor node ID. Node ID of the device that requested the current log file transfer. Invalid if a file transfer is not active.

**uiLog\_number:** Min: 1 / Max: 65535 / Data log number. Log number for the log file currently being transferred via FTP. Invalid if none.

**rComplete:** Min: 0 / Max: 100 / Data log file transfer percent complete. Percent of the current data log file transfer that has been completed. Invalid if none.

### 7.3.109 ST\_LON\_SNVT\_log\_status

Genutzt durch: SNVT\_log\_status

```

TYPE ST_LON_SNVT_log_status :
STRUCT
  eStatus          : E_LON_log_status_t;
  uiLog_number     : UINT;
  rLevel           : REAL;
  diRecord_count   : DINT;
  diByte_count     : DINT;
  diTotal_record_count : DINT;
  diRecords_since_notification : DINT;
  stCurrent_notify_time : TIMESTRUCT;
  stPrevious_notify_time : TIMESTRUCT;
END_STRUCT
END_TYPE
    
```

**eStatus:** Log state [► 516]. State of the selected data log.

**uiLog\_number:** Min: 1 / Max: 65535 / Selected log number. The log number of the reported data log. Logs are numbered from 1 to number\_of\_logs.

**rLevel:** Min: 0 / Max: 100 / Log level (Percent). The percent of maximum records in the selected data log.

**diRecord\_count:** Min: -2147483648 / Max: 2147483647 / Record count.(records). Number of records in the selected data log. A record is a logged value and any associated data such as a timestamp.

**diByte\_count:** Min: -2147483648 / Max: 2147483647 / Byte count. (bytes). Number of bytes in the selected data log.

**diTotal\_record\_count:** Min: -2147483648 / Max: 2147483647 / Total record count. (records). Total records collected in the selected data log since the data log was created. Wraps to 0 on overflow.

**diRecords\_since\_notification:** Min: -2147483648 / Max: 2147483647 / Records since notification. (records). The number of log records collected since the last notification.

**stCurrent\_notify\_time:** Current notify time. Timestamp of the most recently collected data point.

**stPrevious\_notify\_time:** Previous notify time. (seconds). Timestamp of the most recently collected data point in the previous update to the log status.

### 7.3.110 ST\_LON\_SNVT\_muldiv

Genutzt durch: SNVT\_muldiv

```

TYPE ST_LON_SNVT_muldiv :
STRUCT
  uiMultiplier : UINT;
  uiDivisor     : UINT;
END_STRUCT
END_TYPE
    
```

**uiMultiplier:** Min: 0 / Max: 65535 / Multiplier (16-bit unsigned value).

**uiDivisor:** Min: 0 / Max: 65535 / Divisor (16-bit unsigned value).

### 7.3.111 ST\_LON\_SNVT\_nv\_type

Genutzt durch: SNVT\_nv\_type

```

TYPE ST_LON_SNVT_nv_type :
STRUCT
  arrType_program_ID : ARRAY [0..7] OF BYTE;
  byType_scope       : BYTE;
  uiType_index       : UINT;
  eType_category     : E_LON_nv_type_category_t;
END_STRUCT
    
```

```

byType_length      : BYTE;
iScaling_factor_a  : INT;
iScaling_factor_b  : INT;
iScaling_factor_c  : INT;
END_STRUCT
END_TYPE

```

**arrType\_program\_ID:** Min: 0 / Max: 255 / Type program ID. Program ID template of the resource file containing the network variable type definition.

**byType\_scope:** Min: 0 / Max: 6 / Type scope (file scope). Scope of the resource file containing the network variable type definition.

**uiType\_index:** Min: 1 / Max: 65535 / Type index (type index). Index within the specified resource file of the network variable type definition.

**eType\_category:** Type category (type category names, see [E\\_LON\\_nv\\_type\\_category t \[▶ 516\]](#)). Type category of the network variable type

**byType\_length:** Min: 1 / Max: 31 / Type length (bytes). Length of the network variable type

**iScaling\_factor\_a:** Min: -32768 / Max: 32767 / Scaling factor a. Scaling multiplier 'a' where ScaledValue =  $a \cdot (10^b) \cdot (\text{RawValue} + c)$

**iScaling\_factor\_b:** Min: -32768 / Max: 32767 / Scaling factor b. Exponent 'b' where ScaledValue =  $a \cdot (10^b) \cdot (\text{RawValue} + c)$

**iScaling\_factor\_c:** Min: -32768 / Max: 32767 / Scaling Factor c. Offset 'c' where ScaledValue =  $a \cdot (10^b) \cdot (\text{RawValue} + c)$

### 7.3.112 ST\_LON\_SNVT\_obj\_request

Genutzt durch: SNVT\_obj\_request

```

TYPE ST_LON_SNVT_obj_request :
STRUCT
  uiObject_id      : UINT;
  eObject_request  : E_LON_object_request_t;
END_STRUCT
END_TYPE

```

**uiObject\_id:** Min: 0 / Max: 65535 / Object ID (object index).

**eObject\_request:** Object request (object request names, see [E\\_LON\\_object\\_request t \[▶ 517\]](#)).

### 7.3.113 ST\_LON\_SNVT\_obj\_status

Genutzt durch: SNVT\_obj\_status

```

TYPE ST_LON_SNVT_obj_status :
STRUCT
  uiObject_id      : UINT;
  bInvalid_id      : BOOL;
  bInvalid_request : BOOL;
  bDisabled        : BOOL;
  bOut_of_limits   : BOOL;
  bOpen_circuit    : BOOL;
  bOut_of_service  : BOOL;
  bMechanical_fault : BOOL;
  bFeedback_failure : BOOL;
  bOver_range      : BOOL;
  bUnder_range     : BOOL;
  bElectrical_fault : BOOL;
  bUnable_to_measure : BOOL;
  bComm_failure    : BOOL;
  bFail_self_test  : BOOL;
  bSelf_test_in_progress : BOOL;
  bLocked_out      : BOOL;
  bManual_control  : BOOL;
  bIn_alarm        : BOOL;
  bIn_override     : BOOL;
  bReport_mask     : BOOL;

```

```

bProgramming_mode      : BOOL;
bProgramming_fail      : BOOL;
bAlarm_notify_disabled : BOOL;
bReset_complete        : BOOL;
byReserved2            : BYTE;
END_STRUCT
END_TYPE

```

**uiObject\_id:** Min: 0 / Max: 65535 / Object ID (object index).

**blInvalid\_id:** Invalid-ID flag (boolean).

**blInvalid\_request:** Invalid-request flag (boolean).

**bDisabled:** Disabled flag (boolean).

**bOut\_of\_limits:** Out-of-limits flag (boolean).

**bOpen\_circuit:** Open-circuit flag (boolean).

**bOut\_of\_service:** Out-of-service flag (boolean).

**bMechanical\_fault:** Mechanical-fault flag (boolean).

**bFeedback\_failure:** Feedback-failure flag (boolean).

**bOver\_range:** Over-range flag (boolean).

**bUnder\_range:** Under-range flag (boolean).

**bElectrical\_fault:** Electrical-fault flag (boolean).

**bUnable\_to\_measure:** Unable-to-measure flag (boolean).

**bComm\_failure:** Communications-failure flag (boolean).

**bFail\_self\_test:** Failed-self-test flag (boolean).

**bSelf\_test\_in\_progress:** Self-test-in-progress flag (boolean).

**bLocked\_out:** Locked-out flag (boolean).

**bManual\_control:** Manual-control flag (boolean).

**bln\_alarm:** Input-alarm flag (boolean).

**bln\_override:** Input-override flag (boolean).

**bReport\_mask:** Report-mask flag (boolean).

**bProgramming\_mode:** Programming-mode flag (boolean).

**bProgramming\_fail:** Programming-fail flag (boolean).

**bAlarm\_notify\_disabled:** Alarm-notify-disabled flag (boolean).

**bReset\_complete:** Reset (boolean).

**byReserved2:** This field is reserved.

### 7.3.114 ST\_LON\_SNVT\_pos\_ctrl

Genutzt durch: SNVT\_pos\_ctrl

```

TYPE ST_LON_SNVT_pos_ctrl :
STRUCT
    uiReceiver_id      : UINT;
    uiController_id    : UINT;
    byController_prio  : BYTE;
    eFunction           : E_LON_cam_func_t;
    eAction             : E_LON_cam_act_t;

```

```

    stValue      : ST_LON_Value;
END_STRUCT
END_TYPE

```

**uiReceiver\_id:** Min: 0 / Max: 65535 / Receiver ID (ID number).

**uiController\_id:** Min: 0 / Max: 65535 / Controller ID (ID number).

**byController\_prio:** Min: 0 / Max: 100 / Controller priority (priority value).

**eFunction:** Camera function (camera function names) (see [E\\_LON\\_cam\\_func\\_t](#) [► 498]).

**eAction:** Camera action (camera action names) (see [E\\_LON\\_cam\\_act\\_t](#) [► 497]).

**stValue:** Function value (see [ST\\_LON\\_Value](#) [► 556]).

### 7.3.115 ST\_LON\_SNVT\_preset

Genutzt durch: SNVT\_preset

```

TYPE ST_LON_SNVT_preset :
STRUCT
    eLearn      : E_LON_learn_mode_t;
    uiSelector  : UINT;
    arrValue    : ARRAY[0..3] OF BYTE;
    uiDay       : UINT;
    uiHour      : UINT;
    uiMinute    : UINT;
    uiSecond    : UINT;
    uiMillisecond : UINT;
END_STRUCT
END_TYPE

```

**eLearn:** Learn mode (learn mode names, see [E\\_LON\\_learn\\_mode\\_t](#) [► 515]).

**uiSelector:** Min: 0 / Max: 65535 / Selector (16-bit unsigned value). The selector is used to choose which preset.

**arrValue:** Value (array of 4 bytes).

**uiDay:** Min: 0 / Max: 65535 / Days (days). The value 65535 represents NULL or unknown elapsed time.

**uiHour:** Min: 0 / Max: 23 / Hours (hours). This field uses a 24-hour value.

**uiMinute:** Min: 0 / Max: 59 / Minutes (minutes).

**uiSecond:** Min: 0 / Max: 59 / Seconds (seconds).

**uiMillisecond:** Min: 0 / Max: 999 / Milliseconds (milliseconds).

### 7.3.116 ST\_LON\_SNVT\_privacyzone

Genutzt durch: SNVT\_privacyzone

```

TYPE ST_LON_SNVT_privacyzone :
STRUCT
    eAction     : E_LON_privacyzone_t;
    byNumber    : BYTE;
    uiCamera_id : UINT;
END_STRUCT
END_TYPE

```

**eAction:** Privacy zone action type (privacy zone action type names, see [E\\_LON\\_privacyzone\\_t](#) [► 520]).

**byNumber:** Min: 0 / Max: 255 / Zone number (zone number).

**uiCamera\_id:** Min: 0 / Max: 65535 / Camera ID (ID number).



### 7.3.117 ST\_LON\_SNVT\_ptz

Genutzt durch: SNVT\_ptz

```

TYPE ST_LON_SNVT_ptz :
STRUCT
  ePan_dir      : E_LON_pan_dir_t;
  rPan_speed    : REAL;
  eTilt_dir     : E_LON_tilt_dir_t;
  rTilt_speed   : REAL;
  eZoom         : E_LON_zoom_t;
  rZoom_speed   : REAL;
END_STRUCT
END_TYPE
    
```

**ePan\_dir:** Pan direction (pan direction names, see [E\\_LON\\_pan\\_dir\\_t \[► 519\]](#)).

**rPan\_speed:** Min: 0 / Max: 100 / Pan speed (% of full level).

**eTilt\_dir:** Tilt direction (tilt direction names, see [E\\_LON\\_tilt\\_dir\\_t \[► 534\]](#)).

**rTilt\_speed:** Min: 0 / Max: 100 / Tilt speed (% of full level).

**eZoom:** Zoom direction (zoom direction names, see [E\\_LON\\_zoom\\_t \[► 535\]](#)).

**rZoom\_speed:** Min: 0 / Max: 100 / Zoom speed (% of full level).

### 7.3.118 ST\_LON\_SNVT\_pump\_sensor

Genutzt durch: SNVT\_pump\_sensor

```

TYPE ST_LON_SNVT_pump_sensor :
STRUCT
  rRotational_speed      : REAL;
  rBody_temperature      : REAL;
  rMotor_external_temperature : REAL;
  rMotor_internal_temperature : REAL;
  eMotor_overloaded      : E_LON_boolean_t;
  eOil_level_low         : E_LON_boolean_t;
  ePhase_imbalance_detected : E_LON_boolean_t;
  rCurrent_usage         : REAL;
  rPower_usage           : REAL;
  eTemperature_control   : E_LON_unit_temp_t;
  eElectromagnetic_brake_active : E_LON_boolean_t;
  eFriction_brake_active : E_LON_boolean_t;
  eGas_brake_active      : E_LON_boolean_t;
END_STRUCT
END_TYPE
    
```

**rRotational\_speed:** Min: 0 / Max: 6553.5 / Rotational speed.

**rBody\_temperature:** Min: -274 / Max: 6279.5 / Body temperature.

**rMotor\_external\_temperature:** Min: -274 / Max: 6279.5 / Motor external temp.

**rMotor\_internal\_temperature:** Min: -274 / Max: 6279.5 / Motor internal temp.

**eMotor\_overloaded:** Motor overloaded (Boolean, see [E\\_LON\\_boolean\\_t \[► 497\]](#)).

**eOil\_level\_low:** Oil level low (boolean).

**ePhase\_imbalance\_detected:** Phase imbalance (boolean).

**rCurrent\_usage:** Min: -3276.8 / Max: 3276.7 / Current usage.

**rPower\_usage:** Min: 0 / Max: 6553.5 / Power usage.

**eTemperature\_control:** Pump body temp control status (temperature control status names, see [E\\_LON\\_unit\\_temp\\_t \[► 534\]](#)).

**eElectromagnetic\_brake\_active:** Electromagnetic brake active (boolean).

**eFriction\_brake\_active:** Friction brake active (boolean).

**eGas\_brake\_active:** Gas brake active (boolean).

### 7.3.119 ST\_LON\_SNVT\_pumpset\_mn

Genutzt durch: SNVT\_pumpset\_mn

```

TYPE ST_LON_SNVT_pumpset_mn :
STRUCT
  eMain_pump          : E_LON_motor_state_t;
  eBooster_pump      : E_LON_motor_state_t;
  ePriority_level     : E_LON_Priority_level_t;
  eProcess_ready     : E_LON_boolean_t;
  eEmergency_stop_activated : E_LON_boolean_t;
  eMain_pump_drive_enabled : E_LON_boolean_t;
  eBooster_pump_drive_enabled : E_LON_boolean_t;
  eMaintenance_required : E_LON_boolean_t;
END_STRUCT
END_TYPE

```

**eMain\_pump:** Main pump state (motor state names, see [E\\_LON\\_motor\\_state\\_t](#) [► 516]).

**eBooster\_pump:** Booster pump state (motor state names).

**ePriority\_level:** Priority level (priority level names, see [E\\_LON\\_priority\\_level\\_t](#) [► 519]).

**eProcess\_ready:** Process ready (boolean).

**eEmergency\_stop\_activated:** Emergency stop (Boolean, see [E\\_LON\\_boolean\\_t](#) [► 497]).

**eMain\_pump\_drive\_enabled:** Main pump enabled (boolean).

**eBooster\_pump\_drive\_enabled:** Booster pump enabled (boolean).

**eMaintenance\_required:** Maintenance required (boolean).

### 7.3.120 ST\_LON\_SNVT\_pumpset\_sn

Genutzt durch: SNVT\_pumpset\_sn

```

TYPE ST_LON_SNVT_pumpset_sn :
STRUCT
  uiTotal_dilution_flow : UINT;
  rExhaust_temperature  : REAL;
  rExhaust_pressure     : REAL;
  rShaft_seal_purge_pressure : REAL;
  rInlet_vacuum         : REAL;
  rSupply_voltage       : REAL;
  uiCoolant_flow        : UINT;
  eCoolant_flow_low     : E_LON_boolean_t;
  eDilution_active     : E_LON_boolean_t;
  eBallast_dilution_active : E_LON_boolean_t;
  eInlet_purge_dilution_active : E_LON_boolean_t;
  eExhaust_dilution_active : E_LON_boolean_t;
  eDilution_flow_Out_of_range : E_LON_boolean_t;
  ePower_supply_on      : E_LON_boolean_t;
END_STRUCT
END_TYPE

```

**uiTotal\_dilution\_flow:** Min: 0 / Max: 65535 / Dilution gas flow.

**rExhaust\_temperature:** Min: 274 / Max: 6279.5 / Exhaust line external temperature.

**rExhaust\_pressure:** Min: -3276.8 / Max: 3276.7 / Exhaust line pressure.

**rShaft\_seal\_purge\_pressure:** Min: -3276.8 / Max: 3276.7 / Shaft seal purge pressure.

**rInlet\_vacuum:** Min: -3.40E+38 / Max: 3.40E+38 / Process gas inlet pressure.

**rSupply\_voltage:** Min: -3276.8 / Max: 3276.7 / Pumpset power supply voltage.

**uiCoolant\_flow:** Min: 0 / Max: 65535 / Total coolant flow.

**eCoolant\_flow\_low:** Coolant flow too low (Boolean, see [E\\_LON\\_boolean\\_t](#) [► 497]).

**eDilution\_active:** Coolant flow too low (boolean).

**eBallast\_dilution\_active:** Dilution gas being used as ballast (boolean).

**eInlet\_purge\_dilution\_active:** Dilution gas being used to purge process gas (boolean).

**eExhaust\_dilution\_active:** Dilution gas being used to dilute exhaust (boolean).

**eDilution\_flow\_Out\_of\_range:** Dilution gas flow outside normal range (boolean).

**ePower\_supply\_on:** Dilution gas flow outside normal range (boolean).

### 7.3.121 ST\_LON\_SNVT\_rac\_ctrl

Genutzt durch: SNVT\_rac\_ctrl

```

TYPE ST_LON_SNVT_rac_ctrl :
STRUCT
  byAudio_line   : BYTE;
  bDuplex_full   : BOOL;
  bDest_p2p      : BOOL;
  byReserved     : BYTE;
  eAudio_type    : E_LON_rail_audio_type_t;
  stAddr_init    : ST_LON_addr_init;
  stAddr_talk    : ST_LON_addr_talk;
  stAddr_dest    : ST_LON_addr_dest;
END_STRUCT
END_TYPE
    
```

**byAudio\_line:** Min: 0 / Max: 7

**bDuplex\_full:**

**bDest\_p2p:**

**byReserved:**

**eAudio\_type:** (see [E\\_LON\\_rail\\_audio\\_type\\_t](#) [► 521])

**stAddr\_init:** (see [ST\\_LON\\_addr\\_init](#) [► 557])

**stAddr\_talk:** (see [ST\\_LON\\_addr\\_talk](#) [► 557])

**stAddr\_dest:** (see [ST\\_LON\\_addr\\_dest](#) [► 556])

### 7.3.122 ST\_LON\_SNVT\_rac\_req

Genutzt durch: SNVT\_rac\_req

```

TYPE ST_LON_SNVT_rac_req :
STRUCT
  bDest_def      : BOOL;
  bDest_p2p      : BOOL;
  byReserved     : BYTE;
  eAudio_type    : E_LON_rail_audio_type_t;
  stAddr_init    : ST_LON_addr_init;
  stAddr_dest    : ST_LON_addr_dest;
END_STRUCT
END_TYPE
    
```

**bDest\_def:**

**bDest\_p2p:**

**byReserved:**

**eAudio\_type:** (see [E\\_LON\\_rail\\_audio\\_type\\_t](#) [► 521])

**stAddr\_init:** (see [ST\\_LON\\_addr\\_init](#) [► 557])

**stAddr\_dest:** (see [ST\\_LON\\_addr\\_dest](#) [► 556])

### 7.3.123 ST\_LON\_SNVT\_reg\_val

Genutzt durch: SNVT\_rac\_val

```
TYPE ST_LON_SNVT_reg_val :
STRUCT
  diRaw      : DINT;
  eUnit      : E_LON_reg_val_unit_t;
  byNr_decimals : BYTE;
END_STRUCT
END_TYPE
```

**diRaw:** Raw value.

**eUnit:** Unit code (defines unit of measure, see [E\\_LON\\_reg\\_val\\_unit\\_t](#) [► 522]).

**byNr\_decimals:** Number of decimals (digits to right of decimal point).

### 7.3.124 ST\_LON\_SNVT\_reg\_val\_ts

Genutzt durch: SNVT\_rac\_val\_ts

```
TYPE ST_LON_SNVT_reg_val_ts :
STRUCT
  diRaw      : DINT;
  eUnit      : E_LON_reg_val_unit_t;
  byNr_decimals : BYTE;
  byStatus    : BYTE;
  bReg_state  : BOOL;
  uiYear      : UINT;
  uiMonth     : UINT;
  uiDay       : UINT;
  uiHour      : UINT;
  uiMinute    : UINT;
  uiSecond    : UINT;
END_STRUCT
END_TYPE
```

**diRaw:** Min: -2147483648 / Max: 2147483647 / Raw value.

**eUnit:** Unit code (unit names, see [E\\_LON\\_reg\\_val\\_unit\\_t](#) [► 522]).

**byNr\_decimals:** Min: 0 / Max: 7 / Number of decimals (digits to right of decimal point).

**byStatus:** Min: 0 / Max: 15 / Status (status or error during measuring period).

**bReg\_state:** Activation state (activation state of register).

**uiYear:** Min: -1 / Max: 3000 / Year (years). Zero (0) means year not specified. Minus one (-1) represents NULL date.

**uiMonth:** Min: 0 / Max: 12 / Month (months). Zero (0) means month not specified.

**uiDay:** Min: 0 / Max: 31 / Day (days). Zero (0) means day not specified.

**uiHour:** Min: 0 / Max: 23 / Hour (hours). This field uses a 24-hour value.

**uiMinute:** Min: 0 / Max: 59 / Minute (minutes).

**uiSecond:** Min: 0 / Max: 59 / Second (seconds).

### 7.3.125 ST\_LON\_SNVT\_sblnd\_state

Genutzt durch: SNVT\_sblnd\_state

```
TYPE ST_LON_SNVT_sblnd_state :
STRUCT
  stPos      : ST_LON_SNVT_setting;
  eCmd_source : E_LON_sblnd_cmd_source_t;
END_STRUCT
```

```
eError_code : E_LON_sblnd_error_t;
END_STRUCT
END_TYPE
```

**stPos:** (see [ST\\_LON\\_SNVT\\_setting](#) [▶ 586])

**eCmd\_source:** (see [E\\_LON\\_sblnd\\_cmd\\_source\\_t](#) [▶ 524])

**eError\_code:** (see [E\\_LON\\_sblnd\\_error\\_t](#) [▶ 526])

### 7.3.126 ST\_LON\_SNVT\_scene

Genutzt durch: SNVT\_scene

```
TYPE ST_LON_SNVT_scene :
STRUCT
eFunction : E_LON_Scene_t;
byScene_number : BYTE;
END_STRUCT
END_TYPE
```

**eFunction:** Scene control function (scene control function names, see [E\\_LON\\_scene\\_t](#) [▶ 527]).

**byScene\_number:** Min: 0 / Max: 255 / Scene number.

### 7.3.127 ST\_LON\_SNVT\_scene\_cfg

Genutzt durch: SNVT\_scene\_cfg

```
TYPE ST_LON_SNVT_scene_cfg :
STRUCT
eFunction : E_LON_Scene_config_t;
byScene_number : BYTE;
rSetting : REAL;
rRotation : REAL;
rFade_time : REAL;
rDelay_time : REAL;
scene_priority : BYTE;
END_STRUCT
END_TYPE
```

**eFunction:** Scene configuration function (scene configuration function names, see [E\\_LON\\_scene\\_config\\_t](#) [▶ 526]).

**byScene\_number:** Min: 0 / Max: 255 / Scene number.

**rSetting:** Min: 0 / Max: 100.0 Invalid: 255 / Scene setting level (% of full level).

**rRotation:** Min: -359,98 / Max: 360,00 / Scene rotation angle (degrees).

**rFade\_time:** Min: 0 / Max: 6553.5 / Scene fade time (seconds).

**rDelay\_time:** Min: 0 / Max: 6553.5 / Scene delay time (seconds).

**scene\_priority:** Min: 0 / Max: 255 / scene\_priority

### 7.3.128 ST\_LON\_SNVT\_setting

Genutzt durch: SNVT\_setting

```
TYPE ST_LON_SNVT_setting :
STRUCT
eFunction : E_LON_setting_t;
rSetting : REAL;
rRotation : REAL;
END_STRUCT
END_TYPE
```

**eFunction:** Setting control function (setting control function names, see [E\\_LON\\_setting\\_t](#) [▶ 530]).

**rSetting:** Min: 0 / Max: 100 / Scene setting level (% of full level).

**rRotation:** Min: -359.98 / Max: 360.00 / Rotation angle (degrees).

### 7.3.129 ST\_LON\_SNVT\_str\_int

Genutzt durch: SNVT\_str\_int

```
TYPE ST_LON_SNVT_str_int :
STRUCT
  byChar_set      : BYTE;
  arrWide_char    : ARRAY [0..14] OF UINT;
END_STRUCT
END_TYPE
```

**byChar\_set:** Min: 0 / Max: 255 / Locale code (code value).

**arrWide\_char:** Min: 0 / Max: 65535 / Wide character string (array of 15 wide characters).

### 7.3.130 ST\_LON\_SNVT\_switch

Genutzt durch: SNVT\_switch

```
TYPE ST_LON_SNVT_switch :
STRUCT
  rValue          : REAL;
  siState         : SINT;
END_STRUCT
END_TYPE
```

**rValue:** Min: 0 / Max: 100 / Value (% of full level).

**siState:** Min: -1 / Max: 1 / State (state code). This field can either be -1 (NULL), 0 (OFF), or 1 (ON).

### 7.3.131 ST\_LON\_SNVT\_switch\_2

Genutzt durch: SNVT\_switch\_2

```
TYPE ST_LON_SNVT_switch_2 :
STRUCT
  eState          : E_LON_switch_state_t;
  stSetting       : ST_LON_setting;
  byScene_number  : BYTE;
END_STRUCT
END_TYPE
```

**eState:** Switch state. Switch state; maybe a state of the switch or other switch properties such as scene, occupancy state, and level multiplier (see [E\\_LON\\_switch\\_state\\_t](#) [▶ 531]).

**stSetting:** Switch setting. Sets or reports the level, change, or angle for a switch (see [ST\\_LON\\_setting](#) [▶ 559]).

**byScene\_number:** Min: 1 / Max: 255 / Scene number. Scene number that is applied based on the function specified in the state field.

### 7.3.132 ST\_LON\_SNVT\_temp\_setpt

Genutzt durch: SNVT\_temp\_setpt

```
TYPE ST_LON_SNVT_temp_setpt :
STRUCT
  rOccupied_cool  : REAL;
  rStandby_cool   : REAL;
  rUnoccupied_cool : REAL;
  rOccupied_heat  : REAL;
  rStandby_heat   : REAL;
  rUnoccupied_heat : REAL;
END_STRUCT
END_TYPE
```

**rOccupied\_cool:** Min: -273,17 / Max: 237,67 / Occupied cooling setpoint (degrees Celsius).

**rStandby\_cool:** Min: -273,17 / Max: 237,67 / Standby cooling setpoint (degrees Celsius).

**rUnoccupied\_cool:** Min: -273,17 / Max: 237,67 / Unoccupied cooling setpoint (degrees Celsius).

**rOccupied\_heat:** Min: -273,17 / Max: 237,67 / Occupied heating setpoint (degrees Celsius).

**rStandby\_heat:** Min: -273,17 / Max: 237,67 / Standby heating setpoint (degrees Celsius).

**rUnoccupied\_heat:** Min: -273,17 / Max: 237,67 / Unoccupied heating setpoint (degrees Celsius).

### 7.3.133 ST\_LON\_SNVT\_time\_zone

Genutzt durch: SNVT\_time\_zone

```

TYPE ST_LON_SNVT_time_zone :
STRUCT
  diSecond_time_offset      : DINT;
  eType_of_description      : E_LON_calendar_type_t;
  byHour_of_start_DST       : BYTE;
  byMinute_of_start_DST     : BYTE;
  bySecond_of_start_DST     : BYTE;
  stStart_DST               : ST_LON_start_DST;
  byHour_of_end_DST         : BYTE;
  byMinute_of_end_DST       : BYTE;
  bySecond_of_end_DST       : BYTE;
  stEnd_DST                 : ST_LON_end_DST;
END_STRUCT
END_TYPE
    
```

**diSecond\_time\_offset:** Min: -86400 / Max: 86400 / Offset from GMT (seconds). West direction is negative offset

**eType\_of\_description:** Calendar type (calendar type names, see [E\\_LON\\_calendar\\_type\\_t](#) [▶ 497]).

**byHour\_of\_start\_DST:** Min: 0 / Max: 23 / DST start hour (hours).

**byMinute\_of\_start\_DST:** Min: 0 / Max: 59 / DST start minute (minutes).

**bySecond\_of\_start\_DST:** Min: 0 / Max: 59 / DST start second (seconds).

**stStart\_DST:** DST start day (day descriptor). Daylight savings time start day (see [ST\\_LON\\_start\\_DST](#) [▶ 560]).

**byHour\_of\_end\_DST:** Min: 0 / Max: 23 / DST end hour (hours).

**byMinute\_of\_end\_DST:** Min: 0 / Max: 59 / DST end minute (minutes).

**bySecond\_of\_end\_DST:** Min: 0 / Max: 59 / DST end second (seconds).

**stEnd\_DST:** DST end day (day descriptor). Daylight savings time end day (see [ST\\_LON\\_end\\_DST](#) [▶ 559]).

### 7.3.134 ST\_LON\_SNVT\_tod\_event

Genutzt durch: SNVT\_tod\_event

```

TYPE ST_LON_SNVT_tod_event :
STRUCT
  eCurrent_state            : E_LON_occup_t;
  eNext_state               : E_LON_occup_t;
  uiTime_to_next_state     : UINT;
END_STRUCT
END_TYPE
    
```

**eCurrent\_state:** Occupancy, current (occupancy code names, see [E\\_LON\\_occup\\_t](#) [▶ 518]).

**eNext\_state:** Occupancy, next (occupancy code names).

**uiTime\_to\_next\_state:** Min: 0 / Max: 65535 / Time to next state (minutes).

### 7.3.135 ST\_LON\_SNVT\_trans\_table

Genutzt durch: SNVT\_trans\_table

```
TYPE ST_LON_SNVT_trans_table :
STRUCT
  arrPoint      : ARRAY [0..6] OF REAL;
  byInterp_pts_0_to_1 : BYTE;
  byInterp_pts_1_to_2 : BYTE;
  byInterp_pts_2_to_3 : BYTE;
  byInterp_pts_3_to_4 : BYTE;
  byInterp_pts_4_to_5 : BYTE;
  byInterp_pts_5_to_6 : BYTE;
  byInterp_pts_6_to_0 : BYTE;
END_STRUCT
END_TYPE
```

**arrPoint:** Points (array of 7 points).

**byInterp\_pts\_0\_to\_1:** Min: 0 / Max: 1 / Interpolation for point 0 to point 1 (interpolation method code).

**byInterp\_pts\_1\_to\_2:** Min: 0 / Max: 1 / Interpolation for point 1 to point 2 (interpolation method code).

**byInterp\_pts\_2\_to\_3:** Min: 0 / Max: 1 / Interpolation for point 2 to point 3 (interpolation method code).

**byInterp\_pts\_3\_to\_4:** Min: 0 / Max: 1 / Interpolation for point 3 to point 4 (interpolation method code).

**byInterp\_pts\_4\_to\_5:** Min: 0 / Max: 1 / Interpolation for point 4 to point 5 (interpolation method code).

**byInterp\_pts\_5\_to\_6:** Min: 0 / Max: 1 / Interpolation for point 5 to point 6 (interpolation method code).

**byInterp\_pts\_6\_to\_0:** Min: 0 / Max: 1 / Interpolation for point 6 to point 0 (interpolation method code). This field is used when multiple interpolation tables are linked.

### 7.3.136 ST\_LON\_SNVT\_zerospans

Genutzt durch: SNVT\_zerospans

```
TYPE ST_LON_SNVT_zerospans :
STRUCT
  rZero : REAL;
  rSpan : REAL;
END_STRUCT
END_TYPE
```

**rZero:** Min: -163.840 / Max: 163.835 / Zero-term (16-bit signed value).

**rSpan:** Min: 0.0 / Max: 32.7675 / Span-factor (16-bit unsigned value).

### 7.3.137 str\_AddressTable

Adress Tabelle

```
TYPE str_AddressTable :
STRUCT
  bType      : BOOL;
  Node       : USINT;
  bDomain    : BOOL;
  Member     : USINT;
  RPT_Timer  : USINT;
  Retry      : USINT;
  RCV_Timer  : USINT;
  Tx_Timer   : USINT;
  Group      : BYTE;
END_STRUCT
END_TYPE
```

**bType:** Type

**Node:** Node

**bDomain:** Domain



**Member:** Member

**RPT\_Timer:** RPT\_Timer

**Retry:** Retry

**RCV\_Timer:** RCV\_Timer

**Tx\_Timer:** Tx\_Timer

**Group:** Group

## 7.4 Globale\_Variablen\_LON

Defaultwerte für alle Sendebausteine

```
VAR_GLOBAL CONSTANT
  tMinSendTimeDefault := t#1000ms,
  tMaxSendTimeDefault := t#0s,
  bAutoDefault        := FALSE,
  bSendInitDefault    := FALSE,
END_VAR
```

tMinSendTimeDefault: Defaultwert für alle Sendebausteine. Gilt für den [Automatikmodus \[▶ 612\]](#). Der Baustein sendet frühestens nach dieser Zeit. Damit kann die Anzahl der Telegramme begrenzt werden, wenn sich die Werte sehr schnell ändern.

tMaxSendTimeDefault: Defaultwert für alle Sendebausteine. Gilt für den [Automatikmodus \[▶ 612\]](#). Nach Ablauf dieser Zeit wird der Wert gesendet, unabhängig von einer möglichen Wertänderung.

bAutoDefault: Defaultwert für alle Sendebausteine. Einschalten des [Automatikmodus \[▶ 612\]](#)

bSendInitDefault: Defaultwert für alle Sendebausteine. Nach dem Einschalten werden die Daten einmal gesendet.

## 7.5 Fehlercodes

Fehlermeldungen der Bibliothek



Der NV Index in der SPS wird beim Senden nicht mit dem NV Index (Spalte Id) in der KS2000 verglichen. Stimmen beide Indizes nicht überein, können falsche/ungültige Werte gesendet werden.

Senden ohne Binding gibt keine Fehlermeldung aus.

| Wert (hex) | Wert (dez) | Wert (enum)             | Beschreibung   |
|------------|------------|-------------------------|--|
| 0x0000     | 0          | eLON_no_Error           | Es steht kein Fehler an.   |
| 0x0001     | 1          | eLON_Value_out_of_range | Die Eingangsvariable "Value" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. "Value" kann in verschiedenen Formaten vorliegen und besitzt dann einen entsprechenden Präfix (z.B. LREAL = IrValue).   |
| 0x0002     | 2          | eLON_Terminal_not_ready | Der Baustein "FB_LON_KL6401" durchläuft beim Starten der SPS eine Initialisierungsschrittfolge (Klemmentyp abfragen, Firmware abfragen u.s.w.). So lange diese nicht fertig gemeldet hat, wird diese Meldung ausgegeben. Steht der Fehler nach einem SPS Reset an, muss der Controller einmal spannungslos gemacht werden. |

| Wert (hex) | Wert (dez) | Wert (enum)                         | Beschreibung   |
|------------|------------|-------------------------------------|--|
| 0x0003     | 3          | eLON_Wrong_SNVT_Typ                 | Der empfangene SNVT Typ entspricht nicht dem SNVT Typ des angesprochenen NV Index (Eingangsvariable "wld").  |
| 0x0004     | 4          | eLON_Wrong_wNVIndex                 | Falscher NV Index.   |
| 0x0005     | 5          | eKL6401_Wrong_Terminal              | Es wurde keine KL6401 erkannt.   |
| 0x0006     | 6          | eKL6401_Error                       | Der Baustein "FB_LON_KL6401" hat einen Fehler. Der Fehlercode steht am Ausgang "dwErrorKL" an.   |
| 0x0007     | 7          | eKL6401_Terminal_is_not_initialized | Die Klemme ist nicht initialisiert. Üblicherweise gibt es bei dieser Meldung keine Verbindung zur Klemme. Klemme im System Manager mit den Variablen verknüpft? Klemme falsch gesteckt? Alles bereinigen, alles Übersetzen und im System Manager neu eingelesen? |
| 0x0032     | 50         | eLON_L_star_Out_of_range            | SNVT 70 / Die Eingangsvariable "stValue.L_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0033     | 51         | eLON_A_star_Out_of_range            | SNVT 70 / Die Eingangsvariable "stValue.A_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0034     | 52         | eLON_B_star_Out_of_range            | SNVT 70 / Die Eingangsvariable "stValue.B_star" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0037     | 55         | eLON_eRequest_Out_of_range          | SNVT 73 / Die Eingangsvariable "stValue.eRequest" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0042     | 66         | eLON_wYear_Out_of_range             | SNVT 084 / 088 / Die Eingangsvariable "stValue.wYear" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0043     | 67         | eLON_wMonth_Out_of_range            | SNVT 084 / 088 / Die Eingangsvariable "stValue.wMonth" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0044     | 68         | eLON_wDay_Out_of_range              | eLON_wDay_Out_of_range: SNVT 084 / 088 / Die Eingangsvariable "stValue.wDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0045     | 69         | eLON_wHour_Out_of_range             | SNVT 084 / 088 / Die Eingangsvariable "stValue.wHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0046     | 70         | eLON_wMinute_Out_of_range           | SNVT 084 / 088 / Die Eingangsvariable "stValue.wMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0047     | 71         | eLON_wSecond_Out_of_range           | SNVT 084 / 088 / Die Eingangsvariable "stValue.wSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |

| Wert (hex) | Wert (dez) | Wert (enum)                    | Beschreibung   |
|------------|------------|--------------------------------|--|
| 0x0048     | 72         | eLON_wMillisecond_Out_of_range | SNVT 088 / Die Eingangsvariable "stValue.wMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0050     | 80         | eLON_rZero_Out_of_range        | SNVT 085 / Die Eingangsvariable "stValue.rZero" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x0051     | 81         | eLON_rSpan_Out_of_range        | SNVT 085 / Die Eingangsvariable "stValue.rSpan" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x0055     | 85         | eLON_arrValue01_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[1]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0056     | 86         | eLON_arrValue02_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[2]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0057     | 87         | eLON_arrValue03_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[3]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0058     | 88         | eLON_arrValue04_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[4]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0059     | 89         | eLON_arrValue05_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[5]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x005A     | 90         | eLON_arrValue06_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[6]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x005B     | 91         | eLON_arrValue07_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[7]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x005C     | 92         | eLON_arrValue08_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[8]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x005D     | 93         | eLON_arrValue09_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[9]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0064     | 100        | eLON_arrValue10_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[10]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0065     | 101        | eLON_arrValue11_Out_of_range   | SNVT 086 / Die Eingangsvariable "arrValue[11]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |

| Wert (hex) | Wert (dez) | Wert (enum)                  | Beschreibung   |
|------------|------------|------------------------------|--|
| 0x0066     | 102        | eLON_arrValue12_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[12]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0067     | 103        | eLON_arrValue13_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[13]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0068     | 104        | eLON_arrValue14_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[14]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0069     | 105        | eLON_arrValue15_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[15]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x006A     | 106        | eLON_arrValue16_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[16]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x006B     | 107        | eLON_arrValue17_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[17]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x006C     | 108        | eLON_arrValue18_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[18]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x006D     | 109        | eLON_arrValue19_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[19]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0073     | 115        | eLON_arrValue20_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[20]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0074     | 116        | eLON_arrValue21_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[21]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0075     | 117        | eLON_arrValue22_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[22]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0076     | 118        | eLON_arrValue23_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[23]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0077     | 119        | eLON_arrValue24_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[24]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0078     | 120        | eLON_arrValue25_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[25]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |

| Wert (hex) | Wert (dez) | Wert (enum)                  | Beschreibung   |
|------------|------------|------------------------------|--|
| 0x0079     | 121        | eLON_arrValue26_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[26]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x007A     | 122        | eLON_arrValue27_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[27]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x007B     | 123        | eLON_arrValue28_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[28]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x007C     | 124        | eLON_arrValue29_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[29]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0082     | 130        | eLON_arrValue30_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[30]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0083     | 131        | eLON_arrValue31_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[31]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0084     | 132        | eLON_arrValue32_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[32]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0085     | 133        | eLON_arrValue33_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[33]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0086     | 134        | eLON_arrValue34_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[34]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0087     | 135        | eLON_arrValue35_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[35]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0088     | 136        | eLON_arrValue36_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[36]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0089     | 137        | eLON_arrValue37_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[37]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x008A     | 138        | eLON_arrValue38_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[38]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x008B     | 139        | eLON_arrValue39_Out_of_range | SNVT 086 / Die Eingangsvariable "arrValue[39]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |

| Wert (hex) | Wert (dez) | Wert (enum)                        | Beschreibung  |
|------------|------------|------------------------------------|---|
| 0x008C     | 140        | eLON_arrValue40_Out_of_range       | SNVT 086 / Die Eingangsvariable "arrValue[40]" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x0091     | 145        | eLON_087uiDay_Out_of_range         | SNVT 087 / Die Eingangsvariable "stValue.uiDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x0092     | 146        | eLON_087uiHour_Out_of_range        | SNVT 087 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x0093     | 147        | eLON_087uiMinute_Out_of_range      | SNVT 087 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x0094     | 148        | eLON_087uiSecond_Out_of_range      | SNVT 087 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x0095     | 149        | eLON_087uiMillisecond_Out_of_range | SNVT 087 / Die Eingangsvariable "stValue.uiMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x009B     | 155        | eLON_ePriority_level_Out_of_range  | SNVT 088 / Die Eingangsvariable "stValue.ePriority_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x009C     | 156        | eLON_eAlarm_type_Out_of_range      | SNVT 088 / Die Eingangsvariable "stValue.eAlarm_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x00A0     | 160        | eLON_Currency_Out_of_range         | SNVT 089 / Die Eingangsvariable "stValue.Currency" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x00A5     | 165        | eLON_diRw_ptr_Out_of_range         | SNVT 090 / Die Eingangsvariable "stValue.diRw_ptr" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x00AA     | 170        | eLON_Object_request_Out_of_range   | SNVT 092 / Die Eingangsvariable "stValue.Object_request" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x00AF     | 175        | eLON_094eLearn_Out_of_range        | SNVT 094 / Die Eingangsvariable "stValue.eLearn" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x00B0     | 176        | eLON_094uiHour_Out_of_range        | SNVT 094 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x00B1     | 177        | eLON_094uiMinute_Out_of_range      | SNVT 094 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |



| Wert (hex) | Wert (dez) | Wert (enum)                           | Beschreibung  |
|------------|------------|---------------------------------------|---|
| 0x00B2     | 178        | eLON_094uiSecond_Out_of_range         | SNVT 094 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x00B3     | 179        | eLON_094uiMillisecond_Out_of_range    | SNVT 094 / Die Eingangsvariable "stValue.uiMillisecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x00B9     | 185        | eLON_095rValue_Out_of_range           | SNVT 095 / Die Eingangsvariable "stValue.rValue" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x00BA     | 186        | eLON_095siState_Out_of_range          | SNVT 095 / Die Eingangsvariable "stValue.siState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x00BE     | 190        | eLON_byInterp_pts_0_to_1_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_0_to_1" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00BF     | 191        | eLON_byInterp_pts_1_to_2_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_1_to_2" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C0     | 192        | eLON_byInterp_pts_2_to_3_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_2_to_3" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C1     | 193        | eLON_byInterp_pts_3_to_4_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_3_to_4" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C2     | 194        | eLON_byInterp_pts_4_to_5_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_4_to_5" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C3     | 195        | eLON_byInterp_pts_5_to_6_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_5_to_6" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C4     | 196        | eLON_byInterp_pts_6_to_0_Out_of_range | SNVT 096 / Die Eingangsvariable "stValue.byInterp_pts_6_to_0" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00C8     | 200        | eLON_rOccupied_cool_Out_of_range      | SNVT 106 / Die Eingangsvariable "stValue.rOccupied_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.      |
| 0x00C9     | 201        | eLON_rStandby_cool_Out_of_range       | SNVT 106 / Die Eingangsvariable "stValue.rStandby_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x00CA     | 202        | eLON_rUnoccupied_cool_Out_of_range    | SNVT 106 / Die Eingangsvariable "stValue.rUnoccupied_cool" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.    |

| Wert (hex) | Wert (dez) | Wert (enum)                              | Beschreibung   |
|------------|------------|--|--|
| 0x00CB     | 203        | eLON_rOccupied_heat_Out_of_range         | SNVT 106 / Die Eingangsvariable "stValue.rOccupied_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x00CC     | 204        | eLON_rStandby_heat_Out_of_range          | SNVT 106 / Die Eingangsvariable "stValue.rStandby_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x00CD     | 205        | eLON_rUnoccupied_heat_Out_of_range       | SNVT 106 / Die Eingangsvariable "stValue.rUnoccupied_heat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x00D2     | 210        | eLON_111rPercent_Out_of_range            | SNVT 111 / Die Eingangsvariable "stValue.rPercent" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x00D3     | 211        | eLON_111eState_Out_of_range              | SNVT 111 / Die Eingangsvariable "stValue.eState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x00D7     | 215        | eLON_eMode_Out_of_range                  | SNVT 112 / Die Eingangsvariable "stValue.eMode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                  |
| 0x00D8     | 216        | eLON_rHeat_output_primary_Out_of_range   | SNVT 112 / Die Eingangsvariable "stValue.rHeat_output_primary" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x00D9     | 217        | eLON_rHeat_output_secondary_Out_of_range | SNVT 112 / Die Eingangsvariable "stValue.rHeat_output_secondary" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00DA     | 218        | eLON_rCool_output_Out_of_range           | SNVT 112 / Die Eingangsvariable "stValue.rCool_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x00DB     | 219        | eLON_rEcon_output_Out_of_range           | SNVT 112 / Die Eingangsvariable "stValue.rEcon_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x00DC     | 220        | eLON_rFan_output_Out_of_range            | SNVT 112 / Die Eingangsvariable "stValue.rFan_output" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x00E1     | 225        | eLON_115eFunction_Out_of_range           | SNVT 115 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x00E2     | 226        | eLON_eFunction_Out_of_range              | SNVT 116 / 117 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x00E3     | 227        | eLON_rSetting_Out_of_range               | SNVT 116 / 117 / Die Eingangsvariable "stValue.rSetting" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |



| Wert (hex) | Wert (dez) | Wert (enum)                             | Beschreibung  |
|------------|------------|---|---|
| 0x00E4     | 228        | eLON_rRotation_Out_of_range             | SNVT 116 / 117 / Die Eingangsvariable "stValue.rRotation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x00E5     | 229        | eLON_rFade_time_Out_of_range            | SNVT 116 / Die Eingangsvariable "stValue.rFade_time" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x00E6     | 230        | eLON_rDelay_time_Out_of_range           | SNVT 116 / Die Eingangsvariable "stValue.rDelay_time" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x00EB     | 235        | eLON_eChlr_run_mode_Out_of_range        | SNVT 127 / Die Eingangsvariable "stValue.eChlr_run_mode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x00EC     | 236        | eLON_echlr_op_mode_Out_of_range         | SNVT 127 / Die Eingangsvariable "stValue.echlr_op_mode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x00F0     | 240        | eLON_eNext_state_Out_of_range           | SNVT 128 / Die Eingangsvariable "stValue.eNext_state" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x00F1     | 241        | eLON_eCurrent_state_Out_of_range        | SNVT 128 / Die Eingangsvariable "stValue.eCurrent_state" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x00F5     | 245        | eLON_diSecond_time_offset_Out_of_range  | SNVT 134 / Die Eingangsvariable "stValue.diSecond_time_offset" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x00F6     | 246        | eLON_eType_of_description_Out_of_range  | SNVT 134 / Die Eingangsvariable "stValue.eType_of_description" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x00FA     | 250        | eLON_byHour_of_start_DST_Out_of_range   | SNVT 134 / Die Eingangsvariable "stValue.byHour_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x00FB     | 251        | eLON_byMinute_of_start_DST_Out_of_range | SNVT 134 / Die Eingangsvariable "stValue.byMinute_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x00FC     | 252        | eLON_bySecond_of_start_DST_Out_of_range | SNVT 134 / Die Eingangsvariable "stValue.bySecond_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0104     | 260        | eLON_byHour_of_end_DST_Out_of_range     | SNVT 134 / Die Eingangsvariable "stValue.byHour_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x0105     | 261        | eLON_byMinute_of_end_DST_Out_of_range   | SNVT 134 / Die Eingangsvariable "stValue.byMinute_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |

| Wert (hex) | Wert (dez) | Wert (enum)   | Beschreibung  |
|------------|------------|---|---|
| 0x0106     | 262        | eLON_bySecond_of_end_DST_Out_of_range                             | SNVT 134 / Die Eingangsvariable "stValue.bySecond_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                             |
| 0x0107     | 263        | eLON_stStart_DST_uiG_day_of_start_DST_Out_of_range                | SNVT 134 / Die Eingangsvariable "stValue.stStart_DST.uiG_day_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x0108     | 264        | eLON_stStart_DST_uiJ_day_of_start_DST_Out_of_range                | SNVT 134 / Die Eingangsvariable "stValue.stStart_DST.uiJ_day_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x0109     | 265        | eLON_stStart_DST_stM_start_DST_byMonth_of_start_DST_Out_of_range  | SNVT 134 / Die Eingangsvariable "stValue.stStart_DST.stM_start_DST.byMonth_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x010A     | 266        | eLON_stStart_DST_stM_start_DST_byWeek_of_start_DST_Out_of_range   | SNVT 134 / Die Eingangsvariable "stValue.stStart_DST.stM_start_DST.byWeek_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x010B     | 267        | eLON_stStart_DST_stM_start_DST_eDateday_of_start_DST_Out_of_range | SNVT 134 / Die Eingangsvariable "stValue.stStart_DST.stM_start_DST.eDateday_of_start_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x010C     | 268        | eLON_stEnd_DST_uiG_day_of_end_DST_Out_of_range                    | SNVT 134 / Die Eingangsvariable "stValue.stEnd_DST.uiG_day_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x010D     | 269        | eLON_stEnd_DST_uiJ_day_of_end_DST_Out_of_range                    | SNVT 134 / Die Eingangsvariable "stValue.stEnd_DST.uiJ_day_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x010E     | 270        | eLON_stEnd_DST_stM_end_DST_byMonth_of_end_DST_Out_of_range        | SNVT 134 / Die Eingangsvariable "stValue.stEnd_DST.stM_end_DST.byMonth_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x010F     | 271        | eLON_stEnd_DST_stM_end_DST_byWeek_of_end_DST_Out_of_range         | SNVT 134 / Die Eingangsvariable "stValue.stEnd_DST.stM_end_DST.byWeek_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0110     | 272        | eLON_stEnd_DST_stM_end_DST_eDateday_of_end_DST_Out_of_range       | SNVT 134 / Die Eingangsvariable "stValue.stEnd_DST.stM_end_DST.eDateday_of_end_DST" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x0118     | 280        | eLON_byLatitude_deg_Out_of_range                                  | SNVT 135 / Die Eingangsvariable "stValue.byLatitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                      |

| Wert (hex) | Wert (dez) | Wert (enum)                        | Beschreibung  |
|------------|------------|------------------------------------|---|
| 0x0119     | 281        | eLON_rLatitude_min_Out_of_range    | SNVT 135 / Die Eingangsvariable "stValue.rLatitude_min" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x011A     | 282        | eLON_bylongitude_deg_Out_of_range  | SNVT 135 / Die Eingangsvariable "stValue.bylongitude_deg" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x011B     | 283        | eLON_rLongitude_min_Out_of_range   | SNVT 135 / Die Eingangsvariable "stValue.rLongitude_min" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0122     | 290        | eLON_byNr_decimals_Out_of_range    | SNVT 136 / Die Eingangsvariable "stValue.byNr_decimals" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0123     | 291        | eLON_eUnit_Out_of_range            | SNVT 136 / Die Eingangsvariable "stValue.eUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x0127     | 295        | eLON_137eUnit_Out_of_range         | SNVT 137 / Die Eingangsvariable "stValue.eUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x0128     | 296        | eLON_137byNr_decimals_Out_of_range | SNVT 137 / Die Eingangsvariable "stValue.byNr_decimals" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0129     | 297        | eLON_137byStatus_Out_of_range      | SNVT 137 / Die Eingangsvariable "stValue.byStatus" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x012A     | 298        | eLON_137uiYear_Out_of_range        | SNVT 137 / Die Eingangsvariable "stValue.uiYear" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x012B     | 299        | eLON_137uiMonth_Out_of_range       | SNVT 137 / Die Eingangsvariable "stValue.uiMonth" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x012C     | 300        | eLON_137uiDay_Out_of_range         | SNVT 137 / Die Eingangsvariable "stValue.uiDay" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x012D     | 301        | eLON_137uiHour_Out_of_range        | SNVT 137 / Die Eingangsvariable "stValue.uiHour" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x012E     | 302        | eLON_137uiMinute_Out_of_range      | SNVT 137 / Die Eingangsvariable "stValue.uiMinute" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x012F     | 303        | eLON_137uiSecond_Out_of_range      | SNVT 137 / Die Eingangsvariable "stValue.uiSecond" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |

| Wert (hex) | Wert (dez) | Wert (enum)                                | Beschreibung   |
|------------|------------|--|--|
| 0x0136     | 310        | eLON_bySender_prio_Out_of_range            | SNVT 148 / Die Eingangsvariable "stValue.bySender_prio" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x013B     | 315        | eLON_eStatus_Out_of_range                  | SNVT 149 / Die Eingangsvariable "stValue.eStatus" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                  |
| 0x013C     | 316        | eLON_stSender_uid_Out_of_range             | SNVT 149 / Die Eingangsvariable "stValue.stSender.uid" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x013D     | 317        | eLON_stSender_stRange_uiLower_Out_of_range | SNVT 149 / Die Eingangsvariable "stValue.stSender.stRange.uiLower" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x013E     | 318        | eLON_stSender_stRange_uiUpper_Out_of_range | SNVT 149 / Die Eingangsvariable "stValue.stSender.stRange.uiUpper" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x013F     | 319        | eLON_uiController_id_Out_of_range          | SNVT 149 / Die Eingangsvariable "stValue.uiController" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x0145     | 325        | eLON_ePan_dir_Out_of_range                 | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0146     | 326        | eLON_rPan_speed_Out_of_range               | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0147     | 327        | eLON_eTilt_dir_Out_of_range                | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0148     | 328        | eLON_rTilt_speed_Out_of_range              | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0149     | 329        | eLON_eZoom_Out_of_range                    | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x014A     | 330        | eLON_rZoom_speed_Out_of_range              | SNVT 150 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x014F     | 335        | eLON_eAction_Out_of_range                  | SNVT 151 / Die Eingangsvariable "stValue.eAction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                  |
| 0x0154     | 340        | eLON_byController_prio_Out_of_range        | SNVT 152 / Die Eingangsvariable "stValue.byController" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x0155     | 341        | eLON_152eFunction_Out_of_range             | SNVT 152 / Die Eingangsvariable "stValue.eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |

| Wert (hex) | Wert (dez) | Wert (enum)   | Beschreibung  |
|------------|------------|---|---|
| 0x0156     | 342        | eLON_152eAction_Out_of_range                            | SNVT 152 / Die Eingangsvariable "stValue.eAction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                               |
| 0x0157     | 343        | eLON_stValue_stAbspos_rZoom_Out_of_range                | SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rZoom" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x0158     | 344        | eLON_stValue_stAbspos_rTilt_Out_of_range                | SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rTilt" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x0159     | 345        | eLON_stValue_stAbspos_rPan_Out_of_range                 | SNVT 152 / Die Eingangsvariable "stValue.stValue.stAbspos.rPan" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x015E     | 350        | eLON_eMain_pump_Out_of_range                            | SNVT 156 / Die Eingangsvariable "stValue.eMain_pump" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                            |
| 0x015F     | 351        | eLON_eBooster_pump_Out_of_range                         | SNVT 156 / Die Eingangsvariable "stValue.eBooster_pump" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0160     | 352        | eLON_ePriority_level_Out_of_range                       | SNVT 156 / Die Eingangsvariable "stValue.ePriority_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                       |
| 0x0161     | 353        | eLON_eProcess_ready_Out_of_range                        | SNVT 156 / Die Eingangsvariable "stValue.eProcess_ready" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                        |
| 0x0162     | 354        | eLON_eEmergency_stop_activated_Out_of_range             | SNVT 156 / Die Eingangsvariable "stValue.eEmergency_stop_activated" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x0163     | 355        | eLON_eMain_pump_drive_enabled_Out_of_range              | SNVT 156 / Die Eingangsvariable "stValue.eMain_pump_drive_enabled" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x0164     | 356        | eLON_eBooster_pump_drive_enabled_Out_of_range           | SNVT 156 / Die Eingangsvariable "stValue.eBooster_pump_drive_enabled" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x0165     | 357        | eLON_eMaintenance_required_Out_of_range                 | SNVT 156 / Die Eingangsvariable "stValue.eMaintenance_required" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x016D     | 365        | eLON_eControl_status_Out_of_range                       | SNVT 157 / Die Eingangsvariable "stValue.eControl_status" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                       |
| 0x016E     | 366        | eLON_stControl_device_addr_byDomain_length_Out_of_range | SNVT 157 / Die Eingangsvariable "stValue.stControl_device_addr.byDomain_length" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |

| Wert (hex) | Wert (dez) | Wert (enum)                                      | Beschreibung   |
|------------|------------|--|--|
| 0x016F     | 367        | eLON_stControl_device_addr_bySubnet_Out_of_range | SNVT 157 / Die Eingangsvariable "stValue.stControl_device_addr.bySubnet" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0170     | 368        | eLON_stControl_device_addr_byNode_Out_of_range   | SNVT 157 / Die Eingangsvariable "stValue.stControl_device_addr.byNode" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0177     | 375        | eLON_rExhaust_temperature_Out_of_range           | SNVT 158 / Die Eingangsvariable "stValue.rExhaust_temperature" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x0178     | 376        | eLON_rExhaust_pressure_Out_of_range              | SNVT 158 / Die Eingangsvariable "stValue.rExhaust_pressure" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x0179     | 377        | eLON_rShaft_seal_purge_pressure_Out_of_range     | SNVT 158 / Die Eingangsvariable "stValue.rShaft_seal_purge_pressure" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x017A     | 378        | eLON_rSupply_voltage_Out_of_range                | SNVT 158 / Die Eingangsvariable "stValue.rSupply_voltage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x017B     | 379        | eLON_eCoolant_flow_low_Out_of_range              | SNVT 158 / Die Eingangsvariable "stValue.eCoolant_flow_low" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x017C     | 380        | eLON_eDilution_active_Out_of_range               | SNVT 158 / Die Eingangsvariable "stValue.eDilution_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x017D     | 381        | eLON_eBallast_dilution_active_Out_of_range       | SNVT 158 / Die Eingangsvariable "stValue.eBallast_dilution_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x017E     | 382        | eLON_eInlet_purge_dilution_active_Out_of_range   | SNVT 158 / Die Eingangsvariable "stValue.eInlet_purge_dilution_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x017F     | 383        | eLON_eExhaust_dilution_active_Out_of_range       | SNVT 158 / Die Eingangsvariable "stValue.eExhaust_dilution_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x0180     | 384        | eLON_eDilution_flow_Out_of_range                 | SNVT 158 / Die Eingangsvariable "stValue.eDilution_flow" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x0181     | 385        | eLON_ePower_supply_on_Out_of_range               | SNVT 158 / Die Eingangsvariable "stValue.ePower_supply_on" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x0186     | 390        | eLON_rRotational_speed_Out_of_range              | SNVT 159 / Die Eingangsvariable "stValue.rRotational_speed" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |



| Wert (hex) | Wert (dez) | Wert (enum)                                     | Beschreibung  |
|------------|------------|---|---|
| 0x0187     | 391        | eLON_rBody_temperature_Out_of_range             | SNVT 159 / Die Eingangsvariable "stValue.rBody" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x0188     | 392        | eLON_rMotor_external_temperature_Out_of_range   | SNVT 159 / Die Eingangsvariable "stValue.rMotor_external_temperature" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0189     | 393        | eLON_rMotor_internal_temperature_Out_of_range   | SNVT 159 / Die Eingangsvariable "stValue.eMotor_overloaded" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x018A     | 394        | eLON_eMotor_overloaded_Out_of_range             | SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                              |
| 0x018B     | 395        | eLON_eOil_level_low_Out_of_range                | SNVT 159 / Die Eingangsvariable "stValue.ePhase_imbalance_detected" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x018C     | 396        | eLON_ePhase_imbalance_detected_Out_of_range     | SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                              |
| 0x018D     | 397        | eLON_rCurrent_usage_Out_of_range                | SNVT 159 / Die Eingangsvariable "stValue.rCurrent_usage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x018E     | 398        | eLON_rPower_usage_Out_of_range                  | SNVT 159 / Die Eingangsvariable "stValue.Power_usage" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x018F     | 399        | eLON_eTemperature_control_Out_of_range          | SNVT 159 / Die Eingangsvariable "stValue.eElectromagnetic_brake_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0190     | 400        | eLON_eElectromagnetic_brake_active_Out_of_range | SNVT 159 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                              |
| 0x0191     | 401        | eLON_eFriction_brake_active_Out_of_range        | SNVT 159 / Die Eingangsvariable "stValue.eFriction_brake_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.        |
| 0x0192     | 402        | eLON_eGas_brake_active_Out_of_range             | SNVT 159 / Die Eingangsvariable "stValue.eGas_brake_active" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.             |
| 0x019A     | 410        | eLON_164iMilliseconds_Out_of_range              | SNVT 164 / Die Eingangsvariable "stValue.iMilliseconds" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x019B     | 411        | eLON_164ePriority_level_Out_of_range            | SNVT 164 / Die Eingangsvariable "stValue.ePriority_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x019C     | 412        | eLON_164eAlarm_type_Out_of_range                | SNVT 164 / Die Eingangsvariable "stValue.eAlarm" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                        |

| Wert (hex) | Wert (dez) | Wert (enum)   | Beschreibung  |
|------------|------------|---|---|
| 0x01A4     | 420        | eLON_byType_scope_Out_of_range                      | SNVT 166 / Die Eingangsvariable "stValue.byType_scope" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x01A5     | 421        | eLON_uiType_index_Out_of_range                      | SNVT 166 / Die Eingangsvariable "stValue.uiType_index" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x01A6     | 422        | eLON_eType_category_Out_of_range                    | SNVT 166 / Die Eingangsvariable "stValue.eType_category" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x01A7     | 423        | eLON_byType_length_Out_of_range                     | SNVT 166 / Die Eingangsvariable "stValue.byType" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x01AE     | 430        | eLON_eCmd_fb_Out_of_range                           | SNVT 170 / Die Eingangsvariable "stValue.eCmd_fb" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x01B3     | 435        | eLON_byManufacturer_Out_of_range                    | SNVT 172 / Die Eingangsvariable "stValue.byManufacturer" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x01B8     | 440        | eLON_eDevice_select_Out_of_range                    | SNVT 175 / Die Eingangsvariable "stValue.eDevice_select" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x01BD     | 445        | eLON_stPos_eFunction_Out_of_range                   | SNVT 180 / Die Eingangsvariable "stValue.stPos_eFunction" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x01BE     | 446        | eLON_stPos_rSetting_Out_of_range                    | SNVT 180 / Die Eingangsvariable "stValue.stPos.rSetting" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x01BF     | 447        | eLON_stPos_rRotation_Out_of_range                   | SNVT 180 / Die Eingangsvariable "stValue.stPos.rRotation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x01C0     | 448        | eLON_eCmd_source_Out_of_range                       | SNVT 180 / Die Eingangsvariable "stValue.eCmd_source" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x01C1     | 449        | eLON_eError_code_Out_of_range                       | SNVT 180 / Die Eingangsvariable "stValue._eError_code" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x01C7     | 455        | eLON_181stAddr_talk_eAudio_sensor_type_Out_of_range | SNVT 181 / Die Eingangsvariable "stValue.stAddr_talk.eAudio_sensor" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x01C8     | 456        | eLON_181stAddr_talk_byCar_id_Out_of_range           | SNVT 181 / Die Eingangsvariable "stValue.stAddr_talk.byCar_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.      |



| Wert (hex) | Wert (dez) | Wert (enum)   | Beschreibung   |
|------------|------------|---|--|
| 0x01C9     | 457        | eLON_181stAddr_talk_byLocation_Out_of_range               | SNVT 181 / Die Eingangsvariable "stValue.stAddr_talk.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x01CA     | 458        | eLON_181stAddr_talk_byUnit_id_Out_of_range                | SNVT 181 / Die Eingangsvariable "stValue.stAddr_talk.byUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x01CB     | 459        | eLON_181stAddr_init_eAudio_sensor_type_Out_of_range       | SNVT 181 / Die Eingangsvariable "stValue.stAddr_init.eAudio_sensor_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.       |
| 0x01C2     | 450        | eLON_181stAddr_init_byCar_id_Out_of_range                 | SNVT 181 / Die Eingangsvariable "stValue.stAddr_init.byCar" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x01CD     | 461        | eLON_181stAddr_init_byLocation_Out_of_range               | SNVT 181 / Die Eingangsvariable "stValue.stAddr_init.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.               |
| 0x01CE     | 462        | eLON_181stAddr_init_byUnit_id_Out_of_range                | SNVT 181 / Die Eingangsvariable "stValue.stAddr_init.byUnit" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x01CF     | 463        | eLON_181eAudio_type_Out_of_range                          | SNVT 181 / Die Eingangsvariable "stValue.eAudio_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                          |
| 0x01D0     | 464        | eLON_181byAudio_line_Out_of_range                         | SNVT 181 / Die Eingangsvariable "stValue.byAudio_line" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x01D1     | 465        | eLON_181stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range | SNVT 181 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x01D2     | 466        | eLON_181stAddr_dest_stP2p_byCar_id_Out_of_range           | SNVT 181 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x01D3     | 467        | eLON_181stAddr_dest_stP2p_byLocation_Out_of_range         | SNVT 181 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                     |
| 0x01D4     | 468        | eLON_181stAddr_dest_stP2p_byUnit_id_Out_of_range          | SNVT 181 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.byUnit_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x01DB     | 475        | eLON_stAddr_dest_stP2p_eAudio_sensor_type_Out_of_range    | SNVT 182 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.eAudio_sensor_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x01DC     | 476        | eLON_stAddr_dest_stP2p_byCar_id_Out_of_range              | SNVT 182 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.byCar" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |

| Wert (hex) | Wert (dez) | Wert (enum)                                      | Beschreibung   |
|------------|------------|--|--|
| 0x01DD     | 477        | eLON_stAddr_dest_stP2p_byLocation_Out_of_range   | SNVT 182 / Die Eingangsvariable "stValue.stAddr_dest.stP2p.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x01DE     | 478        | eLON_stAddr_dest_stP2p_byUnit_id_Out_of_range    | SNVT 182 / Die Eingangsvariable "stValue.stAddr_init.eAudio_sensor_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x01DF     | 479        | eLON_stAddr_init_eAudio_sensor_type_Out_of_range | SNVT 182 / Die Eingangsvariable "stValue.stAddr_init.byCar_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.           |
| 0x01E0     | 480        | eLON_stAddr_init_byCar_id_Out_of_range           | SNVT 182 / Die Eingangsvariable "stValue." liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                               |
| 0x01E1     | 481        | eLON_stAddr_init_byLocation_Out_of_range         | SNVT 182 / Die Eingangsvariable "stValue.stAddr_init.byLocation" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x01E2     | 482        | eLON_stAddr_init_byUnit_id_Out_of_range          | SNVT 182 / Die Eingangsvariable "stValue.stAddr_init.byUnit_id" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x01E3     | 483        | eLON_eAudio_type_Out_of_range                    | SNVT 182 / Die Eingangsvariable "stValue.eAudio_type" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x01EA     | 490        | eLON_eCycle_Out_of_range                         | SNVT 184 / Die Eingangsvariable "stValue.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                         |
| 0x01EB     | 491        | eLON_eSubcycle_Out_of_range                      | SNVT 184 / Die Eingangsvariable "stValue.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                      |
| 0x01EC     | 492        | eLON_stFunction_eProgram_Out_of_range            | SNVT 184 / Die Eingangsvariable "stValue.stFunction.eProgram" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.            |
| 0x01ED     | 493        | eLON_stFunction_stWash_eLoad_level_Out_of_range  | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stWash_eLoad_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x01EE     | 494        | eLON_stFunction_stWash_ePrewash_Out_of_range     | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stWash.ePrewash" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x01EF     | 495        | eLON_stFunction_stRinse_eOption_Out_of_range     | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stRinse.eOption" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x01F0     | 496        | eLON_stFunction_stRinse_byRepeat_Out_of_range    | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stRinse.byRepeat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.    |

| Wert (hex) | Wert (dez) | Wert (enum)   | Beschreibung  |
|------------|------------|---|---|
| 0x01F1     | 497        | eLON_stFunction_stSpin_eHold_Out_of_range                             | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stSpin.eHold" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                           |
| 0x01F2     | 498        | eLON_stFunction_stDry_byTemp_Out_of_range                             | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stDry.byTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                           |
| 0x01F3     | 499        | eLON_stFunction_stDry_stDuration_eDryness_Out_of_range                | SNVT 184 / Die Eingangsvariable "stValue.stFunction.stDry.stDuration.eDryness" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x01F9     | 505        | eLON_186eCycle_Out_of_range   | SNVT 186 / Die Eingangsvariable "stValue.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x01FA     | 506        | eLON_186eSubcycle_Out_of_range  | SNVT 186 / Die Eingangsvariable "stValue.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x01FB     | 507        | eLON_stWasher_command_data_eCycle_Out_of_range                        | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.eCycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                      |
| 0x01FC     | 508        | eLON_stWasher_command_data_eSubcycle_Out_of_range                     | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.eSubcycle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x01FD     | 509        | eLON_stWasher_command_data_stFunction_eProgram_Out_of_range           | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.eProgram" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x01FE     | 510        | eLON_stWasher_command_data_stFunction_stWash_eLoad_level_Out_of_range | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stWash.eLoad" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.     |
| 0x01FF     | 511        | eLON_stWasher_command_data_stFunction_stWash_ePrewash_Out_of_range    | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stWash.ePrewash" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0200     | 512        | eLON_stWasher_command_data_stFunction_stRinse_eOption_Out_of_range    | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stRinse.eOption" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0201     | 513        | eLON_stWasher_command_data_stFunction_stRinse_byRepeat_Out_of_range   | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stRinse.byRepeat" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |

| Wert (hex) | Wert (dez) | Wert (enum)  | Beschreibung   |
|------------|------------|--|--|
| 0x0202     | 514        | eLON_stWasher_command_data_stFunction_stSpin_eHold_Out_of_range              | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stSpin.eHold" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x0203     | 515        | eLON_stWasher_command_data_stFunction_stDry_byTemp_Out_of_range              | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stDry.byTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.              |
| 0x0204     | 516        | eLON_stWasher_command_data_stFunction_stDry_stDuration_eDryness_Out_of_range | SNVT 186 / Die Eingangsvariable "stValue.stWasher_command_data.stFunction.stDry.stDuration.eDryness" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0206     | 518        | eLON_eState_Out_of_range   | SNVT 189 / Die Eingangsvariable "stValue.eState" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0207     | 519        | eLON_stSetting_rValue_Out_of_range   | SNVT 189 / Die Eingangsvariable "stValue.stSettings.rValue" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x0208     | 520        | eLON_stSetting_rChange_Out_of_range  | SNVT 189 / Die Eingangsvariable "stValue.stSettings.rChange" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.   |
| 0x0209     | 521        | eLON_stSetting_rMultiplier_Out_of_range                                      | SNVT 189 / Die Eingangsvariable "stValue.stSettings.rMultiplier" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                     |
| 0x020A     | 522        | eLON_stSetting_iAngle_Out_of_range   | SNVT 189 / Die Eingangsvariable "stValue.stSettings.iAngle" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.  |
| 0x020B     | 523        | eLON_stSetting_byGroup_number_Out_of_range                                   | SNVT 189 / Die Eingangsvariable "stValue.stSettings.byGroup_number" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                  |
| 0x020C     | 524        | eLON_stSetting_siFan_level_Out_of_range                                      | SNVT 189 / Die Eingangsvariable "stValue.stSettings.siFan_level" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                     |
| 0x020D     | 525        | eLON_stColor_value_stCIE1931_lumen_rX_Out_of_range                           | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_lumen.rX" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                           |
| 0x020E     | 526        | eLON_stColor_value_stCIE1931_lumen_rY_Out_of_range                           | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_lumen.rY" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                           |
| 0x020F     | 527        | eLON_stColor_value_stCIE1931_lumen_uY_Out_of_range                           | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_lumen.uY" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                           |

| Wert (hex) | Wert (dez) | Wert (enum)  | Beschreibung   |
|------------|------------|--|--|
| 0x0210     | 528        | eLON_stColor_value_stCIE1931_percent_rX_Out_of_range         | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_percent.rX" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0211     | 529        | eLON_stColor_value_stCIE1931_percent_rY_Out_of_range         | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_percent.rY" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0212     | 530        | eLON_stColor_value_stCIE1931_percent_rPercent_Y_Out_of_range | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.stCIE1931_percent.rPercent_Y" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0213     | 531        | eLON_stColor_value_uiColor_temperaturre_Out_of_range         | SNVT 190 / Die Eingangsvariable "stValue.stColor_value.uiColor_temperaturre" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0217     | 535        | eLON_191eStatus_Out_of_range                                 | SNVT 191 / Die Eingangsvariable "stValue.Status" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                     |
| 0x0218     | 536        | eLON_uiLog_number_Out_of_range                               | SNVT 191 / Die Eingangsvariable "stValue.uiLog_number" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                               |
| 0x0219     | 537        | eLON_rLevel_Out_of_range                                     | SNVT 191 / Die Eingangsvariable "stValue.rLevel" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                     |
| 0x021A     | 538        | eLON_stCurrent_notify_time_rHundredths_Out_of_range          | SNVT 191 / Die Eingangsvariable "stValue.stCurrent_notify_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.          |
| 0x021B     | 539        | eLON_stPrevious_notify_time_rHundredths_Out_of_range         | SNVT 191 / Die Eingangsvariable "stValue.stPrevious_notify_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.         |
| 0x0221     | 545        | eLON_rHundredths_Out_of_range                                | SNVT 192 / Die Eingangsvariable "stValue.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                |
| 0x0226     | 550        | eLON_stStart_time_rHundredths_Out_of_range                   | SNVT 193 / Die Eingangsvariable "stValue.stStart_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                   |
| 0x0227     | 551        | eLON_stEnd_time_rHundredths_Out_of_range                     | SNVT 193 / Die Eingangsvariable "stValue.stEnd_time.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                     |
| 0x0235     | 565        | eLON_rComplete_Out_of_range                                  | SNVT 194 / Die Eingangsvariable "stValue.rComplete" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                                  |
| 0x023A     | 570        | eLON_stTime_actual_rHundredths_Out_of_range                  | SNVT 199 / Die Eingangsvariable "stValue.stTime_actual.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                  |

| Wert (hex) | Wert (dez) | Wert (enum)                                   | Beschreibung  |
|------------|------------|---|---|
| 0x023B     | 571        | eLON_stTime_previous_rHundredths_Out_of_range | SNVT 199 / Die Eingangsvariable "stValue.stTime_previous.rHundredths" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet. |
| 0x0249     | 585        | eLON_lrEnergy_Out_of_range                    | SNVT 200 / Die Eingangsvariable "stValue.lrEnergy" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                    |
| 0x024A     | 586        | eLON_rPowerFactor_Out_of_range                | SNVT 200 / Die Eingangsvariable "stValue.rPowerFactor" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x024B     | 587        | eLON_rPower_Out_of_range                      | SNVT 200 / Die Eingangsvariable "stValue.rPower" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                      |
| 0x024C     | 588        | eLON_rBallastTemp_Out_of_range                | SNVT 200 / Die Eingangsvariable "stValue.rBallastTemp" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                |
| 0x0253     | 595        | eLON_lrLongitude_Out_of_range                 | SNVT 201 / Die Eingangsvariable "stValue.lrLongitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                 |
| 0x0254     | 596        | eLON_lrLatitude_Out_of_range                  | SNVT 201 / Die Eingangsvariable "stValue.lrLatitude" liegt außerhalb des zulässigen Bereichs. Der Wert wurde nicht gesendet.                  |



## 8 Anhang

### 8.1 Automatisches Senden

Mit der Eingangsvariable **bAuto** wird das automatische Senden eingeschaltet. Die Variable muss über den gesamte Zeitraum, in dem der Baustein selbstständig senden soll, auf TRUE stehen.

Mit den folgenden drei Parametern (VAR\_INPUT) kann Einfluss auf das automatische Senden genommen werden.

**MaxSendTime:** TIME;

Mit diesem Wert werden Übertragungen von Werten in regelmäßigen Abständen ermöglicht. Nach Ablauf der Zeit wird der Wert gesendet, unabhängig von einer möglichen Wertänderung.

**ValueLimit:** Dieser Wert kann in den folgenden Formaten vorliegen: (r/lr/ui/i) **ValueLimit:** REAL / LREAL / UINT / INT.

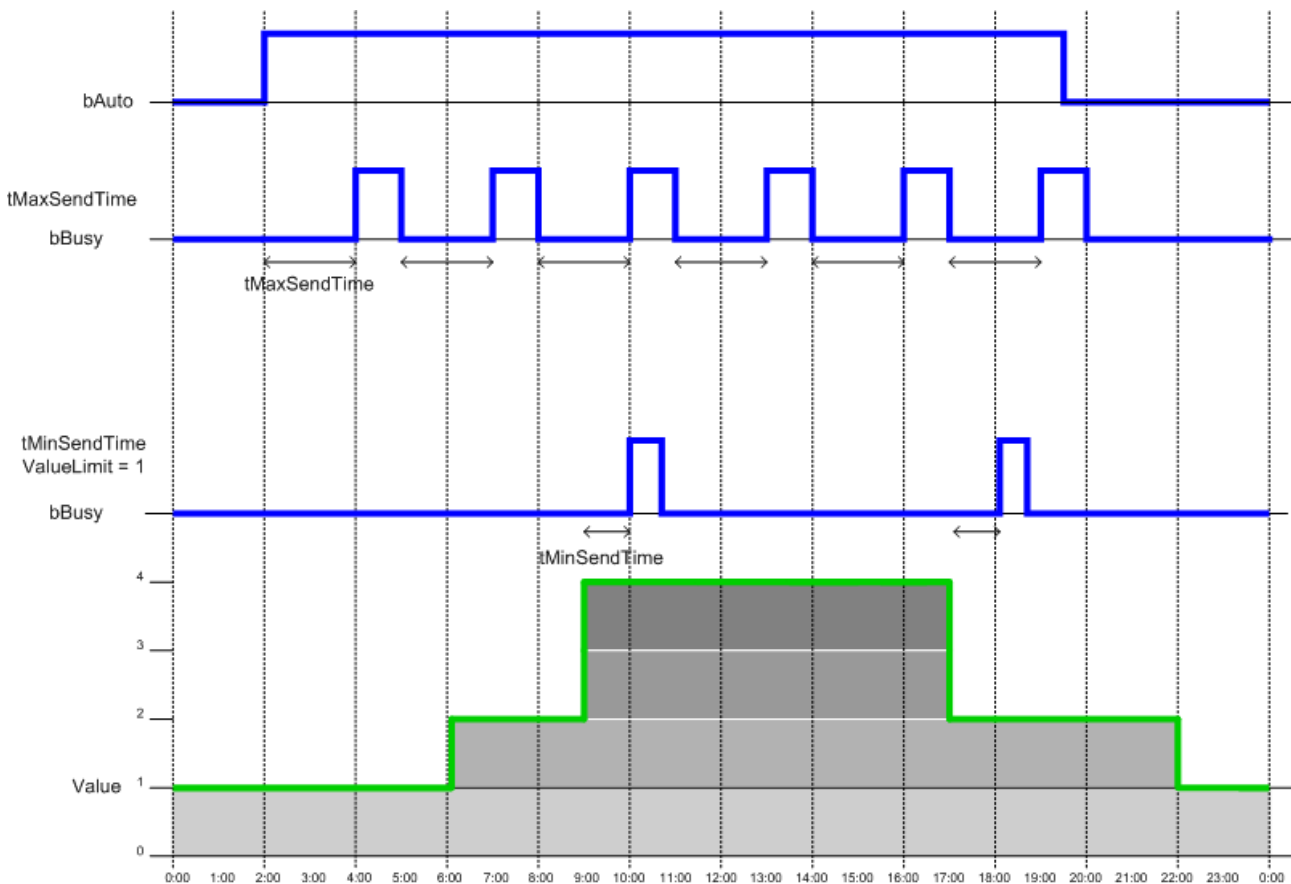
Der Wert wird nur gesendet, wenn der Betrag der Änderung seit der letzten Übertragung größer als dieser Parameter ist. Wenn dieser Wert 0 ist, wird nach jeder Wertänderung (auch minimal) gesendet.

Es ist zu beachten: Bei Enums und Strukturen ist die Variable **ValueLimit** nicht vorhanden, dort wird nach jeder Wertänderung gesendet.

**tMinSendTime:** TIME;

Der Baustein sendet frühestens nach **tMinSendTime**. Damit kann die Anzahl der Telegramme begrenzt werden, wenn sich die Werte sehr schnell ändern. (Reduzierung der Netzwerkbelastung). Ist der Wert t#0s, wird auf jede Wertänderung gesendet (siehe **ValueLimit**).

Abb.1



## 8.2 dwErrorKL

| Wert (hex)  | Firmware | Beschreibung   |
|-------------|----------|--|
| 0x0000 0000 |          | Kein Fehler.   |
| 0x0000 0001 |          | Schreibzugriff auf einen Leseparameter.  |
| 0x0000 0004 |          | Nicht definierter Parameter.   |
| 0x0000 0005 |          | Nicht zugelassener Wert eines Parameters.  |
| 0x0000 0007 |          | Nicht definierter Slot.  |
| 0x0000 0008 |          | Fehler beim auslesen des NV Parameters.  |
| 0x0000 0009 |          | Checksummenfehler beim Download der LON-Konfiguration.   |
| 0x0000 0010 | ab FW5   | OUTGOING_MSG_FAILED  |
| 0x0000 0011 | ab FW5   | OUTGOING_MSG_LATE_ACK  |
| 0x0000 0012 | ab FW5   | OUTGOING_MSG_MALFORMED   |
| 0x0000 0013 | ab FW5   | NEURON_QUERY_FAILED  |
| 0x0000 0014 | ab FW5   | NEURON_UPDATE_FAILED   |
| 0x0000 000A |          | Upload einer LON Konfiguration fehlerhaft.   |
| 0x0000 0Cxx |          | Zu schreibende SNVT-Variable beim Empfänger nicht angekommen. xx entspricht der SNVT-Indexnummer.  |
| 0x0000 0Exx |          | NV Index ist kein Ausgang. Die SPS versucht auf einen NV Index zu schreiben, der mit der KS2000 nicht als Ausgang (NVO) definiert wurde. xx entspricht der SNVT-Indexnummer. |
| 0x0000 0Fxx |          | Die Bestätigung eines zu schreibenden LON-Telegramm ist zu spät eingetroffen (>1500ms). xx entspricht der SNVT-Indexnummer.  |
| 0x0000 400x | ab FW5   | NEURON_MGMT_ERROR. x entspricht dem API Code.  |

## 8.3 Beispiele

| Beispiel  | Beschreibung   |
|---|--|
| <a href="https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997704331.zip">https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997704331.zip</a> | Alle SNVTs werden je 1 mal aufgerufen (7 x KL6401)                   |
| <a href="https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997705739.zip">https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997705739.zip</a> | Alle SNVTs werden je 1 mal aufgerufen (7 x KL6401)                   |
| <a href="https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997707147.zip">https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997707147.zip</a> | Konfigurationsdateien für die KS2000 zur Parametrierung der Klemmen. |

## 8.4 SNVT-Variablen (OFF)

XIF-File: <https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997698699.zip> 

KS2000 BLC-File: <https://infosys.beckhoff.com/content/1031/tcplcliblon/Resources/11997700107.zip> 

### Implementierte SNVT-Variablen (KL6401\_OFF)

| Beschreibung | INPUT/OUTPUT | SNVT ID | Länge | NV ID |
|--------------|--------------|---------|-------|-------|
| nviSwitch00  | INPUT        | 95      | 2     | 0     |
| nviSwitch01  | INPUT        | 95      | 2     | 1     |
| nviSwitch02  | INPUT        | 95      | 2     | 2     |
| nviSwitch03  | INPUT        | 95      | 2     | 3     |
| nviSwitch04  | INPUT        | 95      | 2     | 4     |
| nviSwitch05  | INPUT        | 95      | 2     | 5     |
| nviSwitch06  | INPUT        | 95      | 2     | 6     |
| nviSwitch07  | INPUT        | 95      | 2     | 7     |



| Beschreibung   | INPUT/OUTPUT | SNVT ID | Länge | NV ID |
|----------------|--------------|---------|-------|-------|
| nviSwitch08    | INPUT        | 95      | 2     | 8     |
| nviSwitch09    | INPUT        | 95      | 2     | 9     |
| nviSwitch10    | INPUT        | 95      | 2     | 10    |
| nviSwitch11    | INPUT        | 95      | 2     | 11    |
| nviSetting0    | INPUT        | 117     | 4     | 12    |
| nviSetting1    | INPUT        | 117     | 4     | 13    |
| nviSetting2    | INPUT        | 117     | 4     | 14    |
| nviSetting3    | INPUT        | 117     | 4     | 15    |
| nviTemp0       | INPUT        | 105     | 2     | 16    |
| nviTemp1       | INPUT        | 105     | 2     | 17    |
| nviTemp2       | INPUT        | 105     | 2     | 18    |
| nviTemp3       | INPUT        | 105     | 2     | 19    |
| nviTemp4       | INPUT        | 105     | 2     | 20    |
| nviHvacStatus0 | INPUT        | 112     | 12    | 21    |
| nviHvacStatus1 | INPUT        | 112     | 12    | 22    |
| nviHvacStatus2 | INPUT        | 112     | 12    | 23    |
| nviHvacStatus3 | INPUT        | 112     | 12    | 24    |
| nviHvacStatus4 | INPUT        | 112     | 12    | 25    |
| nvoSwitch00    | OUTPUT       | 95      | 2     | 26    |
| nvoSwitch01    | OUTPUT       | 95      | 2     | 27    |
| nvoSwitch02    | OUTPUT       | 95      | 2     | 28    |
| nvoSwitch03    | OUTPUT       | 95      | 2     | 29    |
| nvoSwitch04    | OUTPUT       | 95      | 2     | 30    |
| nvoSwitch05    | OUTPUT       | 95      | 2     | 31    |
| nvoSwitch06    | OUTPUT       | 95      | 2     | 32    |
| nvoSwitch07    | OUTPUT       | 95      | 2     | 33    |
| nvoSwitch08    | OUTPUT       | 95      | 2     | 34    |
| nvoSwitch09    | OUTPUT       | 95      | 2     | 35    |
| nvoSwitch10    | OUTPUT       | 95      | 2     | 36    |
| nvoSwitch11    | OUTPUT       | 95      | 2     | 37    |
| nvoSwitch12    | OUTPUT       | 95      | 2     | 38    |
| nvoSwitch13    | OUTPUT       | 95      | 2     | 39    |
| nvoSwitch14    | OUTPUT       | 95      | 2     | 40    |
| nvoSwitch15    | OUTPUT       | 95      | 2     | 41    |
| nvoSwitch16    | OUTPUT       | 95      | 2     | 42    |
| nvoSwitch17    | OUTPUT       | 95      | 2     | 43    |
| nvoSetting0    | OUTPUT       | 117     | 4     | 44    |
| nvoSetting1    | OUTPUT       | 117     | 4     | 45    |
| nvoSetting2    | OUTPUT       | 117     | 4     | 46    |
| nvoSetting3    | OUTPUT       | 117     | 4     | 47    |
| nvoLevP0       | OUTPUT       | 81      | 2     | 48    |
| nvoLevP1       | OUTPUT       | 81      | 2     | 49    |
| nvoLevP2       | OUTPUT       | 81      | 2     | 50    |
| nvoLevP3       | OUTPUT       | 81      | 2     | 51    |
| nvoLevP4       | OUTPUT       | 81      | 2     | 52    |
| nvoHvacMode0   | OUTPUT       | 108     | 1     | 53    |
| nvoHvacMode1   | OUTPUT       | 108     | 1     | 54    |
| nvoHvacMode2   | OUTPUT       | 108     | 1     | 55    |

| Beschreibung | INPUT/OUTPUT | SNVT ID | Länge | NV ID |
|--------------|--------------|---------|-------|-------|
| nvoHvacMode3 | OUTPUT       | 108     | 1     | 56    |
| nvoHvacMode4 | OUTPUT       | 108     | 1     | 57    |

## 8.5 Support und Service

Beckhoff und seine weltweiten Partnerfirmen bieten einen umfassenden Support und Service, der eine schnelle und kompetente Unterstützung bei allen Fragen zu Beckhoff Produkten und Systemlösungen zur Verfügung stellt.

### Downloadfinder

Unser [Downloadfinder](#) beinhaltet alle Dateien, die wir Ihnen zum Herunterladen anbieten. Sie finden dort Applikationsberichte, technische Dokumentationen, technische Zeichnungen, Konfigurationsdateien und vieles mehr.

Die Downloads sind in verschiedenen Formaten erhältlich.

### Beckhoff Niederlassungen und Vertretungen

Wenden Sie sich bitte an Ihre Beckhoff Niederlassung oder Ihre Vertretung für den [lokalen Support und Service](#) zu Beckhoff Produkten!

Die Adressen der weltweiten Beckhoff Niederlassungen und Vertretungen entnehmen Sie bitte unserer Internetseite: [www.beckhoff.com](http://www.beckhoff.com)

Dort finden Sie auch weitere Dokumentationen zu Beckhoff Komponenten.

### Beckhoff Support

Der Support bietet Ihnen einen umfangreichen technischen Support, der Sie nicht nur bei dem Einsatz einzelner Beckhoff Produkte, sondern auch bei weiteren umfassenden Dienstleistungen unterstützt:

- Support
- Planung, Programmierung und Inbetriebnahme komplexer Automatisierungssysteme
- umfangreiches Schulungsprogramm für Beckhoff Systemkomponenten

Hotline: +49 5246 963-157  
E-Mail: [support@beckhoff.com](mailto:support@beckhoff.com)

### Beckhoff Service

Das Beckhoff Service-Center unterstützt Sie rund um den After-Sales-Service:

- Vor-Ort-Service
- Reparaturservice
- Ersatzteilservice
- Hotline-Service

Hotline: +49 5246 963-460  
E-Mail: [service@beckhoff.com](mailto:service@beckhoff.com)

### Beckhoff Unternehmenszentrale

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20  
33415 Verl  
Deutschland

Telefon: +49 5246 963-0

E-Mail: [info@beckhoff.com](mailto:info@beckhoff.com)  
Internet: [www.beckhoff.com](http://www.beckhoff.com)



Mehr Informationen:  
**[www.beckhoff.de/tx1200](http://www.beckhoff.de/tx1200)**

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Deutschland  
Telefon: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

