BECKHOFF New Automation Technology

Manual | EN

TX1200

TwinCAT 2 | PLC Library: TcRAIDController

PLC Libraries

Table of contents

BECKHOFF

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

EtherCAT® is a registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2   Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ **DANGER**
Serious risk of injury!
Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ **WARNING**
Risk of injury!
Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ **CAUTION**
Personal injuries!
Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE
Damage to the environment or devices
Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3   Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2   Introduction

Following function blocks are available for RAID controller services.

FB_RAIDFindCntlr [▶ 9] : This function block returns the RAID controller count and the corresponding RAID controller IDs.

FB_RAIDGetInfo [▶ 10] : This function block returns RAID Info, which includes number of RAID controller sets and maximum number of RAID drives per set.
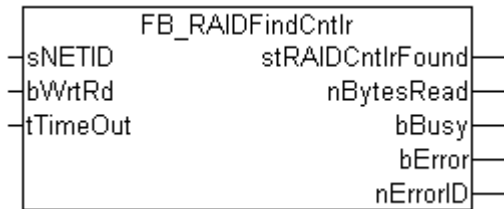
FB_RAIDGetStatus [▶ 11] : This function block returns the RAID set index, RAID type, RAID status, RAID Drive count, RAID Drive status.

i	Calling the function blocks FB_RAIDFindCntlr and FB_RAIDGetInfo only once in PLC program fulfills the need. System performance can be dramatically affected due to repetitive cyclic call of these function block.

3 Function Blocks

3.1 FB_RAIDFindCntlr

```
         FB_RAIDFindCntlr
─┤sNETID         stRAIDCntlrFound├─
─┤bWrtRd            nBytesRead├─
─┤tTimeOut               bBusy├─
                         bError├─
                        nErrorID├─
```

This function block returns the RAID controller count and the corresponding controller IDs.

ⓘ Calling this function block only once in PLC program fulfills the need. System performance can be dramatically affected due to cyclic call of this function block.

VAR_INPUT

```
sNETID          : T_AmsNetId;
bWrtRd          : BOOL;
tTimeOut        : TIME := DEFAULT_ADS_TIMEOUT;
```

sNETID: Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

bWrtRd: The ADS command is triggered by a rising edge at this input.

tTimeOut: States the time before the function is cancelled.

VAR_OUTPUT

```
stRAIDCntlrFound    : ST_RAIDCntlrFound;
nBytesRead          : UDINT;
bBusy               : BOOL;
bError              : BOOL;
nErrorID            : UDINT;
```

stRAIDCntlrFound: includes the count of RAID controller [▶ 13] found and corresponding RAID controller IDs.

nBytesRead: Number of succesfully returned data bytes.

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.

nErrorID: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs.

3.2 FB_RAIDGetInfo

```
        FB_RAIDGetInfo
─┤sNETID              stRAIDInfo├─
─┤bWrtRd             nBytesRead├─
─┤nRAIDCntlrID            bBusy├─
─┤tTimeOut                bError├─
                        nErrorID├─
```

This function block returns RAID info, which includes number of RAID controller sets and maximum number of drives per set.

ℹ Calling this function block only once in PLC program fulfills the need. System performance can be dramatically affected due to cyclic call of this function block.

VAR_INPUT

```
sNETID          : T_AmsNetId;
bWrtRd          : BOOL;
nRAIDCntlrID    : UDINT;
tTimeOut        : TIME := DEFAULT_ADS_TIMEOUT;
```

sNETID: Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

bWrtRd: The ADS command is triggered by a rising edge at this input.

nRAIDCntlrID: The RAID controller ID. (**Hint**: Can be read using FB_RAIDCntlrFind)

tTimeOut: States the time before the function is cancelled.

VAR_OUTPUT

```
stRAIDInfo      : ST_RAIDInfo;
nBytesRead      : UDINT;
bBusy           : BOOL;
bError          : BOOL;
nErrorID        : UDINT;
```

stRAIDInfo: returns RAID info, which includes number of RAID controller [▶ 14] sets and maximum number of drives per set.

nBytesRead: Number of succesfully returned data bytes.

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.

nErrorID: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs.

3.3   FB_RAIDGetStatus

```
          FB_RAIDGetStatus
—|sNETID              stRAIDStatusRes|—
—|bWrtRd                  nBytesRead|—
—|stRAIDConfigReq              bBusy|—
—|tTimeOut                    bError|—
                            nErrorID|—
```

This function block returns the RAID set index, RAID type, RAID status, RAID drive count, RAID drive status.

ⓘ Please not that system performance can be dramatically affected due to cyclic call of this function block. The fastest recommended factor for this function block is once per second.

VAR_INPUT

```
sNETID          : T_AmsNetId;
bWrtRd          : BOOL;
stRAIDConfigReq : ST_RAIDConfigReq;
tTimeOut        : TIME := DEFAULT_ADS_TIMEOUT;
```

sNETID: Is a string containing the AMS network identifier of the target device to which the ADS command is directed.

bWrtRd: The ADS command is triggered by a rising edge at this input.

stRAIDConfigReq: RAID configuration request parameters [▶ 14] are to be stated in this structure. It includes controller ID and RAID set index.

tTimeOut: States the time before the function is cancelled.

VAR_OUTPUT

```
stRAIDStatusRes : ST_RAIDStatusRes;
nBytesRead      : UDINT;
bBusy           : BOOL;
bError          : BOOL;
nErrorID        : UDINT;
```

stRAIDStatusRes: RAID status response [▶ 14] is received in this structure format. It includes RAID set index, RAID type, RAID status, RAID drive count, RAID drive status.

nBytesRead: Number of succesfully returned data bytes.

bBusy: This output remains TRUE until the block has executed a command, but at the longest for the duration supplied to the 'Timeout' input. While Busy = TRUE, no new command will be accepted at the inputs. Please note that it is not the execution of the service but its acceptance whose time is monitored.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in 'nErrorId'. If the block has a timeout error, 'bError' is TRUE and 'nErrorId' is 1861 (hexadecimal 0x745). Is reset to FALSE by the execution of a command at the inputs.

nErrorID: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs.

4 Data Types

4.1 E_RAIDDriveStatus

```
TYPE E_RAIDDriveStatus :
(
eRAID_DRIVE_STATUS_OK         := 0,
eRAID_DRIVE_STATUS_REBUILDING := 1,
eRAID_DRIVE_STATUS_FAILED     := 2,
eRAID_DRIVE_STATUS_DEGRADED   := 3
);
END_TYPE
```

0 = eRAID_DRIVE_STATUS_OK, indicates the physical drive is operational.

1 = eRAID_DRIVE_STATUS_DEGRADED, indicates the physical drive has posted a SMART notification to the controller.

2 = eRAID _DRIVE_STATUS_REBUILDING, indicates the physical drive is the target drive of a RAID set rebuild. Once the rebuild completes successfully, the status will change to **eRAID_DRIVE_STATUS_OK**. If the rebuilding process fails, the status will be updated appropriately.

3 = eRAID_DRIVE_STATUS_FAILED, indicates the physical drive has posted unrecoverable errors to the controller or has triggered a vendor specific action to remove the physical drive from the RAID set. There is no guarantee on the operational behavior of the drive and data loss has occurred or is imminent.

4.2 E_RAIDDriveUsage

```
TYPE E_RAIDDriveUsage :
(
eRAID_DRIVE_CONFIG_NOT_USED   := 0,
eRAID_DRIVE_CONFIG_MEMBER     := 1,
eRAID_DRIVE_CONFIG_SPARE      := 2
);
END_TYPE
```

0 = eRAID_DRIVE_CONFIG_NOT_USED, indicates the physical drive is not part of a RAID set.

1 = eRAID_DRIVE_CONFIG_MEMBER, indicates the physical drive is part of this RAID set.

2 = eRAID_DRIVE_CONFIG_SPARE, indicates the physical drive is part of this RAID set as a hot swap spare.

4.3 E_RAIDStatus

```
TYPE E_RAIDStatus :
(
eRAID_SET_STATUS_OK         := 0,
eRAID_SET_STATUS_DEGRADED   := 1,
eRAID_SET_STATUS_REBUILDING := 2,
eRAID_SET_STATUS_FAILED     := 3
);
END_TYPE
```

0 = eRAID_SET_STATUS_OK, indicates the RAID set is operational.

1 = e RAID_SET_STATUS_DEGRADED, indicates the RAID set is no longer functioning in a fault tolerant mode.

2 = eRAID_SET_STATUS_REBUILDING, indicates the RAID set is rebuilding. This implies a degraded operation. Once the rebuild completes successfully, the status will change to **eRAID_SET_STATUS_OK**. If the rebuilding process fails, the status will be updated appropriately.

3 =eRAID_SET_STATUS_FAILED, indicates the RAID set has failed. There is no guarantee on the operational behavior of the RAID set and data loss has occurred or is imminent.

4.4 E_RAIDType

```
TYPE E_RAIDType :
(
eRAID_TYPE_NONE     := 0,
eRAID_TYPE_0        := 1,
eRAID_TYPE_1        := 2,
eRAID_TYPE_10       := 3,
eRAID_TYPE_5        := 4,
eRAID_TYPE_15       := 5,
eRAID_TYPE_OTHER    := 255
);
END_TYPE
```

0 = eRAID_TYPE_NONE, indicates the RAID set is composed of a single drive. No Set with the given number exists.

1 = eRAID_TYPE_0, indicates the RAID set is a striped set, with no fault tolerance.

2 = eRAID_TYPE_1, indicates the RAID set is a mirrored set.

3 = eRAID_TYPE_10, indicates the RAID set is a striped mirror set.

4 = eRAID_TYPE_5, indicates the RAID set is a parity set.

5 = eRAID_TYPE_15, indicates the RAID set is an advanced parity set.

255 = eRAID_TYPE_OTHER, indicates the RAID set type configuration does not match the standard types.

4.5 ST_RAIDCntlrFound

```
TYPE ST_RAIDCntlrFound :
STRUCT
    nRAIDCntlrCount  : UDINT;
    nRAIDCntlrIds    : ARRAY [1..g_nMAX_NUMBER_OF_RAID_CNTLRS] OF UDINT;
END_STRUCT
END_TYPE
```

nRAIDCntlrCount: Number of RAID controllers

nRAIDCntlrIds: ID of each RAID controller (Default value is 4294967295 and hence invalid). **g_nMAX_NUMBER_OF_RAID_CNTLRS** is the maximum number of RAID controllers and is defined as global constant = 10.

4.6 ST_RAIDDriveStatus

```
TYPE ST_RAIDDriveStatus :
STRUCT
    eRAIDDriveStatus        : E_RAIDDriveStatus;
    eRAIDDriveUsage         : E_RAIDDriveUsage;
    nSATAPort               : UINT;
    sRAIDDriveSerial        : STRING [39];
END_STRUCT
END_TYPE
```

eRAIDDriveStatus: Contains the status [▶ 12] of the physical drive.

eRAIDDriveUsage: Contains whether [▶ 12] the physical drive is part of the RAID set.

nSATAPort: Contains the SAS address of the physical drive. If the drive does not have a SAS address as is the case with a directly attached SATA drive, then this field shall be 0 filled

sRAIDDriveSerial: serial number designation of the RAID drive (40 characters).

4.7   ST_RAIDConfigReq

```
TYPE ST_RAIDConfigReq :
STRUCT
    nRAIDCntlrID    : UDINT;
    nRAIDSetIndex   : UDINT;
END_STRUCT
END_TYPE
```

nRAIDCntlrID: RAID controller ID

nRAIDSetIndex: Contains the number of the RAID set for which information is being requested. Please **note** that in case of Beckhoff Boards CBx051 the sets begin with Index 0.

4.8   ST_RAIDInfo

```
TYPE ST_RAIDInfo :
STRUCT
    nNumRAIDSets    : UDINT;
    nMaxDrivesPerSet    : UDINT;
    bReserved       : ARRAY [1..92] OF BYTE;
END_STRUCT
END_TYPE
```

nNumRAIDSets: Number of logical RAID sets currently defined. If no sets have been defined, then a 0 value is returned.

nMaxDriverPerSet: Maximum number of physical drives within a logical RAID set. This may be an absolute maximum or the actual maximum currently defined for all sets

bReserved: reserved for internal use

4.9   ST_RAIDStatusRes

```
TYPE ST_RAIDStatusRes :
STRUCT
    nRAIDSetIndex       : UDINT;
    eRAIDType           : E_RAIDType;
    eRAIDStatus         : E_RAIDStatus;
    nRAIDDriveCount     : UINT;
    nReserved           : UINT;
    stRAIDDriveStatus   : ARRAY [1..g_nMAX_NUMBER_OF_RAID_DRIVES] OF ST_RAIDDriveStatus;
END_STRUCT
END_TYPE
```

nRAIDSetIndex:RAID set ID, same as input.

eRAIDType: Contains the basic RAID type [▶ 13] of the RAID set. **eRAID_TYPE_NONE**, indicates the RAID set is composed of a single drive. In other words, no set with the given number exists.

eRAIDStatus: Contains the status [▶ 12] of the RAID set.

nRAIDDriveCount: Contains the number of drives in the RAID set

nReserved: reserved

stRAIDDriveStatus: Contains the status of the physical drive and whether the physical drive is part of the RAID set. **g_nMAX_NUMBER_OF_RAID_DRIVES** is the number of RAID drive status [▶ 13] that can be read and is defined as global constant = 10.

More Information:
www.beckhoff.com/tx1200