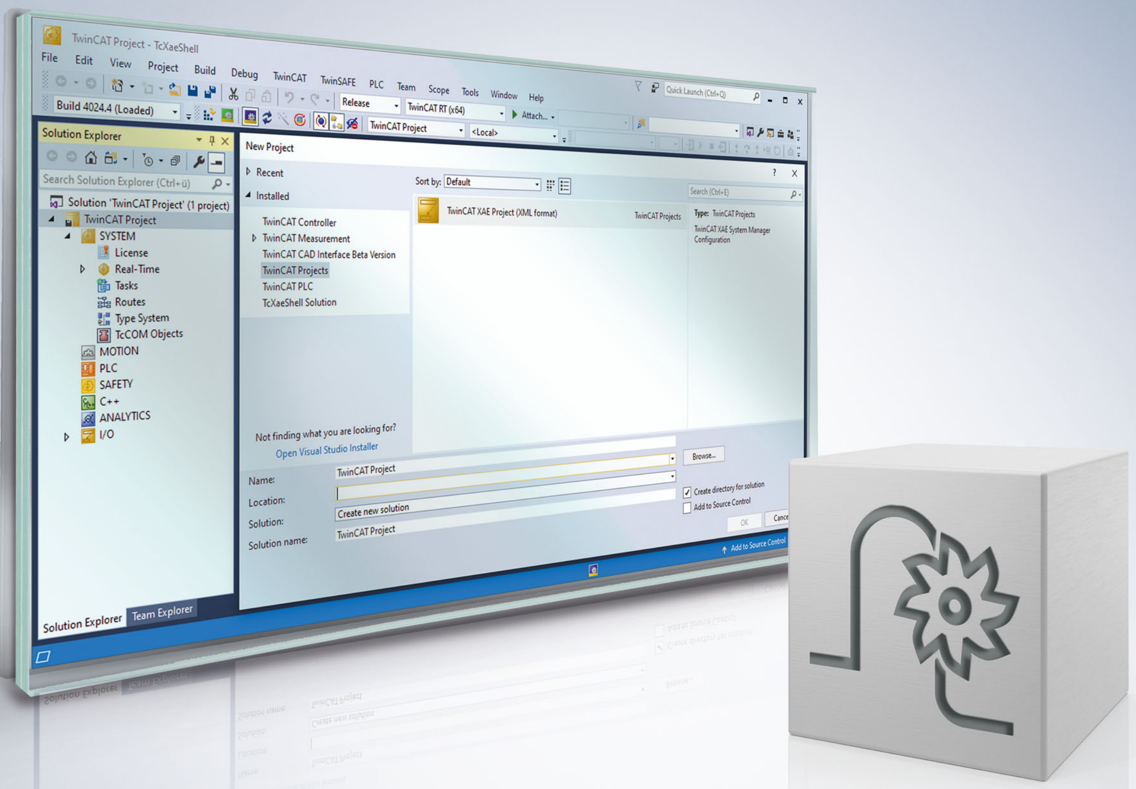
BECKHOFF New Automation Technology

Functional description | EN

TF5200 | TwinCAT 3 CNC

Service Interface



12/18/2023 | Version: 1.03

Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the 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 patent protected, in particular by the following applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

General and safety instructions

Icons used and their meanings

This documentation uses the following icons next to the safety instruction and the associated text. Please read the (safety) instructions carefully and comply with them at all times.

Icons in explanatory text

1. Indicates an action.

⇨ Indicates an action statement.

⚠ DANGER

Acute danger to life!

If you fail to comply with the safety instruction next to this icon, there is immediate danger to human life and health.

⚠ CAUTION

Personal injury and damage to machines!

If you fail to comply with the safety instruction next to this icon, it may result in personal injury or damage to machines.

NOTICE

Restriction or error

This icon describes restrictions or warns of errors.

ℹ **Tips and other notes**

This icon indicates information to assist in general understanding or to provide additional information.

General example

Example that clarifies the text.

NC programming example

Programming example (complete NC program or program sequence) of the described function or NC command.

ℹ **Specific version information**

Optional or restricted function. The availability of this function depends on the configuration and the scope of the version.

Contents

List of figures

1 Overview

Task

The Service Interface is a synchronous communication mechanism to invoke an external service from a CNC program.

The external service can execute a number of different tasks e.g.:

- Process and/or archive machine and log data.
- Send emails about the successful completion of a program.
- Read out wireless dial gauges and return measured values to the CNC program.

NOTICE

To use the Service Interface in practice, 2 software components must be developed: 1. the external service and 2. the CNC program to invoke the service. For this reason, this functional description is only suitable for experienced users or software developers.

Requirement

The Service Interface uses Beckhoff's ADS mechanism.

The requirements are therefore:

- TwinCAT 2.* with CNC Build >= V.2.11.2030.01, or
- TwinCAT 3.* with CNC Build >= V.3.1.3057.03

Mandatory note on references to other documents

For the sake of clarity, links to other documents and parameters are abbreviated, e.g. [PROG] for the Programming Manual or P-AXIS-00001 for an axis parameter.

For technical reasons, these links only function in the Online Help (HTML5, CHM) but not in pdf files since pdfs do not support cross-linking.

2 Description

The Service Interface consists of 5 V.G. variables. Each CNC channel has its own Service Interface.

V.G.SERVICE.UUID

The string variable "V.G.SERVICE.UUID” contains a unique identifier to identify the external service. Several services can listen on the Service Interface at the same time but only the service with the matching UUID will respond to the request.

V.G.SERVICE.REQUEST

The string variable "V.G.SERVICE.REQUEST” contains the request. The Service Interface places no format restrictions on this variable. Only the CNC program and the external service can correctly assign and interpret this variable.

The length of this variable is limited to 128 bytes.

V.G.SERVICE.REQUEST_STATE

For a listening external service, a value unequal to 0 in "V.G.SERVICE.REQUEST_STATE" indicates that a request was issued. The Service Interface ensures that the service only starts its task when the value of the variable is unequal to 0. The rows

```
N00110 ; send request
N00120 V.G.SERVICE.REQUEST_STATE = 1
```

ensure that a listening external service with matching UUID can start its work.

The variable "V.G.SERVICE.REQUEST_STATE” in the Service Interface has no other semantic meaning. To simplify processes, this Service Interface sets this variable to 0 when the external service has finished its task.

V.G.SERVICE.RESPONSE_STATE

The variable "V.G.SERVICE.RESPONSE_STATE" is the counterpart of the "V.G.SERVICE.REQUEST_STATE” for the response. Typically, this variable is written by the external service after it has finished its task. The CNC program waits for the variable V.G.SERVICE.RESPONSE to assume a particular value before continuing its process. For example, this variable can be used to return error codes or results. In the CNC program above, the rows

```
N00140 ; wait for response
N00150 #WAIT FOR V.G.SERVICE.RESPONSE_STATE != 0
```

ensure that the CNC program waits until the external service finishes its task. The external service must be implemented accordingly, i.e. it must return a value of unequal to 0 when it ends.

Otherwise, the variable V.G.SERVICE.RESPONSE_STATE has no other semantic meaning in the Service Interface. To simplify processes, the Service Interface sets this variable to 0 when the CNC program writes the variable "V.G.SERVICE.REQUEST_STATE".

V.G.SERVICE.RESPONSE

The string variable "V.G.SERVICE.RESPONSE” contains the response of the external service. In analogy to "V.G.SERVICE.REQUEST", "V.G.SERVICE.RESPONSE” is only limited in its length (128 bytes), not in its format.

The task of the CNC program is to correctly interpret the variable.

Process

The typical process is as follows.

1. The external service is started and listens to the Service Interface from then on.
2. The CNC program prepares the request.
3. The CNC program writes the request on the Service Interface and waits for a response.
4. After the request is received, the external service executes its task.
5. After its task is finished, the external service prepares a response for the CNC program.
6. The external service writes the response back to the Service Interface.
7. The CNC program receives the response and continues its process.

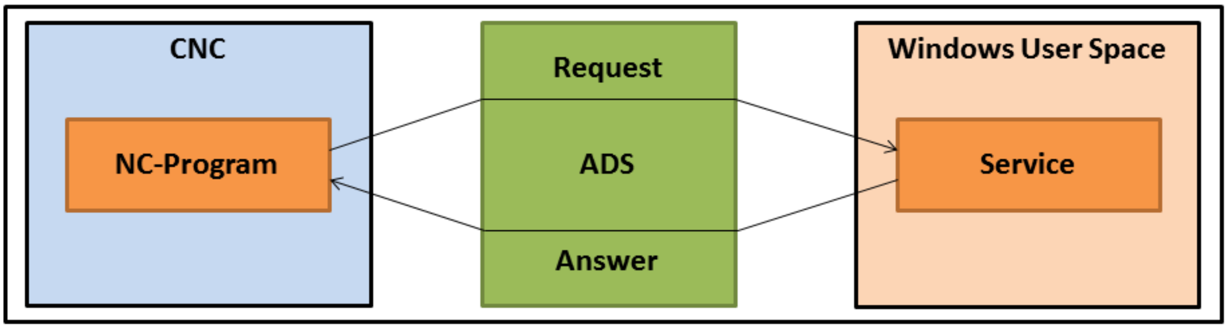


Fig. 1: Process in Service Interface

Process in Service Interface

```
N00010 ; move
N00020 G00 X10

N00040 ; wait for end of movement
N00050 #FLUSH CONTINUE

N00070 ; prepare request for Service Interface
N00080 V.G.SERVICE.UUID = "560AACBF-6335-44a9-895A-B27F3ED5E47"
N00090 V.G.SERVICE.REQUEST = "3"

N00110 ; send request
N00120 V.G.SERVICE.REQUEST_STATE = 1

N00140 ; wait for response
N00150 #WAIT FOR V.G.SERVICE.RESPONSE_STATE != 0

N00170 ; process response
N00180 #MSG SAVE EXCLUSIVE ["response = %s", V.G.SERVICE.RESPONSE]

N00200 ; continue processing
N00210 G00 X20

N00230 ; end
N00240 M30
```

Square root service - define service

```
//
// Service function that interprets the request as numeral
// and returns the square root of the numeral as response.
//
ISG_SERVICE_CODE squareRoot(ISG_SERVICE * service, char * request, ADS_UINT32 requestState, char *
response, size_t maxResponseSize, ADS_UINT32 * responseState)
{
  // interpret the request as numeral
  double x = atof(request);

  // check whether the numeral is non-negative
  if (x >= 0)
  {
    // if yes, calculate the square root
    double y = sqrt(x);

    // convert root into string
    // and write quotient in response buffer
    sprintf_s(response, maxResponseSize, "%f", y);

    // set response status to 1 to signal success
    *responseState = 1;
  }
  else
  {
    // otherwise write error message in the
    // response buffer
    sprintf_s(response, maxResponseSize,
     "ERROR: cannot calc square root of negative number %f", x);

    // set response status to 99 to signal an
    // error
    *responseState = 99;
  }

  // the service function itself was not finished successfully
  return ISG_SERVICE_CODE_OK;
}
```

Start service, set up ADS connection and register service

```
//
// Sets up connection to ADS and defines
// and activates the square root service.
//
void exampleSquareRootService()
{
  // declare connection and service
  ISG_SERVICE_CONNECTION connection;
  ISG_SERVICE service;

  // set up connection to ADS
  // Assumption: SDA-Port = 552, COM-Port = 553
  isgServiceConnectionCreateADSLocal(552, 553, &connection);

  // define service
  isgServiceCreate(

    // this UUID must match with V.G.SERVICE.UUID
    "560AACBF-6335-44a9-895A-8B27F3ED5E47",

    // a simple description of the service
    "a simple square root service",

    // define actual service function
    &squareRoot,

    // define connection of service to ADS
    &connection,

    // service to be created, result buffer
    &service

  );

  // poll service interface
  isgServicePoll(&service);

};
```

Software Development Kit (SDK)

ISG provides a simple SDK to create simple services quickly in C/C++.

The SDK contains:

- Header and library files for the Service Interface
- A sample project which implements the square root service
- A sample CNC program which invokes the service
- A batch service which listens to simple Windows batch files on the Service Interface; no C development is needed

For more information on the required ADS headers/libraries, refer to the text file "sdk/readme.txt” in the zip archive.

3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Download finder

Our download finder contains all the files that we offer you for downloading. You will find application reports, technical documentation, technical drawings, configuration files and much more.

The downloads are available in various formats.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on our internet page: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963-157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963-460
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963-0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

More Information:
www.beckhoff.com/TF5200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com