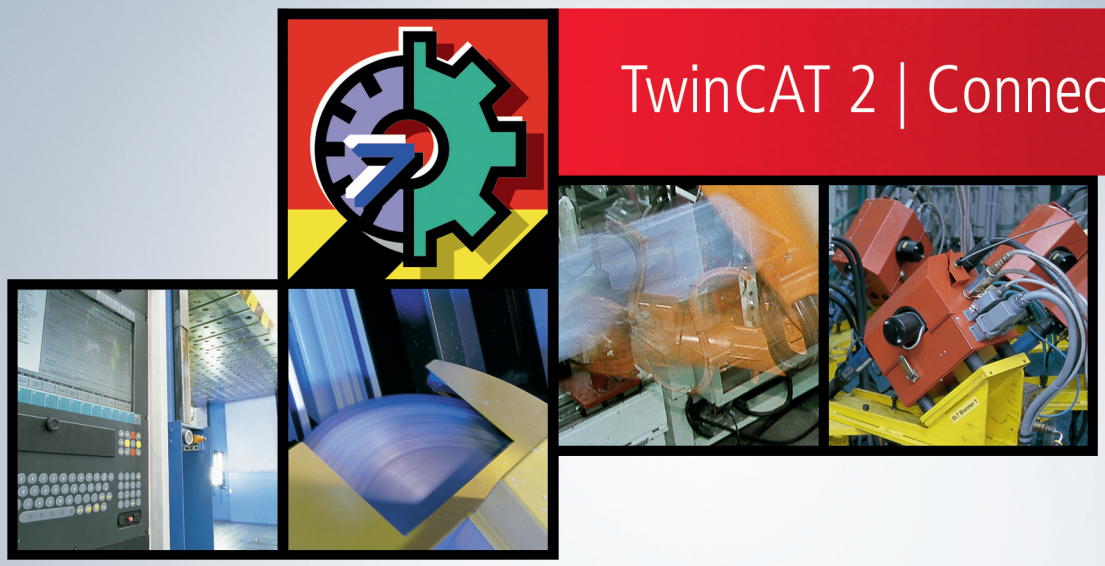
BECKHOFF New Automation Technology

Manual | EN

TX1000

TwinCAT 2 | ADS



TwinCAT 2 | Connectivity

2022-09-28 | Version: 1.2

Table of contents

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

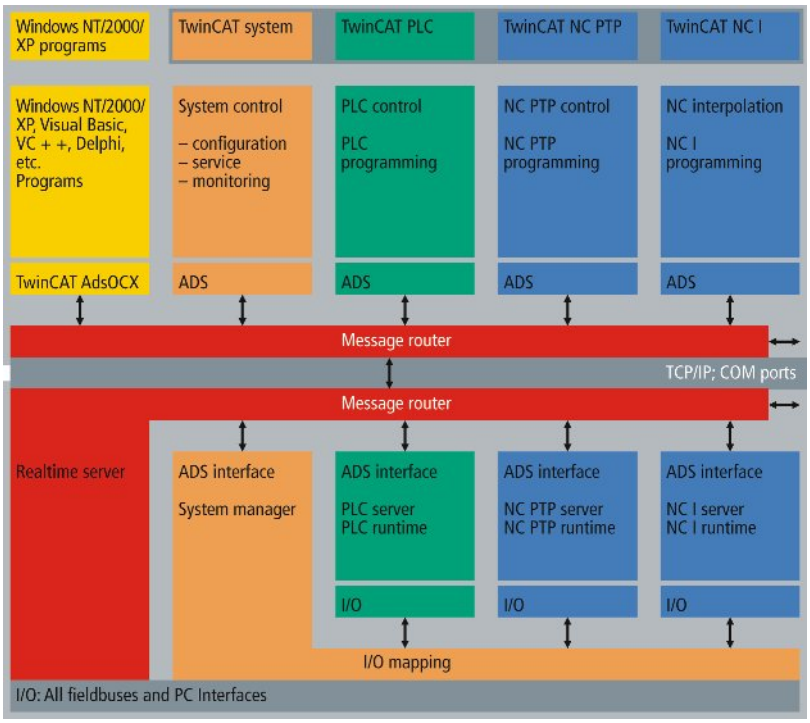
The Automation Device Specification (ADS) describes a device-independent and fieldbus-independent interface governing the type of access to ADS devices.

Device concept

The TwinCAT system architecture allows the individual modules of the software (e.g. TwinCAT PLC, TwinCAT NC, ...) to be treated as independent devices: For every task there is a software module ("Server" or "Client").

The messages between these objects are exchanged through a consistent ADS interface by the "message router". This manages and distributes all the messages in the system and over the TCP/IP connections. TwinCAT message routers exist on every TwinCAT PC and on every Beckhoff BCxxxx Bus Controller.

This allows all TwinCAT server and client programs to exchange commands and data.



Device definition

An object that has implemented the ADS interface (thus being accessible via ADS) and that offers "server services", is known as an ADS device. The detailed meaning of an ADS service is specific to each ADS device, and is described in the relevant ADS device documentation:

- ADS PLC device documentation [▶ 29]
- ADS NC device documentation [▶ 36]
- ADS CAM (cam controller) device documentation [▶ 145]

ADS libraries

To allow participation in ADS communication (as an ADS client or, possibly, as an ADS server) the following software objects are made available:

- "TcSystem.lib" PLC library
  can be used in the TwinCAT PLC.
- ADS-DLL
  for use under e.g. C/C++, Delphi, etc.
- ADS.NET
  for use under e.g. VB.NET, C#, Delphi.NET, Delphi Prism etc.
- ADS-OCX (ActiveX-Control)
  for use under e.g. Visual Basic, C++, Delphi, etc.
- ADS-Script-DLL
  for use under e.g. VBScript, JScript, etc.
- ADS-WebService
  ADS services via HTTP for use under e.g. Visual C#, Delphi.NET, etc.
- ADS-Java-DLL

3 TwinCAT ADS/AMS-Specification

The following description reveals the specification for the ADS/AMS protocol via TCP/IP. Thereby it´s possible to realise own implementations of ADS/AMS, e.g for other operating systems than windows, or for not supported programming languages.

The communication implements via TCP/IP, Port 48898 (0xBF02). The principle operating sequence of the communication is described in the documentation TwinCAT ADS under 'ADS Introduction [▶ 8]'.

3.1 Client-server relationship

ADS services can initially be categorized into "confirmed" and "unconfirmed" services.

ADS-Client	ADS-Server
Request ->	-> Indication
Confirmation <-	<- Response

The progress of an ADS communication begins with an ADS request, which arrives at the ADS server where it is viewed as an ADS indication.

The ADS server replies with an ADS response, which in turn is received by the ADS client as an ADS confirmation.

Messages that are sent autonomously by an ADS server (e.g. error or other status messages) are registered by the ADS client as a notification indication.

General ADS services:

The general ADS communication services are classified into

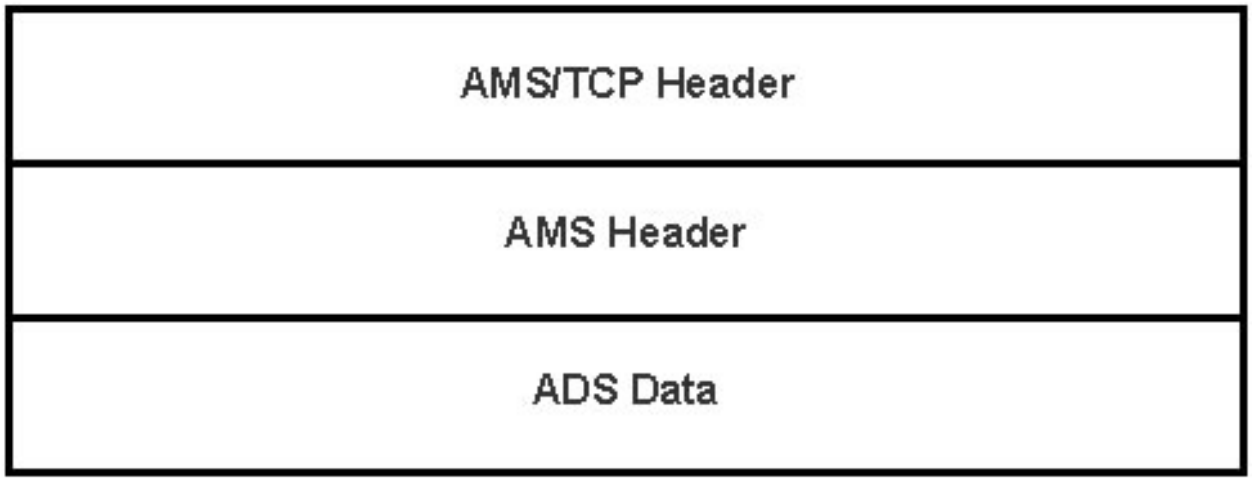
- Asynchronous
  - the client makes the ADS request to the server
  - the client continues to operate (without the ADS confirmation)
  - the server processes the ADS request, and provides the client with the result with a call-back (ADS confirmation to the client)
- Notification
  - the client registers itself with an ADS request to the server for a specific service
  - the server provides the service to the client autonomously by call-back (ADS confirmation to the client), until the client cancels its request for the service
  - the advantage of this kind of communication is the low additional ADS protocol overhead, since cyclical ADS requests from the client program are not occurring

Specific ADS services:

Over and above the general ADS services, additional functions that encapsulate ADS communications and permit operation with, e.g., Visual Basic or Visual C++ are defined. These "specific ADS services " are implemented in ADS-OCX or in the ADS-DLL, and provide, for instance, facilities for synchronous communication, or consider restrictions that might exist (in Visual Basic, for example, variable types with no leading sign).

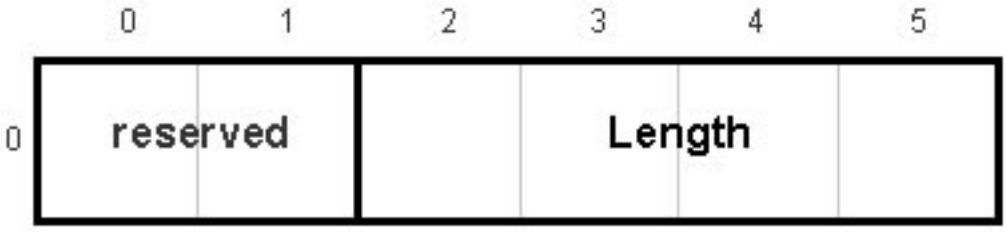
- Synchronous
  - the client makes the ADS request to the server
  - the client thread that makes the call is suspended for the duration of the ADS communication
  - when the ADS request call returns, the result from the ADS server is already available
  - the advantage of this kind of communication is the low administrative overhead required from the client program

3.2 AMS/TCP Packet



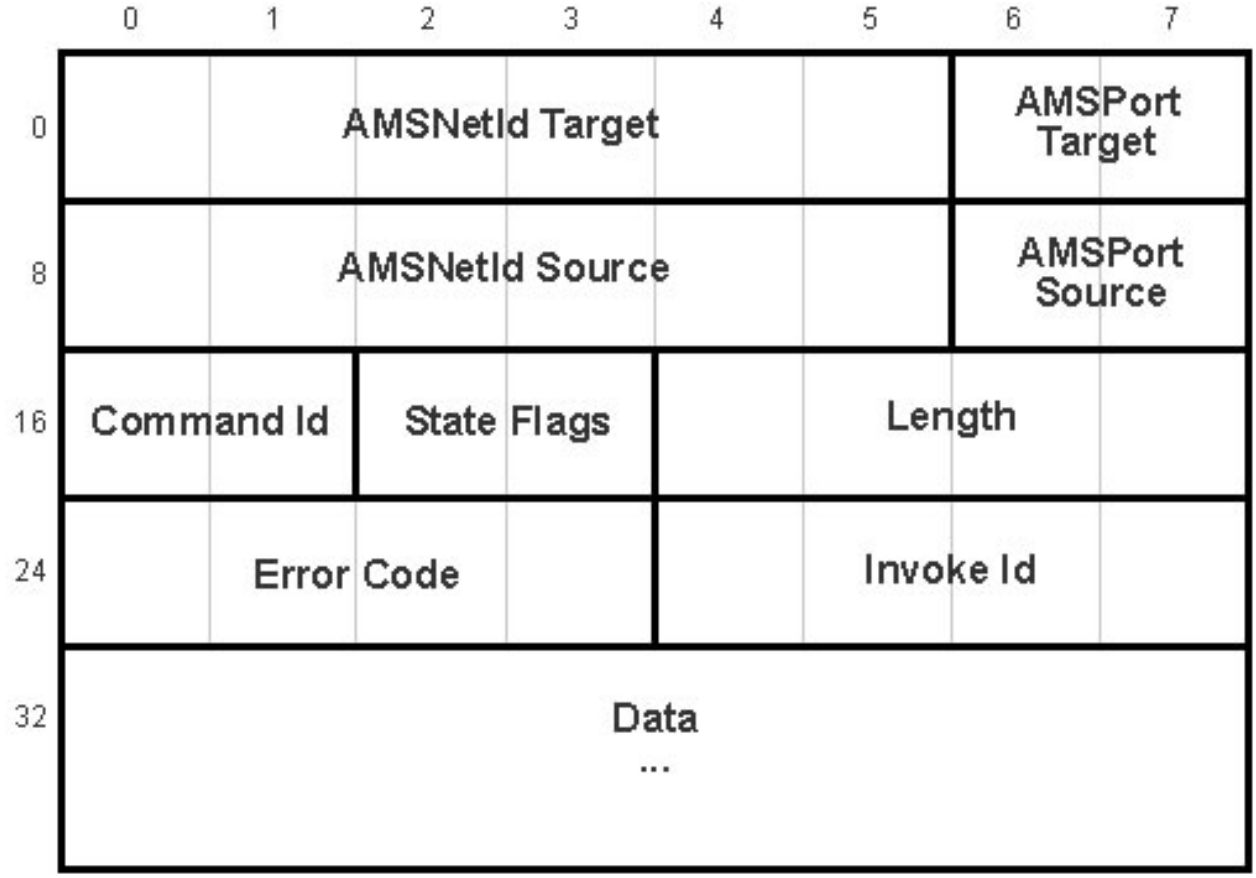
Data array	Size	Description
AMS/TCP Header	6 bytes	contains the length of the data packet.
AMS Header	32 bytes	The AMS/TCP-Header contains the addresses of the transmitter and receiver. In addition the AMS error code , the ADS command Id and some other information.
ADS Data	n bytes	The ADS data range contains the parameter of the single ADS commands. The structure of the data array depends on the ADS command. Some ADS commands require no additional data.

3.2.1 AMS/TCP Header



Data array	Size	Description
reserved	2 bytes	These bytes must be set to 0.
Length	4 bytes	This array contains the length of the data packet. It consists of the AMS-Header and the enclosed ADS data. The unit is bytes.

3.2.2 AMS Header

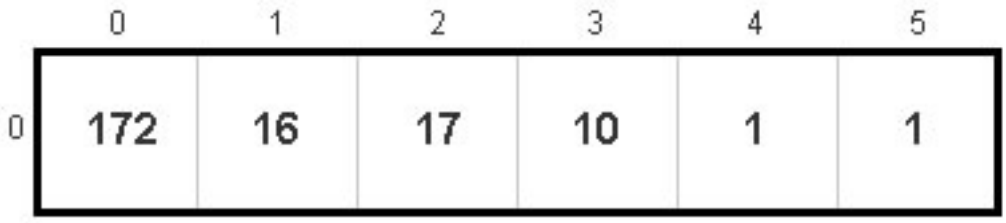


Data array	Size	Description
AMSNetId Target	6 bytes	This is the AMS NetId of the station, for which the packet is intended. Remarks see below [▶ 12].
AMSPort Target	2 bytes	This is the AMSPort of the station, for which the packet is intended.
AMSNetId Source	6 bytes	This contains the AMSNetId of the station, from which the packet was sent.
AMSPort Source	2 bytes	This contains the AMSPort of the station, from which the packet was sent.
Command Id	2 bytes	see below [▶ 13].
State Flags	2 bytes	see below [▶ 13].
Data Length	4 bytes	Size of the data range. The unit is byte.
Error Code	4 bytes	AMS error number. See ADS Return Codes.
Invoke Id	4 bytes	Free usable 32 bit array. Usually this array serves to send an Id. This Id makes is possible to assign a received response to a request, which was sent before.
Data	n bytes	Data range. The data range contains the parameter of the considering ADS commands.

AMS Net Id

AMS Net Id

The AMSNetId consists of 6 bytes and addresses the transmitter or receiver. One possible AMSNetId would be e.g.. 172.16.17.10.1.1. The storage arrangement in this example is as follows:



The AMSNetId is purely logical and has usually no relation to the IP address. The AMSNetId is configurated at the target system. At the PC for this the TwinCAT System Control is used. If you use other hardware, see the considering documentation for notes about settings of the AMS NetId.

Command Id

Cmd	Description
0x0000	Invalid
0x0001	ADS Read Device Info [▶ 14]
0x0002	ADS Read [▶ 14]
0x0003	ADS Write [▶ 15]
0x0004	ADS Read State [▶ 16]
0x0005	ADS Write Control [▶ 17]
0x0006	ADS Add Device Notification [▶ 18]
0x0007	ADS Delete Device Notification [▶ 19]
0x0008	ADS Device Notification [▶ 19]
0x0009	ADS Read Write [▶ 21]

Other commands are not defined or are used internally. Therefore the *Command Id* is only allowed to contain the above enumerated values!

State Flags

Flag	Description
0x0001	0: Request / 1: Response
0x0004	ADS command

The first bit marks, whether it´s a request or response. The third bit must be set to 1, to exchange data with ADS commands. The other bits are not defined or were used for other internal purposes.

Therefore, the other bits must be set to 0!

Flag	Description
0x000x	TCP Protocol
0x004x	UDP Protocol

Bit number 7 marks, if it should be transfered with TCP or UDP.

3.2.3 ADS Commands

Command	Description
ADS Read Device Info [▶ 14]	Reads the name and the version number of the ADS device.
ADS Read [▶ 14]	With *ADS Read* data can be read from an ADS device
ADS Write [▶ 15]	With *ADS Write* data can be written to an ADS device.

Command	Description
ADS Read State [▶ 16]	Reads the ADS status and the device status of an ADS device.
ADS Write Control [▶ 17]	Changes the ADS status and the device status of an ADS device.
ADS Add Device Notification [▶ 18]	A notification is created in an ADS device.
ADS Delete Device Notification [▶ 19]	One before defined notification is deleted in an ADS device.
ADS Device Notification [▶ 19]	Data will carry forward independently from an ADS device to a Client
ADS Read Write [▶ 21]	With *ADS ReadWrite data will be written to an ADS device. Additionally, data can be read from the ADS device.*

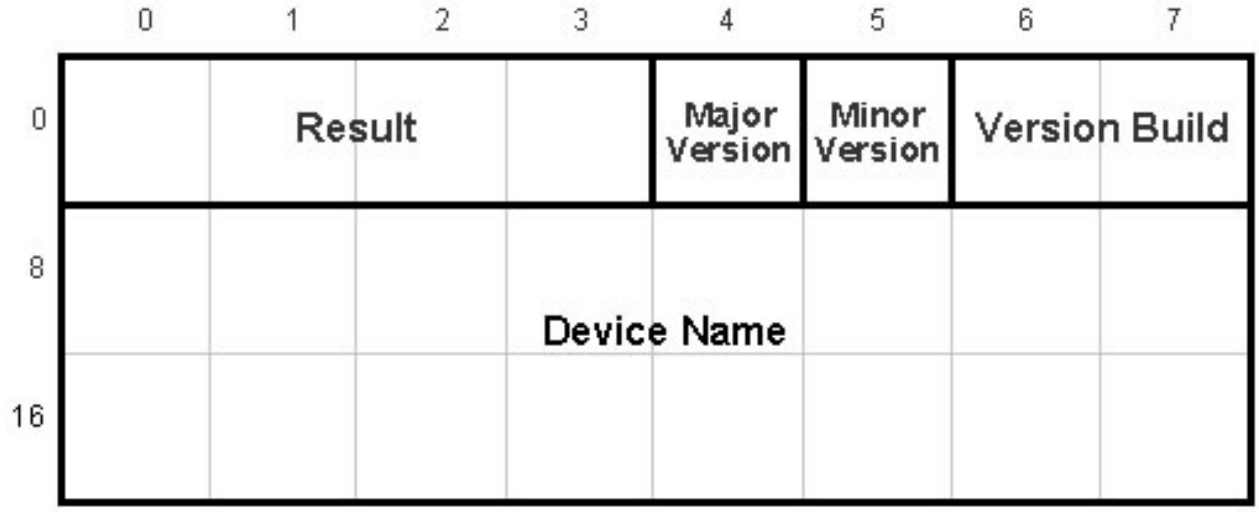
3.2.3.1 ADS Read Device Info

Reads the name and the version number of the ADS device.

Request

No additional data required

Response

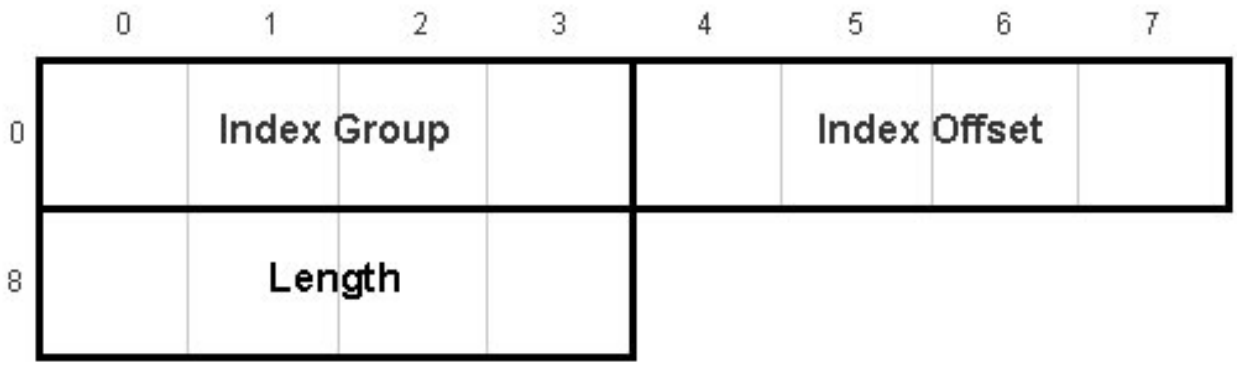


Data array	Size	Description
Result	4 bytes	ADS error number.
Major Version	1 byte	Major version number
Minor Version	1 byte	Minor version number
Version Build	2 bytes	Build number
Device Name	16 bytes	Name of ADS device

3.2.3.2 ADS Read

With *ADS Read* data can be read from an ADS device. The data are addressed by the *Index Group* and the *Index Offset*

Request



Data array	Size	Description
Index Group	4 bytes	Index Group of the data which should be read.
Index Offset	4 bytes	Index Offset of the data which should be read.
Length	4 bytes	Length of the data (in bytes) which should be read.

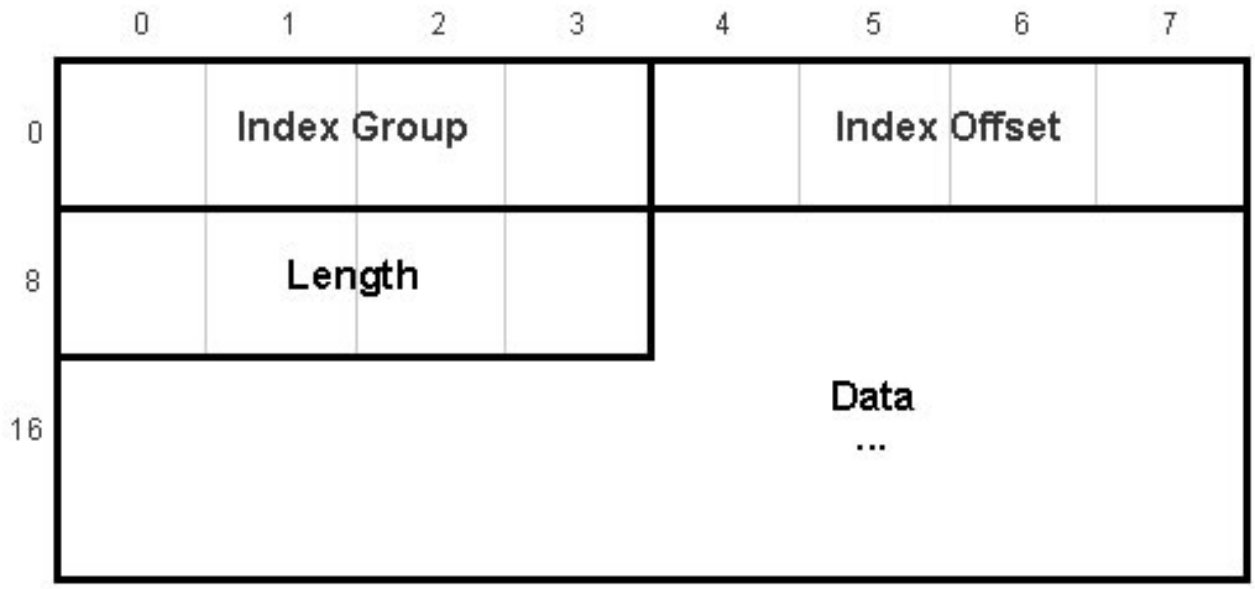
Response

Data array	Size	Description
Result	4 bytes	ADS error number
Length	4 bytes	Length of data which are supplied back.
Data	n bytes	Data which are supplied back.

3.2.3.3 ADS Write

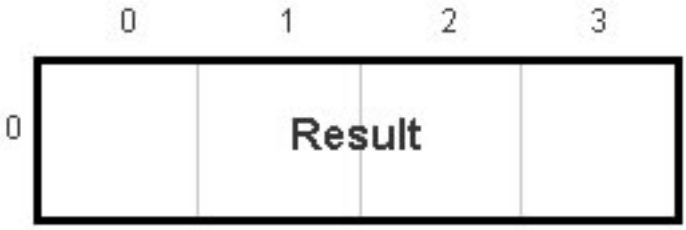
With *ADS Write* data can be written to an ADS device. The data are addressed by the *Index Group* and the *Index Offset*

Request



Data array	Size	Description
Index Group	4 bytes	Index Group in which the data should be written
Index Offset	4 bytes	Index Offset, in which the data should be written
Length	4 bytes	Length of data in bytes which are written
Data	n bytes	Data which are written in the ADS device.

Response



Data array	Size	Description
Result	4 bytes	ADS error number

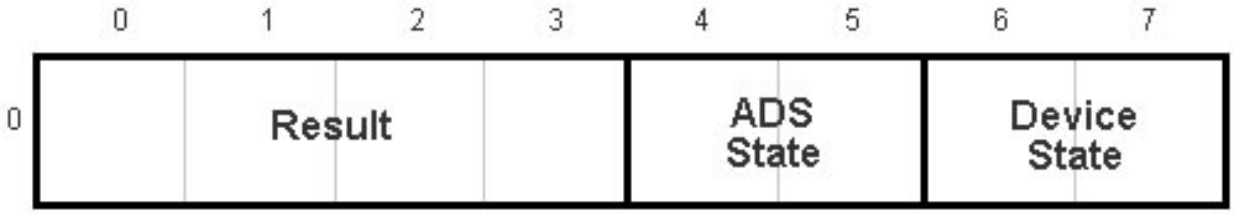
3.2.3.4 ADS Read State

Reads the ADS status and the device status of an ADS device.

Request

No additional data required

Response

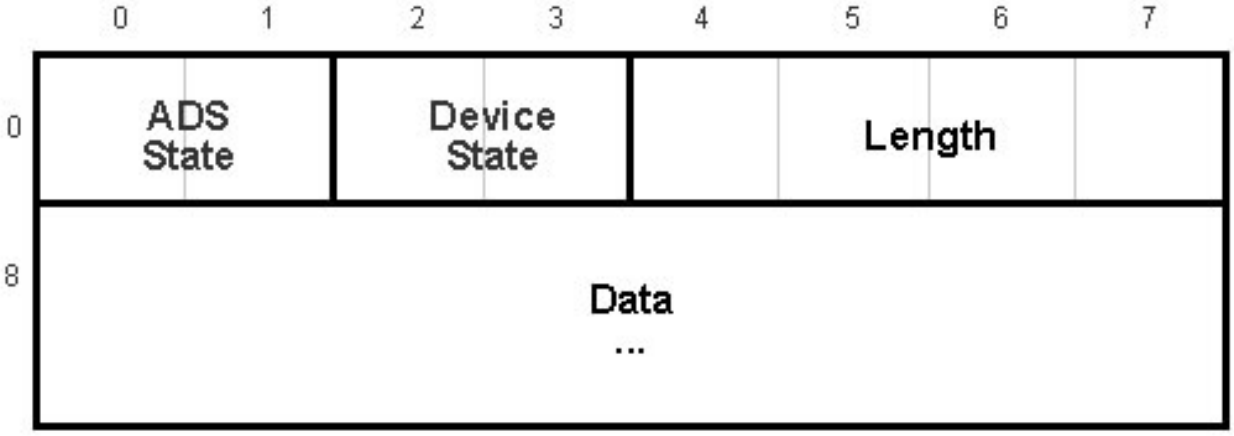


Data array	Size	Description
Result	4 bytes	ADS error number.
ADS State	2 bytes	ADS status (see data type ADSSTATE of the ADS-DLL).
Device State	2 bytes	Device status

3.2.3.5 ADS Write Control

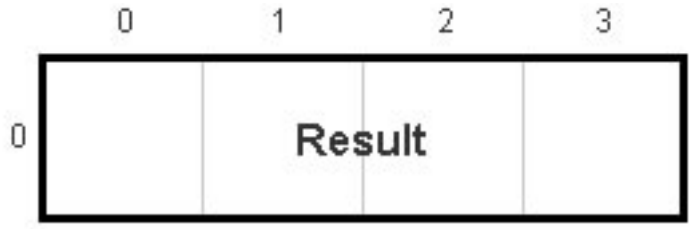
Changes the ADS status and the device status of an ADS device. Additionally, it is possible to send data to the ADS device to transfer further information. These data were not analyzed from the current ADS devices (PLC, NC, ...)

Request



Data array	Size	Description
ADS State	2 bytes	New ADS status (see data type ADSSTATE of the ADS-DLL).
Device State	2 bytes	New device status.
Length	4 bytes	Length of data in byte.
Data	n bytes	Additional data which are sent to the ADS device

Response



Data array	Size	Description
Result	4 bytes	ADS error number.

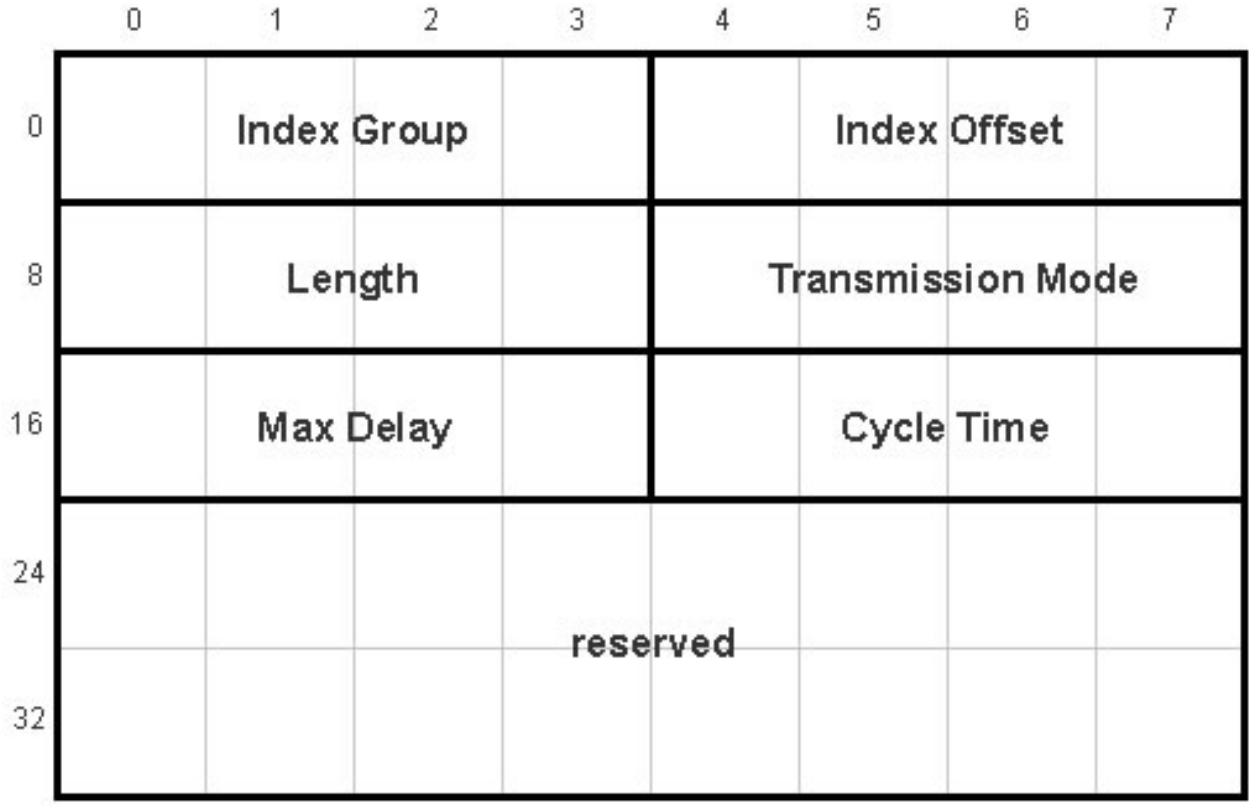
3.2.3.6 ADS Add Device Notification

A notification is created in an ADS device.

Increase notifications

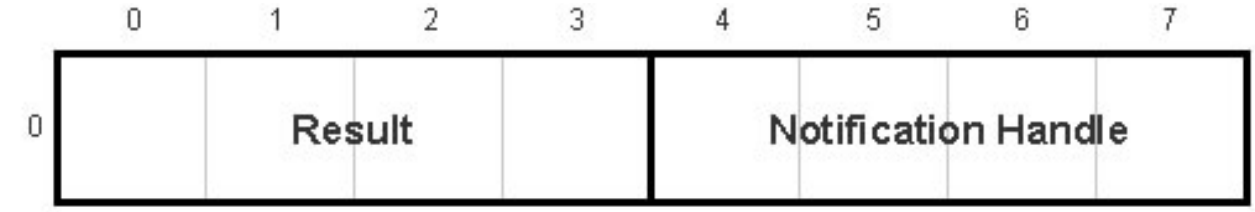
We recommend not to register more than 550 notifications per device. By organizing the data into structures, the payload per notification can be effectively increased.

Request



Data array	Size	Description
Index Group	4 bytes	Index Group of the data, which should be sent per notification.
Index Offset	4 bytes	Index Offset of the data, which should be sent per notification.
Length	4 bytes	Length of data in bytes, which should be sent per notification.
Transmission Mode	4 bytes	See description of the structure ADSTRANSMODE at the ADS-DLL.
Max Delay	4 bytes	At the latest after this time, the *ADS Device Notification* is called. The unit is 1ms.
Cycle Time	4 bytes	The ADS server checks if the value changes in this time slice. The unit is 1ms
reserved	16bytes	Must be set to 0

Response



Data array	Size	Description
Result	4 bytes	ADS error number
Notification Handle	4 bytes	Handle of notification

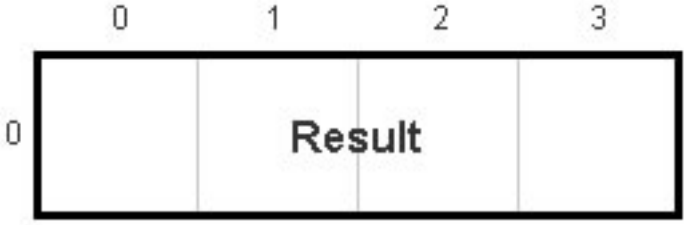
3.2.3.7 ADS Delete Device Notification

One before defined notification is deleted in an ADS device.

Request

Data array	Size	Description
Notification Handle	4 bytes	Handle of notification. The handle is created by the ADS command *Add Device Notification*

Response



Data array	Size	Description
Result	4 bytes	ADS error number

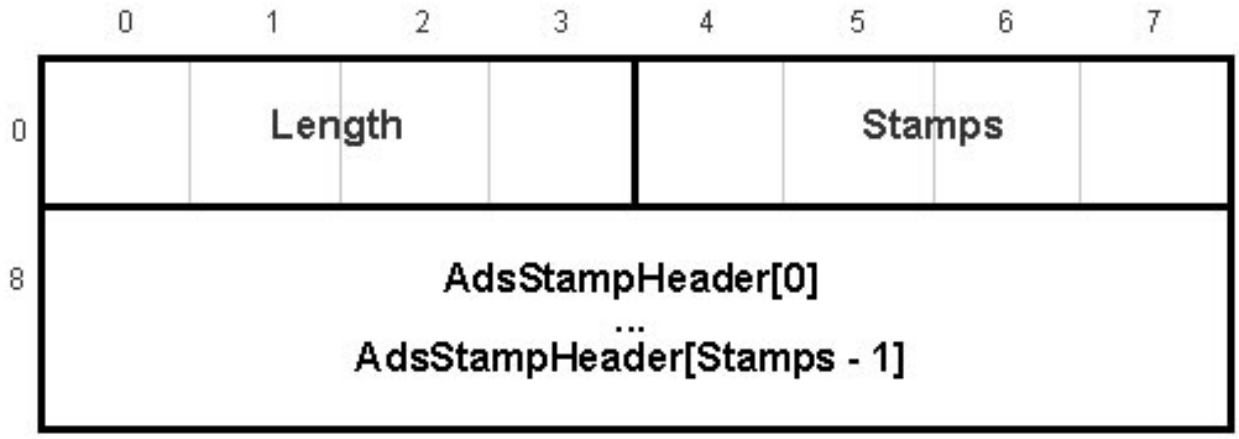
3.2.3.8 ADS Device Notification

Data will carry forward independently from an ADS device to a Client.

Request

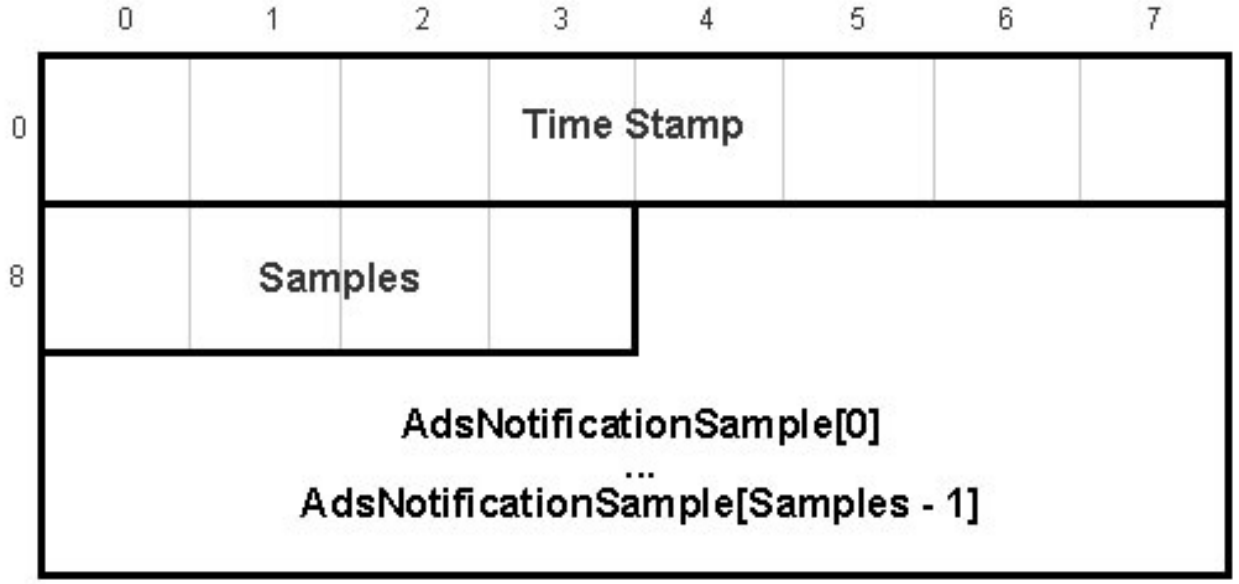
The data which are transferred at the *Device Notification* are multiple nested into one another. The *Notification Stream* contains an array with elements of type *AdsStampHeader.* This array again contains elements of type *AdsNotificationSample*.

AdsNotificationStream



Data array	Size	Description
Length	4 bytes	Size of data in byte.
Stamps	4 bytes	Number of elements of type AdsStampHeader [▶ 20].
AdsStampHeader	n bytes	Array with elements of type AdsStampHeader [▶ 20].

AdsStampHeader



Data array	Size	Description
TimeStamp	8 bytes	The timestamp is coded after the Windows FILETIME format. I.e. the value contains the number of the nano seconds, which passed since 1.1.1601. In addition, the local time change is not considered. Thus, the time stamp is present as universal Coordinated time (UTC).
Samples	4 bytes	Number of elements of type AdsNotificationSample [▶ 20].
AdsNotificationSample	n bytes	Array with elements of type AdsNotificationSample [▶ 20].

AdsNotificationSample

AdsNotificationSample

Data field	Size	Description
Notification Handle	4 bytes	Notification handle.
Sample Size	4 bytes	Size of the data area in bytes.
Data	n bytes	Data.

Single notification

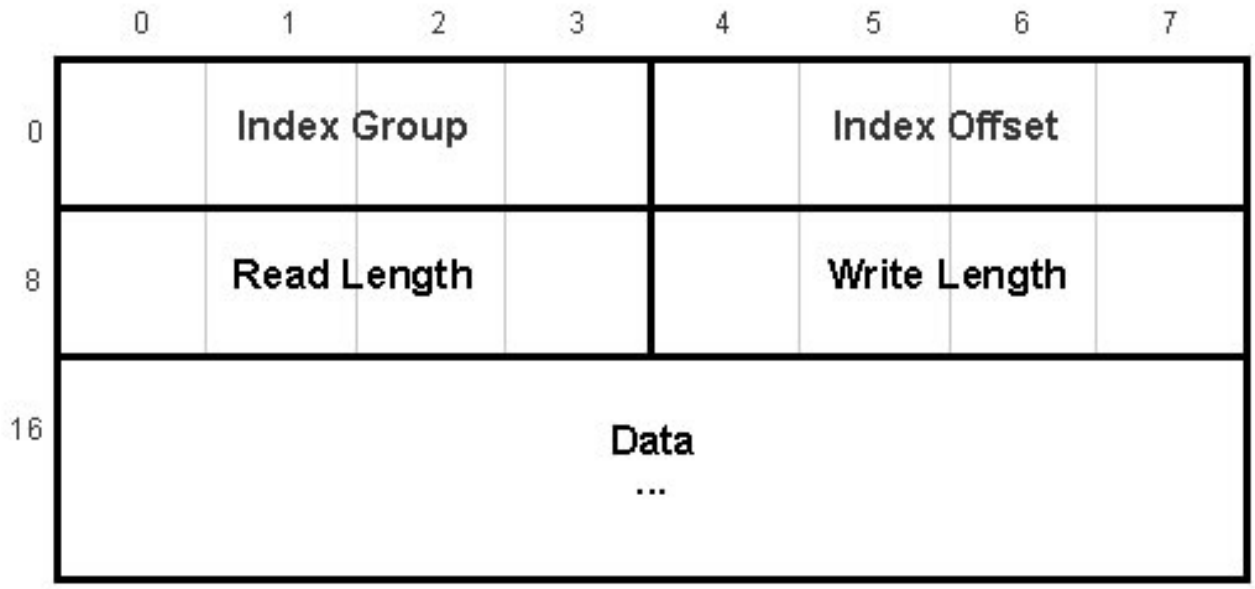
If the handle has become invalid, a notification without data is sent once as a note.

3.2.3.9 ADS Read Write

With *ADS ReadWrite* data will be written to an ADS device. Additionally, data can be read from the ADS device.

The data which can be read are addressed by the *Index Group* and the *Index Offset*

Request



Data array	Size	Description
Index Group	4 bytes	Index Group, in which the data should be written.
Index Offset	4 bytes	Index Offset, in which the data should be written
Read Length	4 bytes	Length of data in bytes, which should be read.

Data array	Size	Description
Write Length	4 bytes	Length of data in bytes, which should be written
Data	n bytes	Data which are written in the ADS device.

Response

Data array	Size	Description
Result	4 bytes	ADS error number
Length	4 bytes	Length of data which are supplied back.
Data	n bytes	Data which are supplied back.

4 TwinCAT ADS/AMS (serial) Specification

The following description reveals the specification for the AMS protocol via RS232. Thereby it´s possible to realise own implementations of AMS via RS232, e.g for other operating systems than windows.

The communication implements via RS232.
The principle operating sequence of the communication is described in the documentation TwinCAT ADS under 'ADS Introduction [▶ 8]'.

The specification of the ADS/AMS protocol is described in the documentation under TwinCAT ADS/AMS - Specification [▶ 10]'.

4.1 AMS serial reset frame configuration

Data field	Size in bytes	Value range/Default	Description
Magic Cookie	2	0xA501/0xA501	Id for detecting an AMS serial frame
Transmitter address	1	0..255 / 0	Address of the sending participant. This value can always be set to 0 for an RS232 communication, since it is a 1 to 1 connection and hence the participants are unique.
Receiver address	1	0..255 / 0..255	Receiver's address. This value can always be set to 0 for an RS232 communication, since it is a 1 to 1 connection and hence the participants are unique.
Fragment number	1	0..255 / 0	Number of the frame sent. Once the number 255 has been sent, it starts again from 0. The receiver checks this number with an internal counter.
User data length	1	0..255 / 0..255	The max. length of the AMS packet to be sent is 255. If larger AMS packets are to be sent then they have to be fragmented (not published at the moment).
User data	n		The AMS packet to be sent.
CRC	2	0..65535	Frame checksum

An AMS packet can be transferred via RS232 with the help of an AMS serial frame. The actual AMS packet is in the user data field of the frame. The max. length of the AMS packet is limited to 255 bytes. Therefore the max. size of an AMS serial frame is 263 bytes. The fragment number is compared with an internal counter by the receiver. The frame number is simply accepted and not checked when receiving the first AMS frame or in case a timeout is exceeded. The CRC16 algorithm is used for calculating the checksum.

If an AMS serial frame has been received and the frame is OK (magic cookie OK, CRC OK, correct fragment number etc.), then the receiver has to send an acknowledge frame, to inform the transmitter that the frame has arrived.

Data field	Size in bytes	Value range/Default	Description
Magic Cookie	2	0x5A01/0x5A01	Id for detecting an AMS serial acknowledge frame
Transmitter address	1	0..255 / 0	Dedicated address.
Receiver address	1	0..255 / 0..255	Address of the participant who sent the original frame.
Fragment number	1	0..255 / 0	Corresponds to the fragment number transmitted by the transmitter.
User data length	1	0/ 0	Is always 0 since no user data is sent.
CRC	2	0..65535	Frame checksum

In case the transmitter does not receive a valid acknowledgement after multiple transmission, then a reset frame is sent. In this way the receiver is informed that a new communication is running, and the receiver then accepts the fragment number during the next AMS-Frame, without carrying out a check.

Data field	Size in bytes	Value range/Default	Description
Magic Cookie	2	0xA503 / 0xA503	Id for detecting an AMS serial reset frame
Transmitter address	1	0..255 / 0	Transmitter address.
Receiver address	1	0..255 / 0..255	Receiver's address.
Fragment number	1	0/0	Is always 0.
User data length	1	0/ 0	Is always 0 since no user data is sent.
CRC	2	0..65535	Frame checksum

4.2 Serial communication process

The transmitter carries out the following steps:

- Application sends AMS command
- If the AMS command is greater than 255 bytes, an error is sent to the application.
- An AMS serial frame is allocated and the AMS command is copied into the user data field. The user data length field is set to the size of the AMS command.
- The frame is sent via the serial interface and acknowledgement is waited for.
- In case the transmitter does not receive a valid acknowledgement after multiple transmission, then a reset frame is sent.

The receiver carries out the following steps:

The serial interface is exported in case data is received:

- Check the magic cookies
- Check the CRC
- Check whether the fragment number agrees with the internal counter. If this is the first received frame, or if no more frames are received after a certain time, then the frame number is simply accepted and not checked.
- If a valid frame has been received:
  - Send an acknowledgement to the transmitter
  - Forward user data (AMS telegram) to the application

Since ADS is only a higher-level protocol from AMS, the same is also valid for the sending of ADS-commands. The ADS command is in the data range of the AMS commands, that can then be sent via the serial interface.

4.3 Example

Terminal --> PLC : Request of 2 bytre data

```
Magic Cookie        01h/A5h
Sender              00h
Empfaenger          00h
Fragmentnummer      06h
Datenlaenge         2Ch
```

```
NetID Empfaenger    C0h/A8h/64h/AEh/01h/01h
Port Nummer         21h/03h
NetID Sender        C0h/A8h/64h/9Ch/01h/01h
Portnummer          01h/80h
Kommando lesen      02h/00h
Status              04h/00h
Anzahl Datenbyte    0Ch/00h/00h/00h/
Fehlercode          00h/00h/00h/00h
InvokeID            07h/00h/00h/00h
Index Gruppe        05h/F0h/00h/00h
Index Offset        04h/00h/00h/9Dh
Anzahl Byte         02h/00h/00h/00h
Checksumme          xxh/xxh/
```

PLC --> Terminal : Acknowledge:

```
Magic Cookie        01h/5Ah
Sender              00h
Empfaenger          00h
Fragmentnummer      06h
Datenlaenge         00h
Checksumme          67h/5Ah
```

PLC sends data:

```
Magic Cookie        01h/A5h
Sender              00h
Empfaenger          00h
Fragmentnummer      ECh
Anzahl Daten        2Ah
NetID Empfaenger    C0h/A8h/64h/9Ch/01h/01h
Portnummer          01h/80h
NetID Sender        C0h/A8h/64h/AEh/01h/01h
Portnummer          21h/03h
Response Lesen      02h/00h
Status              05h/00h
Anzahl Daten        0Ah/00h/00h/00h
Fehlercode          00h/00h/00h/00h
InvokeID            07h/00h/00h/00h
Ergebnis            00h/00h/00h/00h
Anzahl Daten        02h/00h/00h/00h
Daten               AFh/27h
Checksumme          04h/A9h/
```

Terminal --> PLC : Acknowledge:

5 TwinCAT ADS-Device-Documentation

An object that has implemented the ADS interface (thus being accessible via ADS) and that offers "server services", is known as an ADS device. The detailed meaning of an ADS service is specific to each ADS device, and is described in the relevant ADS device documentation:

- TwinCAT ADS Interface PLC [▶ 29]
- TwinCAT ADS Interface NC [▶ 36]
- TwinCAT ADS Device CAM [▶ 145] (cam controller)
- TwinCAT ADS COM Server for ControlNET (with SST 5136 CN PC Card)
- .....

5.1 TwinCAT System Service

The TwinCAT System Service offers an interface for the TwinCAT server for Windows NT services. At present, data access and "System Shutdown" are supported. Refer to the documentation for the ADS DLL and for the ADS OCX for further explanations of ADS services.

File access

Files can be opened or closed, read, and written via ADS index group and ADS index offset. The index groups and index offsets are described below.

Process	Description
Open/create file Indexgroup: SYSTEMSERVICE_OPENCREATE= 100 Indexoffset: SYSTEMSERVICE_OPENGENERIC = 1	The stated file is opened for read and write access. If the file does not exist, a file is created and opened. If the file exists, it is opened, and the contents are deleted. ADS Service: ADS Write Parameter: Length of the path name Path name as a character string Return value: If successful: file handle for the opened file In the event of an error: -1
Opening file for reading Indexgroup: SYSTEMSERVICE_OPENREAD = 101 Indexoffset: SYSTEMSERVICE_OPENGENERIC = 1	The stated field is opened for read access. If the file does not exist, the service returns the value -1. ADS service: ADS Write Parameter: Length of the path name Path name as a character string Return value If successful: file handle for the opened file In the event of an error: -1
Opening file for writing Indexgroup: SYSTEMSERVICE_OPENWRITE = 102 Indexoffset: SYSTEMSERVICE_OPENGENERIC = 1	The stated file is opened for read and write access. If the file does not exist, the service returns the value -1. If the file exists. it is opened and the contents are deleted. ADS service: ADS Write Parameter: Length of the path name Path name as a character string Return value If successful: file handle for the opened file In the event of an error: -1

Process	Description
Close file Indexgroup: File Handle Indexoffset: SYSTEMSERVICE_NOSEEK = -1	The file which belongs to the File Handle is closed. It must be noted that the data length for this service must be 0! ADS service: ADS Write Data length: 0
Write data to file Indexgroup: File Handle Indexoffset: SYSTEMSERVICE_NOSEEK = -1 or Seek Offset in byte	The number of bytes stated in the data length is written to the file. If an index offset is not equal to -1, the index offset is interpreted as the offset in the file. The number of written bytes is returned by this service in the reply. In the event of an error, the service returns the value -1. ADS service: ADS Write Data length: number of bytes to be written.
Read data from file Indexgroup: File Handle Indexoffset: SYSTEMSERVICE_NOSEEK = -1 or Seek Offset in byte	The number of bytes stated in the data length is read from the file. If the index offset is unequal to -1, the index offset is interpreted as the offset in the file. With a data length of 0, only the file indicator in the file is offset by the value in the index offset. The number of read bytes is returned by this service in the reply. In the event of an error, the services returns the value -1. ADS service: ADS Read Data length: number of bytes to be read.
System shutdown	ADS WriteControl enables the TwinCAT System Service to be instructed to shut down the operating system. ADS service: ADS WriteControl AdsState: ADSSTATE_SHUTDOWN = 12 DeviceState: Shutdown of Windows NT with DEVICESTATE = 0, restart of Windows NT with DEVICESTATE = 1 DeviceData: timeout in seconds until system shutdown Data length: 4 bytes

5.2 TwinCAT ADS Interface: Scope

The TwinCAT Scope provides an ADS-interface to act like an ADS-device. The commands described here can be used to operate Scope View.

Since a fixed ADS port is required for this, only the first instance of the Scope can be operated via ADS.

TwinCAT Scope : ADS port **14000** (decimal)

Index-Group (Hex)	Description	Remark
0x00001000	Scope Root States	
0x00002000	Scope Root Functions	
0x00003000	Scope View	

Scope Root States

Index-Offset (Hex)	Access	Data type	Physical unit	Scope of Definition	Description	Remark
0x00000001	Read	BYTE			Get Online Mode 0: Scope is offline 1: Scope is online	
0x00000001	Write	void			Set Scope to Online Mode	
0x00000002	Write	void			Set Scope to Offline Mode	

Scope Root Functions

Index-Offset (Hex)	Access	Data type	Physical unit	Scope of Definition	Description	Remark
0x00000001	Write	char[]			Load *.scp File (Scope Configuration Project)	D:\\TwinCAT\ \scope\ \achse2.scp

Scope View

Requirements

Index-Offset (Hex)	Access	Data type	Physical unit	Scope of Definition	Description	Remark
0x00000100 + Id	Write	void			**Manual Trigger**	Issuing this command triggers the Scope. It must, however, be online.
0x00000200 + Id	Read/Write	Real64			Record Length	Specifies the duration of the recording
0x00001000 + Id	Write	char[]			Export data as an ASCII file	D:\\TwinCAT\ \Scope\\Data\ \Test.dat
0x00001010 + Id	Write	char[]			Export ScopeView	

Note:

The Scope View properties can currently only be used if only one view is active in the application. In other words, the ID is always 1.

5.3 TwinCAT ADS Interface PLC

The PLC software can be described as a virtual field unit (Automation Device), since it is a pure software PLC. It therefore provides a Beckhoff ADS (Automation Device Specification) interface for other communication partners (e.g. other virtual field units or Windows programs), via which it can be parameterised or interrogated. Use of the ADS standardises access to the PLC and incorporates it into the range of available virtual field units.

The READ and WRITE operations take place on the PLC interface (as defined by ADS) via two numbers: the index group (16 bit) and the index offset (32 bit). The ADS interface of the PLC will be described in more detail in the following pages with regard to the group and offset indices.

Specification "Index-Group" of the PLC

The four global ranges of an ADS unit are shown as follows for the PLC as four sections in the index groups:

Index-Group (0x = hex)	Index Group description
0x00000000 0x00000FFF	reserved
0x00001000	PLC ADS parameter range
0x00002000	PLC ADS status range
0x00003000	PLC ADS unit function range
0x00004000	PLC ADS services (includes services to access PLC memory range (%M field))
0x00006000 0x0000EFFF	reserved for PLC ADS extension
0x0000F000 0x0000FFFF	general TwinCAT ADS system services (includes services to acces PLC process diagramm of the physical inputs and outputs)

5.3.1 "Index-Group/Offset" Specification of the PLC services

This section includes services to access the PLC memory range (%M field).

Index Group	Index Offset	Access	Data type	Phys. unit	Def. range	Description	Remarks
0x00004020	0x00000000- 0x0000FFFF	R/W	UINT8[n]			**READ_M - WRITE_M** PLC memory range(%M field).Offset is byte offset.	

Index Group	Index Offset	Access	Data type	Phys. unit	Def. range	Description	Remarks
0x00004021	0x00000000-0xFFFFFFFF	R/W	UINT8			**READ_MX - WRITE_MX** PLC memory range (%MX field).The low word of the index offset is the byte offset. The index offset contains the bit address calculated from the byte number *8 + bit number	
0x00004025	0x00000000	R	ULONG			**PLCADS_IGR_RMSIZE** Byte length of the process diagram of the memory range	
0x00004030	0x00000000-0xFFFFFFFF	R/W	UINT8			**PLCADS_IGR_RWRB** Retain data range. The index offset is byte offset	
0x00004035	0x00000000	R	ULONG			**PLCADS_IGR_RRSIZE** Byte length of the retain range	
0x00004040	0x00000000-0xFFFFFFFF	R/W	UINT8			**PLCADS_IGR_RWDB** Data range. The index offset is byte offset.	

Index Group	Index Off-set	Access	Data type	Phys. unit	Def. range	Descrip-tion	Remarks
0x00004045	0x00000000	R	ULONG			**PLCADS_IGR_RDSIZE** Byte length of the data range	

5.3.2 "Index-Group/Offset" Specification of the TwinCAT ADS system services

This section covers those ADS services which have identical meanings and effects with every TwinCAT ADS unit. In this section are also included services to access the PLC process diagram of the physical inputs and outputs.

Index Group	Index Offset	Access	Data type	Description	Remarks
0x0000F003	0x00000000	R&W	W: UINT8[n] R: UINT32	**GET_SYMHANDLE_BYNAME** A handle (code word) is assigned to the name contained in the write data and is returned to the caller as a result.	
0x0000F004	0x00000000			Reserved.	
0x0000F005	0x00000000- 0xFFFFFFFF =symHandle	R/W	UINT8[n]	**READ_/WRITE_SYMVAL_BYHANDLE** Reads the value of the variable identified by ‚symHdl' or assigns a value to the variable. The ‚symHdl' must first have been determined by the GET_SYMHANDLE_BYNAME services.	

Index Group	Index Offset	Access	Data type	Description	Remarks
0x0000F006	0x00000000	W	UINT32	**RELEASE_SYMHANDLE** The code (handle) contained in the write data for an interrogated, named PLC variable is released.	
0x0000F020	0x00000000-0xFFFFFFFF	R/W	UINT8[n]	**READ_I - WRITE_I** PLC process diagram of the physical inputs(%I field). Offset is byte offset.	
0x0000F021	0x00000000-0xFFFFFFFF	R/W	UINT8	**READ_IX - WRITE_IX** PLC process diagram of the physical inputs(%IX field). The index offset contains the bit address which is calculated from byte number *8 + bit number	
0x0000F025	0x00000000	R	ULONG	**ADSIGRP_IOIMAGE_RISIZE** Byte length of the PLC process diagram of the physical inputs.	
0x0000F030	0x00000000-0xFFFFFFFF	R/W	UINT8[n]	**READ_Q - WRITE_Q** PLC process diagram of the physical outputs(%Q field). Offset is byte offset.	

Index Group	Index Offset	Access	Data type	Description	Remarks
0x0000F031	0x00000000- 0xFFFFFFFF	R/W	UINT8	**READ_QX - WRITE_QX** PLC process diagram of the physical outputs(%QX field). The index offset contains the bit address which is calculated from the byte number *8 + bit number.	
0x0000F035	0x00000000	R	ULONG	**ADSIGRP_IOIMAGE_ROSIZE** Byte length of the PLC process diagram of the physical outputs.	
0x0000F080	0x00000000- 0xFFFFFFFF = n (number of internal sub-commands)	R&W	W: n * ULONG[3] := IG1, IO1, Len1, IG2, IO2, Len2, ..., IG(n), IO(n), Len(n) R: n * ULONG + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] := Result1, Result2, ..., Result(n), Data1, Data2, ..., Data(n)	**ADSIGRP_SUMUP_READ** The write-data contains a list of multiple, separate AdsReadReq(IG, IO, Len, Data) sub-commands. The read-data contains a list of return codes followed by the requested data.	PLC / IO implemented in TwinCAT v2.10 Build >= 1324

Index Group	Index Offset	Access	Data type	Description	Remarks
0x0000F081	0x00000000- 0xFFFFFFFF = n (number of internal sub-commands)	R&W	W: (n * ULONG[3]) + UINT8[Len1] + UINT8[Len2] + ..., + UINT8[Len(n)] : = **IG1, IO1, Len1, IG2, IO2, Len2, ..., IG(n), IO(n), Len(n), Data1, Data2, ..., Data(n)** R: n * ULONG := **Result1, Result2, ..., Result(n)**	**ADSIGRP_SUMUP_WRITE** The write-data contains a list of multiple, separate AdsWriteReq(IG, IO, Len, Data) sub-commands. The read-data contains a list of return codes.	PLC / IO implemented in TwinCAT v2.11 Build >= 1550

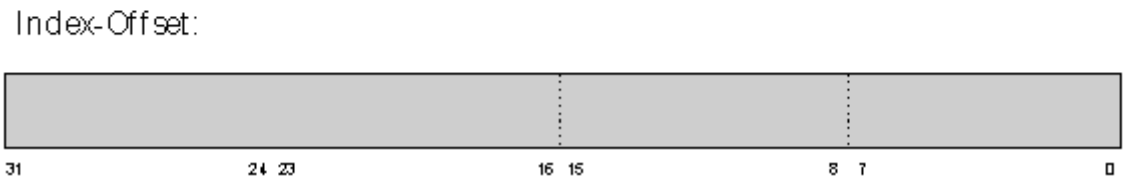
Index Group	Index Offset	Access	Data type	Description	Remarks
0x0000F082	0x00000000- 0xFFFFFFFF = n (number of internal sub-commands)	R&W	W: (n * ULONG[4]) + UINT8[WriteLen1] + UINT8[WriteLen2] + ..., + UINT8[WriteLen(n)] := **IG1, IO1, ReadLen1, WriteLen1, IG2, IO2, ReadLen2, WriteLen2, ..., IG(n), IO(n), ReadLen(n), ..., WriteLen(n), WriteData1, WriteData2, ..., WriteData(n)** R: (n * ULONG[2]) + UINT8[ReturnLen1] + UINT8[ReturnLen2] + ..., + UINT8[ReturnLen(n)] := **Result1, ReturnLen1, Result2, ReturnLen2, ..., Result(n), ReturnLen(n), Read**Data1, ReadData2, ..., ReadData(n)	**ADSIGRP_SUMUP_READWRITE** The write-data contains a list of multiple, separate AdsReadWriteReq(IG, IO, readLen, writeLen, Data) sub-commands. The read-data contains a list of return codes and return data length followed by the requested data.	PLC / IO implemented in TwinCAT v2.11 Build >= 1550

5.4 ADS Interface NC

5.4.1 Specification "Index group" for NC (ID [0x01...0xFF])

Index group (Hex)	Description		Remarks
0x1000	Ring-0-Manager: Parameter [▶ 37]		Optional !
0x1100	Ring-0-Manager: State [▶ 38]		Optional !
0x1200	Ring-0-Manager: Functions [▶ 39]		Optional !
0x1300	Ring-0-Manager: Cyclic process data		Not implemented !
0x2000 + ID	Channel with corr. ID: Parameter [▶ 39]		
0x2100 + ID	Channel with corr. ID: State [▶ 42]		
0x2200 + ID	Channel with corr. ID: Functions [▶ 43]		
0x2300 + ID	Channel with corr. ID: Cyclic process data [▶ 45]		
0x3000 + ID	Group with corr. ID: Parameter [▶ 46]		Optional!
0x3100 + ID	Group with corr. ID: State [▶ 50]		Optional!
0x3200 + ID	Group with corr. ID: Functions [▶ 55]		Optional!
0x3300 + ID	Group with corr. ID: Cyclic process data		Not implemented!
0x4000 + ID	Axis with corr. ID: Parameter [▶ 60]		
0x4100 + ID	Axis with corr. ID: State [▶ 74]		
0x4200 + ID	Axis with corr. ID: Functions [▶ 82]		
0x4300 + ID	Axis with corr. ID: Cyclic process data [▶ 98]		
0x5000 + ID	Encoder with corr. ID: Parameter [▶ 102]		Optional!
0x5100 + ID	Encoder with corr. ID: State [▶ 106]		Optional!
0x5200 + ID	Encoder with corr. ID: Functions [▶ 110]		Optional!
0x5300 + ID	Encoder with corr. ID: Cyclic process data [▶ 112]		Optional!
0x6000 + ID	Controller with corr. ID: Parameter [▶ 113]		Optional!
0x6100 + ID	Controller with corr. ID: State [▶ 117]		Optional!
0x6200 + ID	Controller with corr. ID: Functions [▶ 120]		Optional!
0x6300 + ID	Controller with corr. ID: Cyclic process data		Not implemented!
0x7000 + ID	Drive with corr. ID: Parameter [▶ 120]		Optional!
0x7100 + ID	Drive with corr. ID: State [▶ 123]		Optional!
0x7200 + ID	Drive with corr. ID: Functions [▶ 124]		Optional!
0x7300 + ID	Drive with corr. ID: Cyclic process data [▶ 125]		Optional!
0xA000 + ID	table (n x m) with corr. ID: Parameter [▶ 126]		
0xA100 + ID	table (n x m) with corr. ID: State [▶ 130]		
0xA200 + ID	table (n x m) with corr. ID: Functions [▶ 130]		
0xA300 + ID	table (n x m) with corr. ID: Cyclic process data		Not implemented!
0xF000 ... 0xFFFF	reserved area (TwinCAT system area)		
IndexGroup:	IndexOffset:		
0xF081	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_WRITE The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Write-commands* (like a group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1), WriteLen(1), ..., IdxGrp(n), Idx-Off(n), WriteLen(n),*	

Index group (Hex)	Description		Remarks
		WriteData(1), ..., WriteData(n)] Structure of the Read-Data: *[* Error*(1), ..., Error(n)]*	
0xF082	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_READWRITE The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Read-Write-commands* (like a group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1),ReadLen(1), WriteLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n), WriteLen(n), WriteData(1), ..., WriteData(n)]* Structure of the Read-Data: *[Error(1), ReadLen(1), ..., Error(n), ReadLen(n), ReadData(1), ..., ReadData(n)]*	
0xF084	0x00000000 ... 0xFFFFFFFF (n elements)	ADSIGRP_SUMUP_READ (READEX2) The *Read-Write-command* contains a list in the Write-data of multiple separate *ADS-Read-commands* (like a group request). Structure of the Write-Data: *[IdxGrp(1), IdxOff(1), ReadLen(1), ..., IdxGrp(n), IdxGrp(n), ReadLen(n)]* Structure of the Read-Data: *[Error(1), ReadLen(1), ..., Error(n), ReadLen(n), ReadData(1), ..., ReadData(n)]*	



5.4.1.1 Specification Ring-0-Manager

5.4.1.1.1 "Index offset" specification for Ring-0 parameter (Index group 0x1000)

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Remarks
0x00000010	Read	every	UINT32	100 ns		Cycle time SAF task	
0x00000012	Read	every	UINT32	100 ns		Cycle time SVB task	
0x00000014	Read	every	INT32	ns		Global Time Compensation Shift (for SAF task)	From TC 2.11R3 Build 2234
0x00000020	Read / Write	every	UINT16	1	0/1	Cyclic data consistence check and correction of the NC set-point values	From TC 2.11 B1550

5.4.1.1.2 "Index offset" specification for Ring-0 state (Index group 0x1100)

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT32	1	0, 1...255	Number of channels	
0x00000002	Read	every	UINT32	1	0, 1...255	Number of groups	
0x00000003	Read	every	UINT32	1	0, 1...255	Number of axes	
0x00000004	Read	every	UINT32	1	0, 1...255	Number of encoders	
0x00000005	Read	every	UINT32	1	0, 1...255	Number of controllers	
0x00000006	Read	every	UINT32	1	0, 1...255	Number of drives	
0x0000000A	Read	every	UINT32	1	0, 1...255	Number of tables (n x m)	
0x00000010	Read	every	UINT32	1		Cycle time error counter SAF task (not scopeable)	Reserved!
0x00000014	Read	every	UINT32	1		IO cycle time error counter SAF task (not scopeable)	Reserved!
0x00000020	Read	every	UINT32	µs		Computing time SAF task (not scopeable)	Reserved!
0x00000031	Read	every	UINT32[number]	1	0, 1...255	Supplies the channel IDs for all channels in the system	
0x00000032	Read	every	UINT32[number]	1	0, 1...255	Supplies the group IDs for all groups in the system	
0x00000033	Read	every	UINT32[number]	1	0, 1...255	Supplies the axis IDs for all axes in the system	
0x00000034	Read	every	UINT32[number]	1	0, 1...255	Supplies the encoder IDs for all encoders in the system	
0x00000035	Read	every	UINT32[number]	1	0, 1...255	Supplies the controller IDs for all controllers in the system	
0x00000036	Read	every	UINT32[number]	1	0, 1...255	Supplies the Drive IDs for all Drives in the system	
0x0000003A	Read	every	UINT32[number]	1	0, 1...255	Supplies the table IDs for all tables in the system	
0x000001nn	Read	every	UINT32	1	0, 1...255	Supplies for the encoder ID the appropriate axis ID nn = Encoder ID	Reserved!

Index offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Remarks
0x000002nn	Read	every	UINT32	1	0, 1...255	Supplies for the Controller ID the appropriate axis ID nn = Controller ID	Reserved!
0x000003nn	Read	every	UINT32	1	0, 1...255	Supplies for the Drive ID the appropriate axis nn = Drive ID	Reserved!

5.4.1.1.3 "Index offset" specification for Ring-0 functions (Index group 0x1200)

Index-Offset (Hex)	Access	Ring-0-Manager	Data type	Phys. unit	Definition range	Description	Remarks
0x00000020	Write	every	VOID	1		Clear cycle time error counter SAF & SVB	Reserved!

5.4.1.2 Specification Channels

5.4.1.2.1 "Index offset" specification for channel parameter (Index group 0x2000 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT32	1		Channel ID	
0x00000002	Read	every	UINT8[30+1]	1		Channel name	
0x00000003	Read	every	UINT32	1	ENUM	Channel type [▶ 131]	
0x00000004	Read	every	UINT32	1	ENUM	Interpreter type [▶ 132]	
0x00000005	Read	every	UINT32	1		Program load buffer size in bytes	
0x00000006	Read	every	UINT32	1		Program no. according to job list	
0x00000007	Read/Write	every	UINT32	1	ENUM	Set load log mode [▶ 132]	
0x00000008	Read/Write	every	UINT32	1	ENUM	Set trace mode [▶ 132]	
0x00000009	Read/Write	every	UINT32	1		RESERVED	
0x0000000A	Read/Write	every	UINT32	1	0/1	Records all feeder entries in a log file named "TcNci.log"	
0x0000000B	Read/Write	every	UINT32	1	0/1	Channel specific level for NC logger messages 0: errors only 1: all NC messages	from TwinCAT V2.8 B747 V2.9 B948
0x00000010	ReadWrite	every	**Write**				
			{				

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
			UINT32	1	0..159	Start index of M-function	
			UINT32	1	1..160	Number of M-functions to be read	
			}				
			Read[number]				
			{				
			UINT8	1	0..159	Rule bit mask of the M-function	
			INT32[10]	1	-1..159	Number of M-functions to be cleared	
			}				
0x00000011	Write	Interpolation				Write M-function description	From TC V2.9 B803 internal use only!
0x00000012	Read/Write	Interpolation	LREAL64	1		Factor for G70	From TC V2.9 Build 803
0x00000013	Read/Write	Interpolation	LREAL64	1		Factor for G71	From TC V2.9 Build 803
0x00000014	Write	Interpolation	{			Axes user symbols	Not yet released
			char[32]			User symbol (null-terminated)	
			char[10]			System symbol (null-terminated)	
			}				
0x00000015	Read/Write	Interpolation	UINT16 resp. UINT32	1	0/1 Default: FALSE	Activation of default G-code	From TC 2.11R3 B2241
0x00000021	Read	every	UINT32	1		Group ID (only explicit for 3D and FIFO channel)	
0x00000031	Read/Write	Interpolation	UINT16	1		Standard Output Port of the interpreter	Reserved function, no standard!
0x00000032	Read/Write	Interpolation	UINT16	1	0/1	Cartesian tool offset entry	Reserved function, no standard!
0x00000040	Read/Write	Interpolation	{			Target address of interpreter hooks	Reserved function, no standard!
			char[6]			Ams Net ID	
			UINT16			Port	
			UINT32			Index group	
			UINT32			Index offset	
			}				
0x00000050	Read/Write	Interpolation	UINT32	1	ENUM	Reaction if at the radius compensation a bottle neck recognized 0: Error and abort 1: Note & trouble shooting	From TwinCAT Version 2.8

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
						2: Only note, without outline modulation	
0x00000051	Read/Write	Interpolation	UINT32	1	1..24	Look ahead for bottleneck detection	From TwinCAT Version 2.8
0x00000052	Read/Write	Interpolation	UINT32	1	0/1	Chamfer on/off	Reserved function, no standard!
0x00000053	Read/Write	Interpolation	UINT32	1		Activation for reading the currently effective interpolation rules, zero shifts and rotation 0: off 1: on	From TC V2.10 B1303
0x00000054	Read/Write	Interpolation	UINT32	1	0/1	Retrace on/off	Reserved function, no standard!
0x00000055	Read/Write	Interpolation	UINT32[4]	1		Configuration of the cyclic channel interface for UINT 32 Up to 4 index offsets can be configured	From TC V2.10 B1320
0x00000056	Read/Write	Interpolation	UINT32[4]	1		Configuration of the cyclic channel interface for LREAL Up to 4 index offsets can be configured	From TC V2.10 B1320
0x00010K0L	Read/Write	every	REAL64	e.g. mm	±MAX REAL64	Value for zero shift (NPV)	
					[1..3]	Axis index: K=1 → X K=2 → Y K=3 → Z	
					[1..0xA]	L=1 → G54F L=2 → G54G L=3 → G55F ...	
0x0002ww00	Read/Write	every	UINT16			Tool number: values for tool compensation	
0x0003ww00	Read/Write	every	UINT16		[1...50]	Tool type: ww = tool 1...50	
0x0004wwnn	Read/Write	every	REAL64		[1...14]	Parameter: nn = Index 1...14	
0x000500gg	Read/Write	every	REAL64	e.g. mm	≥ 0 (value) [1...9] (g)	Radius of the tolerance sphere gg = group of the Channel (default: 1)	

5.4.1.2.2 "Index offset" specification for channel state (Index group 0x2100 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	INT32	1	ENUM	Error code Channel	
0x00000002	Read	every	UINT32	1		Number of groups in the Channel	
0x00000003	Read	every	UINT32	1	ENUM	Interpreter state [▶ 132]	Cannot be traced by oscilloscope!
0x00000004	Read	every	UINT32	1		Interpreter/ channel operation mode	
0x00000005	Read	every	UINT32	1		Currently loaded program	
0x00000007	Read	every	UINT8[...]	1		Program name of currently loaded program (100 characters, null-terminated)	Max. 100 characters, null-terminated
0x00000008	Read	Interpreter	UINT32	1	[0,1]	Interpreter simulation mode 0: off (default) 1: on	From V2.9 B946 Cannot be traced by oscilloscope!
0x00000010	Read	Interpreter	UINT32	1		Text index If the interpreter is in the aborted state, the current text index can be read out here	Cannot be traced by oscilloscope!
0x00000011	ReadWrite	Interpreter	**Write**				Cannot be traced by oscilloscope!
			UINT32	1		Text index	
			Read				
			UINT8[..]	1		Line of the NC part program from the text index	
0x00000012	Read	Interpreter	{				From TC V2.10 B1243
			UINT32	1		Current display for 1: SAF 2: Interpreter 3: Error offset	
			UINT32	1		File offset	
			UINT8[260]	1		Path + program name	
			}				
0x00000013	Read	Interpreter	UINT32[18]			Display for currently effective G-code	From TC V2.10 B1303
0x00000014	Read	Interpreter	{			Determines the currently effective zero shift	From TC V2.10 B1303
			UINT32	1		Block counter	
			UINT32			Dummy	
			LREAL[3]	1		Zero shift G54..G57	
			LREAL[3]	1		Zero shift G58	
			LREAL[3]	1		Zero shift G59	
			}				

Index offset (Hex)	Access	Channel type	Data type	Phys. Unit	Definition range	Description	Note
0x00000015	Read	Interpreter	{			Determines the currently effective rotation	From TC V2.10 B1303
			UINT32	1		Block counter	
			UINT32	1		Dummy	
			LREAL[3]	1		Rotation of X, Y & Z in degrees	
			}				
0x00000016	Read	Interpreter	UINT32	1	[0,1]	Feeder Info	Internal usage, no standard!
0x00000100	Read	every	UINT32[number]	1	[0, 1...255]	Returns the respective axis ID in the channel	Number: [1...255] axis IDs: [0, 1...255] Cannot be traced by oscilloscope!

5.4.1.2.3 "Index offset" specification for channel functions (Index group 0x2200 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Write	every	UINT32	1		Load NC program by program number	
0x00000002	Write	every	VOID			Start Interpreter	
0x00000003	Write	every	VOID			RESERVED	
0x00000004	Write	every	UINT8[...]			Load NC program by name. The standard NC path does not also have to be given, although it may. Other paths are also permitted.	
0x00000005	Write	every	UINT16	ENUM	cf. appendix interpreter operating modes [▶ 132]	Set the interpreter/channel operation mode	From TwinCAT V2.9 Build 901
0x00000006	Write	Interpreter	UINT8[...]			Set path for subroutines	From TwinCAT V2.9, Build 1001
0x00000008	Write	Interpreter	UINT32	1		Interpreter simulation mode: 0: off (default) 1: on	Not yet released
0x0000000F	Write	every	VOID			RESERVED	
0x00000010	Write	every	VOID			"Reset" Channel	
0x00000011	Write	every	VOID			"Stop" Channel	
0x00000012	Write	every	VOID			"Retry" Channel (restart Channel)	Not implemented!
0x00000013	Write	every	VOID			"Skip" Channel (skip task/ block)	Not implemented!
0x00000014/ 0x00000015	Write	every	{			"Enable Retrace" /	Reserved function, no standard!

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
						"Disable Retrace"	
			UINT32	1	>0	Feeder processing direction: 1: forward 2: backward	
			UINT32	1	≥ 0	Entry index	
			REAL64[3]	mm	±∞	Pos. of the main axes X, Y, Z	
			REAL64[5]	mm	±∞	Pos. of the auxiliary axes Q1, ..., Q5	
			}				
0x00000018	Write	Interpreter	_ST_ItpBlockSearchParams			Enable Blocksearch	
0x00000019	Write	Interpreter	VOID			StepOnAfterBlockSearch	
0x00000020	Write	every	VOID			"Save" zero shift (NPV)	
0x00000021	Write	every	VOID			"Load" zero shift (NPV)	
0x00000022	Write	every	VOID			"Save" tool compensations	
0x00000023	Write	every	VOID			"Load" tool compensations	
0x00000024	Write	Interpolation	{			Saves a snapshot of the interpreter in a specified file	From TwinCAT V2.9, Build 1002
			char[32]			File name in the TwinCAT\CNC directory	
			UINT32	1	0..1	Mask: 0x1: R-parameters 0x2: zero shifts (from Build 1235) 0x4: Tool descriptions (from Build 1235)	
			}				
0x00000025	Write	Interpolation	{			Reads snapshot from a specified file into the interpreter	From TwinCAT V2.9, Build 1002
			char[32]			File name in the TwinCAT\CNC directory	
			UINT32	1	0..1	Mask: 0x1: R-parameters 0x2: zero shifts (from Build 1235) 0x4: tool description (from build 1235)	
			}				

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000026	Write	Interpolation	VOID			Set all tool parameters (incl. type and number) to null	From TwinCAT V2.9, Build 1031
0x00000027	Write	Interpolation	VOID			Set all zero shifts to null	From TwinCAT V2.9, Build 1031
0x00000030	Write	every	VOID			Restart (Go Ahead) of the Interpreter after programmed Interpreter stop	
0x00000040	Write	every	VOID			Trigger event for deletion of any remaining travel in the NCI	
0x00000041	Write	every				RESERVED for measuring event	
0x00000050	Write	Interpolation	VOID	1		Set ExecIdle-Info in the interpreter	Reserved function, no standard!
0x00000051	Write	Interpolation	UINT32	1		Set block skipping mask in the interpreter Parameter: SkippingMask	Reserved function, no standard!
0x00000052	Write	Interpolation	UINT32	1		Set ItpOperationMode in the interpreter Parameter: Mask of the operation mode	Reserved function, no standard!
0x00000053	Write	Interpolation	VOID			Set ScanningFlag in the NC device	Reserved function, no standard!
0x00000054	Write	Interpolation				*ScanPosition*	Reserved function, no standard!
			double[8]			Position	
0x00000055	Write	Interpolation				Reserved	
0x00000056	Write	Interpolation	VOID			Set interpreter in aborted status	Reserved function, no standard!
0x00000060	Write	Interpolation	UINT16	1	0..159	Manual reset of a fast M-function	

5.4.1.2.4 "Index offset" specification for cyclic channel process data (Index group 0x2300 + ID)

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000000	Read	every (PLC→NC)	{150 Byte}		STRUCT s. Channel interface	CHANNEL STRUCTURE (PLC→NC)	The associated PLC structure is: NciChannelFromPlc Old structure: PLCTONC_CHANNELSTRUCT

Index offset (Hex)	Access	Channel type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT8[...] min. 30 Byte	1		Interpreter program display	Cannot be traced by oscilloscope!
0x00000002	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Speed override Channel (axes in the Channel)	1000000 = 100%
0x00000003	Read/Write	every (PLC→NC)	UINT32	%	[0...1000000]	Speed override spindle	1000000 = 100%
0x00000080	Read	every (NC→-PLC)	{150 Byte}		STRUCT s. Channel interface	CHANNEL STRUCTURE (NC→PLC)	The current associated PLC structure is: NciChannelToPlc old structure: NCTOPLC_CHANNELSTRUCT
0x10000000 + Register index	Read/Write	every	REAL64	1	[0...999]	R-parameter of the interpreter	Cannot be traced by oscilloscope!
0x20000001	Read	every	UINT8[...] min. 30 Byte	1	[1...9]	Program display of the group processing (SAF)	Cannot be traced by oscilloscope!

5.4.1.3 Specification Groups

5.4.1.3.1 "Index offset" specification for group parameter (Index group 0x3000 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1		Group ID	
0x00000002	Read	every	UINT8[30+1]	1		Group name	
0x00000003	Read	every	UINT32	1		Group type	
0x00000004	Read	every	UINT32	µs		SAF cycle time group	
0x00000005	Read	every	UINT32	µs		SVB cycle time group	
0x00000006	Read/Write	every	UINT16	1	0/1	Single block operation mode?	
0x0000000B	Read	every	UINT32	1		Size of the SVB table (max. number of SVB entries	
0x0000000C	Read	every	UINT32	1		Size of the SAF table (max. number of SAF entries	
0x00000010	Read/Write	every	UINT32	1	[1,2...32]	Internal SAF cycle time divisor (divides the internal SAF cycle time by this factor)	Default: 1

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000021	Read	Channel: every	UINT32	1		Channel ID	
0x00000022	Read	Channel: every	UINT8[30+1]	1		Channel name	
0x00000023	Read	Channel: every	UINT32	1		Channel type	
0x00000024	Read	Channel: every	UINT32	1	>0	Number in the Channel	
0x00000500	Read/Write	DXD group	INT32	ENUM	[0, 1]	Curve velocity reduction method 0: Coulomb-Scattering 1: Cosinus law 2: VeloJump	
0x00000501	Read/Write	DXD group	REAL64	1	[0.0...1.0]	Velocity reduction factor C0 transition (continual course, but neither once nor twice differentiable)	
0x00000502	Read/Write	DXD group	REAL64	1	[0.0...1.0]	Velocity reduction factor C1 transition (continual course and once differentiable)	
0x00000503	Read/Write	DXD group	REAL64	degree	[0.0...180.0]	Critical angle on the segment transition "Low" (must genuinely be smaller than or equal to the velocity reduction angle C0)	
0x00000504	Read/Write	DXD group	REAL64	degree	[0.0...180.0]	Critical angle on the segment transition "High" (must genuinely be smaller than or equal to the velocity reduction angle C0)	
0x00000505	Read/Write	DXD group	REAL64	mm/s	≥ 0	Minimum velocity that must be maintained at segment transitions in spite of any possible velocity reduction.	
0x00000506	Read/Write	DXD group	REAL64	e.g. mm	[0.0...1000.0]	Radius of the tolerance sphere for smoothing	Not implemented!
0x00000507	Read/Write	DXD group	REAL64	1		Velocity reduction factor C2 transition	
0x00000508	Read/Write	DXD group	UINT32	1	0/1	Enables calculation of the total remaining path length	
0x00000509	Read/Write	DXD group	UINT16	1	0/1	General activation of the software limit position monitoring	From TwinCAT V2.9 B959

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						for the main axes (X, Y, Z) (see encoder parameters)	
0x0000050A	Read/Write	DXD group	UINT32	1	0/1	NCI Override type 0: related to internal reduced velocity (without iteration) 1: related to original external (programmed) velocity	From TwinCAT V2.9 B948
0x0000050B	Read/Write	DXD group	UINT16	1	0/1	NCI override 0: Override limited to 100% 1: Override >100% possible	From TwinCAT V2.10 B1226
0x00000510	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method Velo-Jump Reduction factor for C0 transitions: X axis	Not implemented!
0x00000511	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method Velo-Jump Reduction factor for C0 transitions: Y axis	Not implemented!
0x00000512	Read/Write	DXD group	REAL64	1	≥ 0	For reduction method Velo-Jump Reduction factor for C0 transitions: Z axis	Not implemented!
0x00000513	Read/Write	DXD group	LREAL64	1	]0.0..1.0[	Blending for auxiliary axes: If the effective path velo is smaller than the programmed one multiplied with this factor, then an accurate stop is inserted and the tolerance ball is deleted	From TwinCAT V2.11 B1552
0x00000604	Read/Write	Encoder group	REAL64	e.g. mm/s	[0.0...1000.0]	Velocity window resp. standstill window	Base Unit / s
0x00000605	Read/Write	Encoder group	REAL64	s	[0.0...60.0]	Filter time for standstill window in seconds	
0x00000606	Read/Write	Encoder group	REAL64	s	[0.0...60.0]	Dead time compensation master/slave coupling ("angle pre-control")	
0x00000701	Read	FIFO group	UINT32	1	[1...8] resp. [1...16]	FIFO dimension (m = number of axes)	(n x m)-FIFO Boot data!

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						(from TC 2.11 Build 1547 the FIFO dimension has been increased from 8 to 16)	
0x00000702	Read	FIFO group	UINT32	1	[1...10000]	FIFO size (length) (n = number of FIFO entries)	(n x m)-FIFO Boot data!
0x00000703	Read	FIFO group	UINT32	1	[0, 1, 4]	Interpolation type for FIFO setpoint generator 0: INTERPOLATION-TYPE_LINEAR (Default) 1: INTERPOLATION-TYPE_4POINT 4: INTERPOLATION-TYPE_CUBIC-SPLINE (with 6 points)	From TwinCAT 211R3 Build 2257
0x00000704	Read/Write	FIFO group	UINT32	1	[1, 2]	Override type for FIFO set-point generator type 1: OVERRIDETYPE_INSTANTANEOUS (default) type 2: OVERRIDE-TYPE_PT2	
0x00000705	Read/Write	FIFO group	REAL64	s	> 0.0	P-T2-time for override change (T1=T2=T0)	
0x00000706	Read/Write	FIFO group	REAL64	s	≥ 0.0	Time delta for two sequenced FIFO entries (FIFO entry timebase)	
0x00000801	Read/Write	Kinematic group	**Write**			Calculation of the kinematic forward transformation for the positions (ACS -> MCS)	
			{				
			REAL64 [8]	e.g. degree	±∞	Positions of the ACS axes (Axis Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
			Read				
			{				

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
			REAL64 [8]	e.g. mm	±∞	Positions of the MCS axes (Machine Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
0x00000802	ReadWrite	Kinematic group	**Write**			Calculation of the kinematic backward transformation for the positions (MCS -> ACS)	
			{				
			REAL64 [8]	e.g. mm	±∞	Positions of the MCS axes (Machine Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				
			Read				
			{				
			REAL64 [8]	e.g. degree	±∞	Positions of the ACS axes (Axis Coordinate System), max. dimension: 8	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve	
			}				

5.4.1.3.2 "Index offset" specification for group state (Index group 0x3100 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	INT32	1	ENUM	Error code group	
0x00000002	Read	every	UINT32	1		Number of master axes	
0x00000003	Read	every	UINT32	1		Number of slave axes	
0x00000004	Read	every	UINT32	1	s. ENUM	SVB group state (state)	
0x00000005	Read	every	UINT32	1	s. ENUM	SAF group state (main state)	
0x00000006	Read	every	UINT32	1	s. ENUM	moving state (state)	
0x00000007	Read	every	UINT32	1	s. ENUM	SAF sub-group state (sub state)	
0x00000008	Read	every	UINT32	1	s. ENUM	Referencing state (state)	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000009	Read	every	UINT32	1	s. ENUM	Coupling state (state)	Cannot be traced by oscilloscope!
0x0000000A	Read	every	UINT32	1	≥0	Coupling table index	Cannot be traced by oscilloscope!
0x0000000B	Read	every	UINT32	1	≥0	Current number of SVB entries/ tasks	*Symbolic access:* *'SvbEntries' (DXD)*
0x0000000C	Read	every	UINT32	1	≥0	Current number of SAF entries/ tasks	*Symbolic access:* *'SafEntries' (DXD)*
0x0000000D	Read	every	UINT32	1		Current block number (active only for interpolation group)	*Symbolic access:* *'BlockNumber' (DXD)*
0x0000000E	Read	every	UINT32	1	≥0	Current number of free SVB entries/tasks	From TwinCAT V2.9 B903 Cannot be traced by oscilloscope!
0x0000000F	Read	every	UINT32	1	≥0	Current number of free SAF entries/tasks	From TwinCAT V2.9 B903 Cannot be traced by oscilloscope!
0x00000011	Read	every	UINT16	1	0/1	Emergency Stop (E-Stop) active?	Cannot be traced by oscilloscope!
0x00000110	Read	PTP group	{			Internal NC information (resolutions)	Reserved!
			REAL64	e.g. mm	± ∞	ExternalEndPosition	
			REAL64	e.g. mm/s	>0	ExternalTargetVelocity	
			REAL64	e.g. mm/s^2	>0	ExternalAcceleration	
			REAL64	e.g. mm/s^2	>0	ExternalDeceleration	
			REAL64	e.g. mm/s^3	>0	ExternalJerk	
			UINT32	1	>0	ExternalOverrideType	
			REAL64	e.g. mm	± ∞	InternalEndPosition	
			REAL64	e.g. mm/s	>0	InternalTargetVelocity (refers to 100 %)	
			REAL64	%	[0 ... 100]	InternalActualOverride	
			REAL64	e.g. mm/s^2	>0	InternalAcceleration	
			REAL64	e.g. mm/s^2	>0	InternalDeceleration	
			REAL64	e.g. mm/s^3	>0	InternalJerk	
			REAL64	e.g. mm	>0	PositionResolution	
			REAL64	e.g. mm/s	≥0	VelocityResolution	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
			REAL64	e.g. mm/s^2	≥0	AccelerationResolution	
			REAL64	e.g. mm/s	≥0	VelocityResolutionAtAccelerationZero	
			}				
0x00000500	Read	DXD group	REAL64	e.g. mm	≥ 0	Path rest way (remaining arc length) on the current path segment	*Symbolic access:* *'SetPathRemLength'*
0x00000501	Read	DXD group	REAL64	e.g. mm	≥ 0	Racked out arc length on the current path segment	*Symbolic access:* *'SetPathLength'*
0x00000502	Read	DXD group	REAL64	e.g. mm/s	≥ 0	Current path set velocity	*Symbolic access:* *'SetPathVelo'*
0x00000503	Read	DXD group	REAL64	e.g. mm/s^2	± ∞	Current path set acceleration	*Symbolic access:* *'SetPathAcc'*
0x00000504	Read	DXD group	REAL64	e.g. mm/s^2	≥ 0	Amount of the current vectorial set acceleration	*Symbolic access:* *'SetPathAbsAcc'*
0x00000505	Read	DXD group	REAL64	e.g. mm/s	≥ 0	Maximum segment end path set velocity	*Symbolic access:* *'SetPathVeloEnd'*
0x00000506	Read	DXD group	REAL64	e.g. mm/s	≥ 0	Segment maximum path set velocity	*Symbolic access:* *'SetPathVeloMax'*
0x00000507	Read	DXD group	REAL64	e.g. mm	≥ 0	current relative braking distance based on the current arc length	*Symbolic access:* *'SetPathStopDist'*
0x00000508	Read	DXD group	REAL64	e.g. mm	± ∞	Safety distance = segment arc length - current arc length - relative braking distance	*Symbolic access:* *'SetPathSecurityDist'*
0x00000509	Read	DXD group	REAL64	1	0/1	Segment transition	*Symbolic access:* *'SetPathSegmentChange'*
0x0000050A	Read	DXD group	REAL64	%	[0 ... 100]	Path velocity override	*Symbolic access:* *'SetPathOverride'*
0x00000511	Read	DXD group	REAL64	e.g. mm/s	≥ 0	Amount of the path actual velocity	*Symbolic access:* *'ActPathAbsVelo'*
0x00000512	Read	DXD group	REAL64	e.g. mm/s^2	± ∞	Path actual acceleration on the current segment	*Symbolic access:* *'ActPathAcc'*

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000513	Read	DXD group	REAL64	e.g. mm/s^2	≥ 0	Amount of the path actual acceleration on the current segment	*Symbolic access:* *'ActPathAbsAcc'*
0x00000514	Read	DXD group	REAL64	e.g. mm	± ∞	Position error on the path in a tangential direction (signed for lead and lag)	*Symbolic access:* *'PathDiffTangential'*
0x00000515	Read	DXD group	REAL64	e.g. mm	≥ 0	Position error on the path in orthogonal direction	Symbolic access: *'PathDiffOrthogonal'*
0x00000520	Read	DXD group	REAL64	1	≥ 0	Covered arc length of the current segment, normalized to 1.0	
0x00000521	Read	DXD group	REAL64	1	0/1	Change of partial segment (radius of tolerance ball)	
0x00000522	Read	DXD group	REAL64	1	≥ 0	Total remaining path length to the last geometry entry or the next accurate stop. Refers to group parameter 0x508.	
0x00000523	Read	DXD group	REAL64	1	≥ 0	Programmed velocity of the current segment	From TC V2.9 B1031
0x00000530	Read	DXD group	{			Current or last target position of the main axes X, Y and Z	
			REAL64	e.g. mm	± ∞	Target position X-axis	
			REAL64	e.g. mm	± ∞	Target position Y-axis	
			REAL64	e.g. mm	± ∞	Target position Z-axis	
			}				
0x00000531	Read	DXD group	{			Current or last target position of the auxiliary axes Q1 to Q5	
			REAL64 [5]	e.g. mm	± ∞	Target position of axis Q1 to Q5	
			}				
0x00000532	Read	DXD group	{			Reads path length, H parameter and Entry ID of the next 11 segments in relation to the current DC time	From TC 2.11 B2226
			UINT32			DC Time	
			UINT32			Reserved	
			PreViewTab[11]			11*24 Bytes	
			}				

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
			PreViewTab				
			{				
			REAL64	e.g. mm		Segment length	
			UINT32	1		block number	
			UINT32	1		H-Parmeter	
			UINT32	1		Entry ID	
			UINT32	1		Reserved	
			}				
0x0000054n	Read	DXD group	REAL64	1	0/1	Within the tolerance ball of the auxiliary axis n = 1..5 number of the auxiliary axis (not axis ID)	From TC V2.9 B932
0x00000550	Read	DXD group	{			Reading axis IDs inside the 3D-group:	
			UINT32	1	[0, 1...255]	X axis ID	
			UINT32	1	[0, 1...255]	Y axis ID	
			UINT32	1	[0, 1...255]	Z axis ID	
			}				

0x00000552	Read	DXD group
		FIFO group
		Kinematic group
			{
			UINT32[m]
			}
				1	[0, 1...255]	Axis allocation of the group:
						1. axis ID. ..., m. axis ID
						m: dimension of the 3D group with main and auxiliary axes (X, Y, Z, Q1, Q2, Q3, Q4, Q5) or the FIFO group or the ACS axes of the kinematic group	

0x00000553	Read	Kinematic group	{			Reading the axis allocation (IDs) inside the kinematic group:	
			UINT32[8]	1	[0, 1...255]	MCS axis IDs (Machine Coordinate System)	
			UINT32[8]	1	[0, 1...255]	ACS-axis ID's (Axis Coordinate System)	
			UINT32	1	≥ 0	Reserve	
			UINT32	1	≥ 0	Reserve (NEW)	
			}				
0x00000556	Read	DXD group	ST_ItpBlock-SearchData			Reading the block search data	
0x0000056n	Read	DXD group	REAL64	1	± ∞	Current position error of the auxiliary axis within the tolerance ball (set value side only) only for auxiliary axes	From TC V2.9 B932

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						n = 1..5 number of the auxiliary axis (not axis ID)	

5.4.1.3.3 "Index offset" specification for group functions (Index group 0x3200 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Write	every	VOID			Reset group	
0x00000002	Write	every	VOID			Stop group	
0x00000003	Write	every	VOID			Clear group (buffer/task)	
0x00000004	Write	PTP group, 3D group	{			Emergency Stop (E-Stop) (Emergency stop with regulated ramp)	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration (must be greater than or equal to the original deceleration)	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk (must be greater than or equal to the original jerk)	
			}				
0x00000005	Write	PTP group	{			Parametrized stop (with regulated ramp)	Reserved function, no standard!
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000006	Write	PTP group, 3D group	VOID			"Step on" after Emergency Stop (E-Stop)	
0x00000050	Write	PTP group 3D group	{			Axis allocation of the group:	
			UINT32	1	[0, 1...255]	X axis ID	
			UINT32	1	[0, 1...255]	Y axis ID	
			UINT32	1	[0, 1...255]	Z axis ID	
			}				
0x00000051	Write	PTP group 3D group FIFO group	{			Axis allocation of the group:	
			UINT32	1	[1...255]	Axis ID	
			UINT32	1	[0 ... (m-1)]	Place index of the axis in the group m: group dimension (PTP: 1;DXD: 3, FIFO: 8 resp. 16) (from TC 2.11 Build 1547 the FIFO dimension	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						has been increased from 8 to 16)	
			}				

0x00000052	Write	3D group
		FIFO group
			{
			UINT32[m]
			}
				1	[0, 1...255]	Axis allocation of the group:
						1. axis ID. ..., m. axis ID
						m: dimension of the 3D group (X, Y, Z, Q1, Q2, Q3, Q4, Q5) resp. FIFO group	

0x00000053	Write	3D grou FIFO group Kinematic group	VOID			Delete the 3D axis allocation, FIFO axis allocation or Kinematic axis allocation and return of the axes to their own PTP groups	
0x00000054	Write	Kinematic group	{			Axis allocation of the kinematic group:	
			UINT32[8]	1	[0, 1...255]	MCS axis IDs (Machine Coordinate System)	
			UINT32[8]	1	[0, 1...255]	ACS-axis IDs (Axis Coordinate System)	
			UINT32	1	≥ 0	Reserved	
			UINT32	1	≥ 0	Reserved (NEW)	
			}				
0x00000060	ReadWrite	3D group		1		Internal "feed group" command ("Feeder")	Internal command!
0x00000061	ReadWrite	3D group		1		Internal "feed group" command ("Feeder")	Internal command!
0x00000110	Write	1D group	VOID			Reference 1D group ("calibration")	
0x00000111	Write	1D group	{			New end position 1D group	
			UINT32	ENUM	s. appendix	End position type (s. appendix)	
			REAL64	e.g. mm	±∞	New end position (target position)	
			}				
0x0000011A	Write	1D group	{			Set actual position 1D group	**Caution by using!** Always to SAF Port 501!
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	Actual position for axis	
			}				

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x0000011B	Write	1D group	UINT32	1	0/1	Set reference flag ("calibrate flag")	**Caution by using!**
0x00000120	Write	1D group	{			Start 1D group (standard start)	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥ 0.0	Required velocity	
			}				
0x00000121	Write	1D group (SERVO)	{			Start 1D group (extended start):	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g.mm	±∞	End position (target position)	
			REAL64	mm/s	≥0.0	Required velocity	
			UINT32	1	0/1	Standard acceleration?	
			REAL64	mm/s^2	≥ 0.0	Acceleration	
			UINT32	1	0/1	Standard deceleration?	
			REAL64	mm/s^2	≥ 0.0	Deceleration	
			UINT32	1	0/1	Standard jerk?	
			REAL64	mm/s^3	≥ 0.0	Jerk	
			}				
0x00000122	Write	1D group (MW-SERVO)	{			Start 1D group (special start):	Reserved start function, no standard!
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥0.0	Required start velocity	
			REAL64	e.g. mm	±∞	Position for a new velocity level	
			REAL64	mm/s	≥0.0	New end velocity level	
			UINT32	1	0/1	Standard acceleration?	
			REAL64	mm/s^2	≥0.0	Acceleration	
			UINT32	1	0/1	Standard deceleration?	
			REAL64	mm/s^2	≥0.0	Deceleration	
			UINT32	1	0/1	Standard Jerk?	
			REAL64	mm/s^3	≥0.0	jerk	
			}				
0x00000126	Write	1D group	{			Start drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000127	Write	1D group	VOID			Start drive output	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000128	Write	1D group	{			Change the drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000130	Write	1D group (SERVO)	{			1D section compensation (SERVO):	
			UINT32	ENUM	s. appendix	Compensation type (s. appendix)	
			REAL64	mm/s/s	≥ 0.0	Max. acceleration increase	
			REAL64	mm/s/s	≥ 0.0	Max. deceleration increase	
			REAL64	mm/s	≥ 0.0	Max. increase velocity	
			REAL64	mm/s	≥ 0.0	Base velocity for the process	
			REAL64	e.g. mm	±∞	Path difference to be compensated	
			REAL64	e.g. mm	≥ 0.0	Path distance for compensation	
			}				
0x00000131	Write	1D group SERVO	VOID			Stop section compensation (SERVO)	
0x00000140 (0x00n00140)	Write	Master/Slave coupling: 1D group (SERVO)	{			Master/slave coupling (SERVO):	Extension for "flying saw"! Angle > 0.0 and £ 90.0 degree (parallel saw: 90.0 degree)
			UINT32	ENUM	s. appendix	Slave type/coupling type (s. appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis/ group	
			UINT32	1	[0...8]	Subindex n of the master axis (default: -value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: -value: 0)	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: gearing factor FlySawVelo: Reserve FlySaw: abs. synchron position master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve FlySawVelo: Reserve	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
						FlySawPos: abs. synchron position slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve FlySawVelo: angle of inclination in [DEGREE] FlySawPos: angle of inclination in [DEGREE]	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve FlySawVelo: Gearing factor FlySawPos: Gearing factor	
			}				
0x00000141	Write	Master/Slave decoupling: 1D group (SERVO)	VOID			Master/slave decoupling (SERVO)	
0x00000142	Write	Master/Slave parameter 1D group (SERVO)	{			Change of the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gearing factor	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000144	Write	Slave stop 1D group (SERVO)	VOID			Stop the "flying saw" (SERVO)	Only for "flying saw"
0x00000149	Write	Slave tables 1D group (SERVO)	REAL64	1	±∞	Set the slave table scaling of a solo table coupling (SERVO)	Only for Solo table slave
0x00000150	Write	1D group	VOID			Deactivate complete 1D group/axis (disable)	
0x00000151	Write	1D group	VOID			Activate complete 1D group / axis (enable)	
0x00000160	Write	1D group	VOID			Deactivate drive output of the 1D group (disable)	

Index offset (Hex)	Access	Group type	Data type	Phys. Unit	Definition range	Description	Note
0x00000161	Write	1D group	VOID			Activate drive output of the 1D group (enable)	
0x00000362	Write	High/low speed group	UINT16	1	0/1	Release parking brake? 0: automatic activation (default 1: mandatorily always released!	
0x00000701	Write	FIFO group	VOID			Start FIFO group (FIFO table must first have been filled)	(n*m)-FIFO

0x00000710	Write	FIFO group	{
			REAL64[x*m]
			}
				e.g. mm	±∞	Write x FIFO entries (lines):
						(x*m)-values (one or more lines)
						n: FIFO length (number of lines)
						m: FIFO dimension (number of columns)
						value range x: [1 ... n]
							Only possible on a line-by-line basis!
							(multiple integer)

0x00000711	Write	FIFO group	{
			REAL64[x*m]
			}
				e.g. mm	±∞	Overwrite the last x FIFO entries (lines):
						(x*m)-values (one or more lines)
						n: FIFO length (number of lines)
						m: FIFO dimension (number of columns)
						value range x: [1 ... n]
							Only possible on a line-by-line basis!
							(multiple integer)

0x00000801	Write	Kinematic group	VOID			Start Kinematic group	Reserved function, no standard!

5.4.1.4 Specification Axes

5.4.1.4.1 "Index offset" specification for axis parameter (Index group 0x4000 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00000	Read	every (Structure for all axis parameters)	{			General AXIS PARAMETER STRUCTURE (NC/CNC), also	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						contains the sub-elements such as encoder, controller and drive (s. MC_ReadParameterSet in TcMc.lib)	
			UINT32	1		Axis ID	
			UINT8[30+1+1]	e.g. mm		Axis name	
			UINT32	1		Axis type	
			. . .	. . .	. . .	. . .	
			}			512 bytes	
0x00000001	Read	every	UINT32	1		Axis ID	
0x00000002	Read	every	UINT8[30+1]	1		Axis name	
0x00000003	Read	every	UINT32	ENUM		Axis type	
0x00000004	Read	every	UINT32	µs		Cycle time axis (SAF)	
0x00000005	Read	every	UINT8[10+1]	1		Physical unit	
0x00000006	Read/Write	every	REAL64	e.g. mm/s		Ref. velocity in cam direction	
0x00000007	Read/Write	every	REAL64	e.g. mm/s		Ref. velocity in sync direction	
0x00000008	Read/Write	every	REAL64	e.g. mm/s		Velocity hand slow	
0x00000009	Read/Write	every	REAL64	e.g. mm/s		Velocity hand fast	
0x0000000A	Read/Write	every	REAL64	e.g. mm/s		Velocity rapid traverse	
0x0000000F	Read/Write	every	UINT16	1	0/1	Position range monitoring?	
0x00000010	Read/Write	every	REAL64	e.g. mm		Position range window	
0x00000011	Read/Write	every	UINT16	1	0/1	Motion monitoring?	
0x00000012	Read/Write	every	REAL64	s		Motion monitoring time	
0x00000013	Read/Write	every	UINT16	1	0/1	Loop?	
0x00000014	Read/Write	every	REAL64	e.g. mm		Loop movement (±)	
0x00000015	Read/Write	every	UINT16	1	0/1	Target position monitoring?	
0x00000016	Read/Write	every	REAL64	e.g. mm		Target position window	
0x00000017	Read/Write	every	REAL64	s		Target position monitoring time	
0x00000018	Read/Write	every	REAL64	e.g. mm		Pulse way in pos. direction	
0x00000019	Read/Write	every	REAL64	e.g. mm		Pulse way in neg. direction	
0x0000001A	Read/Write	every	UINT32	ENUM (≥0)		Error reaction mode: 0: instantaneous (default) 1: delayed (e.g. for master/slave coupling)	From TC 2.11
0x0000001B	Read/Write	every	REAL64	s	[0...1000]	Error reaction delay (if error reaction mode "delayed" is activated)	From TC 2.11

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x0000001C	Read/Write	every	UINT16	1	0/1	Couple slaves via actual values if not operational?	From TC 2.11
0x0000001D	Read/Write	every	REAL64	e.g. mm/s^2	[0, 0.01...1.0E10]	Acceleration for fading profile when switching from SET to ACTUAL values: Default: 0 (in this case the minimum from the axis acceleration is used, i.e. MIN(Acc, Dec))	From TC 2.11 R2
0x0000001E	Read/Write	every	UINT32	ENUM (≥0)		Fast Axis Stop Signal Type: Signal type selection to force a Fast Axis Stop (s. Bit 7 from Drive->n-Status4)*"0 (Signal-Type_OFF)","1 (Signal-Type_RisingEdge)","2 (Signal-Type_FallingEdge)","3 (Signal-Type_Both-Edges)","4 (Signal-Type_HighActive)","5 (Signal-Type_LowActive)"*	From TC 2.11 R3
0x00000020	Read/Write	every	UINT16	1	0/1	Allow motion commands for slave axis? Default: FALSE	From TC 2.11
0x00000021	Read/Write	every	UINT16	1	0/1	Allow motion commands for axes with active external set-point generator? Default: FALSE	From TC 2.11 R2
0x00000026	Read/Write	every	UINT32	1		Interpretation of the units (position, velocity, time) Bit 0: velocity in x/min instead of x/s Bit 1: position in thousands of the base unit Bit 2: modulo position display	See encoder! bit array
0x00000027	Read/Write	every	REAL64	e.g. mm/s		Max. allowed velocity	
0x00000028	Read/Write	every	REAL64	e.g. mm		Motion monitoring window	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000029	Read/Write	every	UINT16	1	0/1	PEH time monitoring?	Position end and accurate stop
0x0000002A	Read/Write	every	REAL64	s		PEH monitoring time	
0x0000002B	Read/Write	every	UINT16	1	0/1	Backlash compensation?	
0x0000002C	Read/Write	every	REAL64	e.g. mm		Backlash	
0x00000030	Read	every	UINT16	1	[0,1]	Persistent data e.g. for actual position and reference state of the encoder?	Boot parameter

0x00000031	Read	every	{
			UINT8[6]
			UINT16
			}
				AmsAddr	1	Read the hardware AMS address (AMS Net ID and device port)	OLD!

0x00000031	Read	every	{
			UINT8[6]
			UINT16
			UINT16
			}
				AmsAddr
				ChannelNo
					1	Read the hardware AMS address (AMS Net ID and device port) and the channel number (0=Channel A, 1=Channel B)	

0x00000033	Read	every	{ UINT16 ApplRequestBit UINT16 ApplRequestType UINT32 ApplCmdNo UINT32 ApplCmdVersion . . . } 1024 Byte	1 1 1 1	0/1 ≥0 >0 ≥0	General APPLICATION REQUEST STRUCTURE (NC/NCI), e.g. for ApplicationHoming request (s. *MC_ReadApplicationRequest* in *TcMc2.lib*)	From TC 2.11 R2
0x00000051	Read	Channel: every	UINT32			Channel ID	
0x00000052	Read	Channel: every	UINT8[30+1]			Channel name	
0x00000053	Read	Channel: every	UINT32			Channel type	
0x00000054	Read	Group: every	UINT32			Group ID	
0x00000055	Read	Group: every	UINT8[30+1]			Group name	
0x00000056	Read	Group: every	UINT32			Group type	
0x00000057	Read	every	UINT32			Number of encoders	
0x00000058	Read	every	UINT32			Number of controllers	
0x00000059	Read	every	UINT32			Number of drives	
0x0000005A	Read	every	{			Read all subelements of an axis:	
			UINT32[9]	1	[0, 1...255]	Axis encoder IDs	
			UINT32[9]	1	[0, 1...255]	Axis controller IDs	
			UINT32[9]	1	[0, 1...255]	Axis drive IDs	
			} 108 bytes				
0x00000101	Read/Write	Servo	REAL64	e.g. mm/s^2		Acceleration	
0x00000102	Read/Write	Servo	REAL64	e.g. mm/s^2		deceleration	
0x00000103	Read/Write	Servo	REAL64	e.g. mm/s^3		Jerk	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000104	Read/Write	Servo	REAL64	s	[0.0 ... 1.0]	Damping time between velocity and position values of the setpoint generator in seconds	Default value: 0.0 s
0x00000105	Read/Write	Servo	UINT32	ENUM		Override type for velocity: 1: related to internal reduced velocity (without iteration) 2: related to original external start velocity (without iteration) 3: related to internal reduced velocity (optimization by means of iteration) 4: related to original external start velocity (optimization by means of iteration)	Default value: type 1
0x00000106	Read/Write	Servo	REAL64	1	[0.0 ... 1.0E6]	Maximum allowed velocity jump for dynamic reduction *DV = factor * min(A+, A-) * DT*	Default value: 0.0
0x00000107	Read/Write	Servo	UINT16	1	[0,1]	Activates acceleration and jerk limitation for the auxiliary axis (Q1 to Q5)	Default value: 1
0x00000108	Read/Write	Servo	REAL64	e.g. mm	[0.0..1000.0]	Radius of the tolerance sphere for the auxiliary axes	From TC V2.9 B932
0x00000109	Read/Write	Servo	REAL64	e.g. mm	[0.0..10000.0]	Maximum allowed position deviation if the tolerance sphere is reduced only for auxiliary axes	From TC V2.9 B1013
0x0000010A	Read/Write	Servo	REAL64	e.g. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: Acceleration (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3
0x0000010B	Read/Write	Servo	REAL64	e.g. mm/s^2	[0.01 ... 1.0E20]	Fast Axis Stop: deceleration (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3
0x0000010C	Read/Write	Servo	REAL64	e.g. mm/s^3	[0.1 ... 1.0E30]	Fast Axis Stop: jerk (s.a. Fast Axis Stop Signal Type)	From TC 2.11 R3

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000201	Read/Write	Stepper motor	UINT32	ENUM		Operation mode stepper motor	
0x00000202	Read/Write	Stepper motor	REAL64	e.g. mm/STEP	[1.0E-6 ... 1000.0]	Distance scaling of a motor step	
0x00000203	Read/Write	Stepper motor	REAL64	e.g. mm/s	[0.0 ... 1000.0]	Minimum velocity for velocity profile	
0x00000204	Read/Write	Stepper motor	UINT32	1	[0 ... 100]	Number of steps per frequency/velocity step	
0x00000205	Read/Write	Stepper motor	UINT32	1		Motor mask as sync pulse	Not implemented!
0x00000301	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance in pos. direction	
0x00000302	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance in neg. direction	
0x00000303	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Braking distance in pos. direction	
0x00000304	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Braking distance in neg. direction	
0x00000305	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Braking deceleration in pos. direction	
0x00000306	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Braking deceleration in neg. direction	
0x00000307	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Switching time from high to low speed	
0x00000308	Read/Write	high/low	REAL64	e.g. mm	[0.0 ... 100000.0]	Creep distance stop	
0x00000309	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Damping time to release brake	
0x0000030A	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Pulse time in pos. direction	
0x0000030B	Read/Write	high/low	REAL64	s	[0.0 ... 60.0]	Pulse time in neg. direction	
ENCODER:							
0x00n10001	Read	Encoder: every	UINT32	1	[1 ... 255]	Encoder ID n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional)	
0x00n10002	Read	Encoder: every	UINT8[30+1]	1	30 characters	Encoder name	
0x00n10003	Read	Encoder: every	UINT32	1	s. ENUM (>0)	Encoder type [▶ 137]	
0x00n10004	Read/Write	Encoder: every	UINT32	1	Byteoffset	Input address offset (IO-Input-Image)	change I/O address
0x00n10005	Read/Write	Encoder: every	UINT32	1	Byteoffset	Output address offset (IO-Output-Image)	change I/O address

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n10006	Read/Write	Encoder: every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Resulting scaling factor (numerator / denominator) Note: from TC 3.0 the scaling factor consists of two components – numerator and denominator (default: 1.0).	
0x00n10007	Read/Write	Encoder: every	REAL64	e.g. mm	[±1.0E+9]	Position offset	
0x00n10008	Read/Write	Encoder: every	UINT16	1	[0,1]	Encoder count direction	
0x00n10009	Read/Write	Encoder: every	REAL64	e.g. mm	[0.001 ... 1.0E+9]	Modulo factor	
0x00n1000A	Read/Write	Encoder: every	UINT32	1	s. ENUM (>0)	Encoder mode [▶ 138]	
0x00n1000B	Read/Write	Encoder: every	UINT16	1	0/1	Soft end min. monitoring?	
0x00n1000C	Read/Write	Encoder: every	UINT16	1	0/1	Soft end max. monitoring?	
0x00n1000D	Read/Write	Encoder: every	REAL64	mm		Soft end position min.	
0x00n1000E	Read/Write	Encoder: every	REAL64	mm		Soft end position max.	
0x00n1000F	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder evaluation direction [▶ 138] (enable for log. counting direction)	s. appendix
0x00n10010	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for actual position value in seconds (P-T1)	
0x00n10011	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for actual velocity value in seconds (P-T1)	
0x00n10012	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for actual acceleration value in seconds (P-T1)	
0x00n10013	Read/Write	Encoder: every	UINT8[10+1]	1		Physical unit	Not implemented!
0x00n10014	Read/Write	Encoder: every	UINT32	1		Interpretation of the units (position, velocity, time) Bit 0: velocity in x/min instead of x/s Bit 1: position in thousands of the base unit	Not implemented! bit array
0x00n10015	Read	Encoder: every	UINT32	INC	[0x0... 0xFFFFFFFF]	Encoder mask (maximum value of the encoder actual value in increments) Note: From TwinCAT 2.11 R2 the encoder mask can be any numerical value (e.g.	Read-only parameter s. parameter "Encoder Sub Mask"

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						3600000) and no longer needs to be a continuous sequence of binary ones (2^n-1).	
0x00n10016	Read/Write	Encoder: every	UINT16	1	0/1	Actual position correction (measurement system error correction)?	
0x00n10017	Read/Write	Encoder: every	REAL64	s	[0.0...60.0]	Filter time for actual position correction in seconds (P-T1)	
0x00n10019	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder absolute dimensioning system [▶ 139]	s. appendix
0x00n1001A	Read/Write	Encoder: every	UINT32	1	s. ENUM (≥0)	EncoderPositionInitialization [▶ 139]	Not implemented!
0x00n1001B	Read/Write	Encoder: every	REAL64	e.g. mm	[≥0, modulo factor/2]	Tolerance window for modulo-start	
0x00n1001C	Read	Encoder: every	UINT32	1	s. ENUM (≥0)	Encoder sign interpretation (data type) [▶ 138]	
0x00n1001D	Read	Encoder: every	UINT16	1	0/1	Incremental or absolute encoder? 0: incremental encoder type 1: absolute encoder type	
0x00n10023	Read/Write	Encoder: every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: Counter (=> scaling factor numerator / scaling factor denominator)	From TC 3.0
0x00n10024	Read/Write	Encoder: every	REAL64	1	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator) Default: 1.0	From TC 3.0

0x00n10025	Read/Write	Encoder: every	{
			REAL64
			REAL64
			}
				e.g. mm/INC
				1
					[1.0E-12 ... 1.0E+30]
					[1.0E-12 ... 1.0E+30]
						Component of the scaling factor: Counter
						Component of the scaling factor: denominator
						(=> scaling factor numerator / scaling factor denominator)
							From TC 3.0

0x00n10030	Read/Write	Encoder: every	UINT32	1		Internal Encoder Control DWORD for	From 211R3 B2227

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						specifying the operation modes and properties	
0x00n10101	Read/Write	E: INC	UINT16	1	[0,1]	Inverse search direction for ref.cam?	
0x00n10102	Read/Write	E: INC	UINT16	1	[0,1]	Inverse search direction for sync pulse?	
0x00n10103	Read/Write	E: INC	REAL64	e.g. mm	[±1000000.0]	Reference position	
0x00n10104	Read/Write	E: INC	UINT16	1	[0,1]	Distance monitoring between Ref. cams and sync pulse active?	Not implemented!
0x00n10105	Read/Write	E: INC	UINT32	INC	[0 ... 65536]	Minimum distance between Ref. cams and sync pulse in increments	Not implemented
0x00n10106	Read/Write	E: INC	UINT16	1	[0,1]	External sync pulse?	
0x00n10107	Read/Write	E: INC	UINT32	1	s. ENUM (>0)	Reference mode	s. appendix
0x00n10108	Read/Write	E: INC	UINT32	1	[0x0000000F... 0xFFFFFFFF] Binary mask: $(2^n - 1)$	Encoder Sub Mask (maximum value of the absolute range of the encoder actual value in increments) Used, for example, as a reference mark for the referencing mode "Software Sync" and for the NC Retain Data "ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)". Note 1: The Encoder Sub Mask must be smaller than or equal to the Encoder Mask. Note 2: The Encoder Mask must be an integer multiple of the Encoder Sub Mask. Note 3: The Encoder Sub Mask must be a continuous sequence of binary ones (2^n-1), e.g. 0x000FFFFF.	NEW s. parameter "Encoder Mask"

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n10110	Read/Write	E: INC (encoder simulation)	REAL64	1	[0.0 ... 1000000.0]	Scaling/weighting of the noise component for the simulation encoder	
CONTROLLER:							
0x00n20001	Read	Controller: every	UINT32	1	[1 ... 255]	Controller ID n = 0: standard controller of the axis n > 0: n-th controller of the axis (optional)	
0x00n20002	Read	Controller: every	UINT8[30+1]	1	30 characters	Controller name	
0x00n20003	Read	Controller: every	UINT32	1	s. ENUM (>0)	Controller type [▶ 136]	
0x00n2000A	Read/Write	Controller: every		1	s. ENUM (>0)	Controller mode	
0x00n2000B	Read/Write	Controller: every	REAL64	%	[0.0 ... 1.0]	Weight of the velocity precontrol (standard value: 1.0 = 100 %)	
0x00n20010	Read/Write	Controller: every	UINT16	1	0/1	Following error monitoring position?	
0x00n20011	Read/Write	Controller: every	UINT16	1	0/1	Following error monitoring velocity?	
0x00n20012	Read/Write	Controller: every	REAL64	e.g. mm		Max. following error position	
0x00n20013	Read/Write	Controller: every	REAL64	s		Max. following error time position	
0x00n20014	Read/Write	Controller: every	REAL64	e.g. mm/s		Max. following error velocity	
0x00n20015	Read/Write	Controller: every	REAL64	s		Max. following error time velocity	
0x00n20100	Read/Write	P/PID (Pos., (velocity)	REAL64	1	[0.0...1.0]	Maximum output limitation (±) for controller total output	(standard value: 0.5 == 50%)
0x00n20102	Read/Write	P/PID (pos.)	REAL64	e.g. mm/s/ mm	[0.0...1000.0]	Proportional gain kp or kv respectively Unit: base unit / s / base unit	position control
0x00n20103	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Integral action time Tn	Position control
0x00n20104	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Derivative action time Tv	Position control
0x00n20105	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Damping time Td	Position control
0x00n20106	Read/Write	PP (Pos.)	REAL64	e.g. mm/s/ mm	[0.0...1000.0]	Additional proportional gain, kp or kv respectively, that applies above a limit velocity in percent.	Position control

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Unit: base unit / s / base unit	
0x00n20107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Threshold velocity in percent above which the additional proportional gain, kp or kv respectively, applies	
0x00n20108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportional gain ka	Acceleration pre control
0x00n2010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	"dead band" for position error (control deviation) (for P/PID controllers with velocity or torque interface)	Reserved function
0x00n2010F	Read/Write	P/PP/PID (Pos.) slave control	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave coupling difference control: proportional gain k_{cp}	Slave coupling difference control:
0x00n20110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: active/passive	
0x00n20111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: hold mode	
0x00n20112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: fading mode	
0x00n20114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatic offset calibration: pre-control limit	
0x00n20115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatic offset calibration: time constant	
0x00n20116	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I-part in percent (default setting: 0.1 = 10%)	
0x00n20117	Read/Write	PID (pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D-part in percent (default setting: 0.1 = 10%)	
0x00n20118	Read/Write	PID (pos.)	UINT16	1	0/1	Deactivation of the I-component during an active positioning process (assuming I-component active)? (default setting: 0 = FALSE)	
0x00n20120	Read/Write	P/PID (pos.)	REAL64	s	≥0	PT-1 filter value for position error (pos. control difference)	Reserved function, no standard!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20202	Read/Write	P/PID (velocity)	REAL64	1	[0.0...1000.0]	Proportional gain kp or kv respectively	Velocity control
0x00n20203	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	integral action time Tn	Velocity control
0x00n20204	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Derivative action time Tv	Velocity control
0x00n20205	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Damping time Td	Velocity control
0x00n20206	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I-part in percent (default setting: 0.1 = 10%)	Velocity control
0x00n20207	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D-part in percent (default setting: 0.1 = 10%)	velocity control
0x00n2020D	Read/Write	P/PID (velocity)	REAL64	mm/s	[0.0 ... 10000.0]	"dead band" for velocity error (control deviation) (for P/PID controllers with velocity or torque interface)	Reserved function
0x00n20220	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-2 filter value for velocity error (vel. control difference)	Velocity control no standard!
0x00n20221	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-1 filter value for velocity error (vel. control difference)	Reserved function, no standard!
0x00n20250	Read/Write	P/PI (observer)	UINT32	1	s. ENUM (≥0)	Observer mode [▶ 136] for controller with torque interface 0: OFF (default) 1: LUENBERGER	From TC 2.10 Build 1320
0x00n20251	Read/Write	P/PI (observer)	REAL64	Nm / A	>0.0	Motor: torque constant K_T	
0x00n20252	Read/Write	P/PI (observer)	REAL64	kg m^2	>0.0	Motor: moment of inertia J_M	
0x00n20253	Read/Write	P/PI (observer)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandwidth f_0	
0x00n20254	Read/Write	P/PI (observer)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Correction factor k_c	
0x00n20255	Read/Write	P/PI (observer)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Velocity filter (1^{st} order): time constant T	
0x00n20A03	Read/Write	P/PID (MW)	REAL64	cm^2	[0.0 ... 1000000]	Cylinder area A $_A$ of the A side in cm^2	Reserved parameters !
0x00n20A04	Read/Write	P/PID (MW)	REAL64	cm^2	[0.0 ... 1000000]	Cylinder area AB of the B side in cm^2	Reserved parameters !

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20A05	Read/Write	P/PID (MW)	REAL64	cm^3/s	[0.0 ... 1000000]	Nominal volume flow Q_{nom} in cm^3/s	Reserved parameters !
0x00n20A06	Read/Write	P/PID (MW)	REAL64	bar	[0.0 ... 1000000]	Nominal pressure or valve pressure drop, P_{nom} in bar	Reserved parameters !
0x00n20A07	Read/Write	P/PID (MW)	UINT32	1	[1 ... 255]	Axis ID for the system pressure Po	Reserved parameters !
DRIVE:							
0x00n30001	Read	Drive: every	UINT32	1	[1 ... 255]	Drive ID	
0x00n30002	Read	Drive: every	UINT8[30+1]	1	30 characters	Drive name	
0x00n30003	Read	Drive: every	UINT32	1	s. ENUM (>0)	Drive type [▶ 140]	
0x00n30004	Read/Write	Drive: every	UINT32	1	Byte offset	Input address offset (IO-Input-Image)	Change I/O address
0x00n30005	Read/Write	Drive: every	UINT32	1	Byte offset	Output address offset (IO-Output-Image)	Change I/O address
0x00n30006	Read/Write	Drive: every	UINT16	1	[0,1]	Motor polarity	
0x00n3000A	Read/Write	Drive: every	UINT32	1	s. ENUM (>0)	Drive mode	
0x00n3000B	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Minimum output limit (output limitation) (default setting: -1.0 = -100%)	
0x00n3000C	Read/Write	Drive: every	REAL64	%	[-1.0 ... 1.0]	Maximum output limit (output limitation) (default setting: 1.0 = 100%)	
0x00n3000D	Read	Drive: every	UINT32	INC		Maximum number of output increments (output mask)	
0x00n30010	Read/Write	Drive: every	UINT32	1		Internal Drive Control DWord to determine the drive operation modes	Reserved!
0x00n30011	Read/Write	every	UINT32	1	≥ 5	Internal Drive Reset Counter (time in NC cycles for enable and reset)	Reserved!
0x00n30101	Read/Write	D: Servo	REAL64	e.g. mm/s	>0.0	Reference velocity at reference output (velocity precontrol)	
0x00n30102	Read/Write	D: Servo	REAL64	%	[0.0 ... 5.0]	Reference output in percent (default setting: 1.0 = 100%)	
0x00n30103	Read	D: Servo	REAL64	e.g. mm/s	>0.0	Resulting velocity at 100% output	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n30104	Read/Write	D: Servo	REAL64	e.g. mm/s	±∞	velocity offset (DAC offset) for drift calibration (offset calibration) of the axis	
0x00n30105	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Velocity scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00n30106	Read/Write	D: Profi Drive DSC	UINT32	0.001 * 1/s	≥ 0	Profibus/Profi Drive DSC: position control gain Kpc	Only for Profi Drive DSC
0x00n30107	Read/Write	D: Profi Drive DSC	REAL64	1	≥ 0.0	Profibus/Profi Drive DSC: scaling for calculating 'XERR' (Default: 1.0)	Only for Profi Drive DSC
0x00n30109	Read/Write	D: Servo (Sercos, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Position scaling (scaling factor to react to the weight in the drive)	For Sercos, CANopen
0x00n3010A	Read/Write	D: Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	acceleration scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00n30120	Read/Write	D: Servo/ Hydraulic/	UINT32	1	≥ 0	Table ID (0: no table)	Only for KL4xxx, M2400, Universal
0x00n30121	Read/Write	D: Servo/ Hydraulic	UINT32	1	≥ 0	Interpolation type 0: Linear 2: Spline	Only for KL4xxx, M2400, Universal
0x00n30122	Read/Write	Servo/ Hydraulic	REAL64	%	[-1.0 ... 1.0]	Output offset in percent Note: Acts according to the characteristic evaluation !	Only for KL4xxx, M2400, Universal
0x00n30151	Read/Write	D: Servo / nonlinear	REAL64	1	[0.0 ... 100.0]	Quadrant equalizing factor (relation between quadrants I and III)	
0x00n30152	Read/Write	D: Servo / nonlinear	REAL64	1	[0.01 ... 1.0]	Velocity reference point in percent (1.0 = 100 %)	
0x00n30153	Read/Write	D: Servo / nonlinear	REAL64	1	[0.01 ... 1.0]	Output reference point in percent (1.0 = 100 %)	
0x00030301	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 1	
0x00030302	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 2	
0x00030303	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 3	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00030304	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 4	
0x00030305	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 5	
0x00030306	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 6	
0x00030307	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 7	
0x00030308	Read/Write	D: Stepper motor	UINT8	1		Bit mask: cycle 8	
0x00030310	Read/Write	D: Stepper motor	UINT8	1		Bit mask: holding current	

5.4.1.4.2 "Index offset" specification for axis state (Index group 0x4100 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00000	Read	every (online structure for axis data)	{			AXIS ONLINE STRUCTURE (NC/CNC)	Cannot be traced by oscilloscope! (NCAXISSTATE_ ONLINESTRUCT)
			INT32	1		Error state	
			REAL64	e.g. mm		Actual position	
			REAL64	e.g. degree		Modulo actual position	
			REAL64	e.g. mm		Set position	
			REAL64	e.g. degree		Modulo set position	
			REAL64	e.g. mm/s		Optional: actual velocity	
			REAL64	e.g. mm/s		Set velocity	
			UINT32	%	0...1000000	Speed override (1000000 == 100%)	
			REAL64	e.g. mm		Following error position	
			REAL64	e.g. mm		PeakHold value for max. negative following error (pos.)	
			REAL64	e.g. mm		PeakHold value for max. positive following error (pos.)	
			REAL64	%		Controller output in percent	
			REAL64	%		Total output in percent	
			UINT32	1	≥ 0	Axis-Status-DWord	
			UINT32	1	≥ 0	Axis-Control-DWord	
			UINT32	1	≥ 0	Slave coupling state (state)	
			UINT32	1	0; 1,2,3...	Axis control loop index	
			}			112 bytes	
0x00000001	Read	every	UINT32	1		Error code axis state	*Symbolic access:* *'ErrState"*
0x00n00009	Read	every	UINT32	1	≥ 0	Set cycle counter	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						(SAF-Time-stamp)	
0x00n0000A	Read	every	REAL64	e.g. mm		Set position	*Symbolic access:* *'SetPos"*
0x00n0000B	Read	every	REAL64	e.g. degree		Modulo set position	*Symbolic access:* *'SetPosModulo"*
0x00n0000C	Read	every	INT32	1		Modulo set rotation	
0x00n0000D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Set direction	
0x00n0000E	Read	every	REAL64	e.g. mm/s		Set velocity	*Symbolic access:* *'SetVelo"*
0x00n0000F	Read	every	REAL64	e.g. mm/s^2		Set acceleration	*Symbolic access:* *'SetAcc"*
0x00n00010	Read	every	REAL64	e.g. mm/s^3		Set jerk (time derivation of the set acceleration)	
0x00n00011	Read	every	REAL64	Nm resp. N		Set torque (rotary motor) resp. set force (linear motor)	
0x00n00012	Read	every	REAL64	1		Set coupling factor (set gear ratio)	
0x00n00013	Read	every	REAL64	e.g. mm		Expected target position	
0x00n00014	Read	Servo	{			Remaining travel time and distance (SERVO):	Always to SAF Port 501!
			REAL64	s	≥ 0	Remaining travel time	
			REAL64	e.g. mm	≥ 0	Remaining distance	
			}				
0x00n00015	Read	every	UINT32	1	≥ 0	Set command number	
0x00n00016	Read	Servo	REAL64	s	≥ 0	Positioning time of the last motion command (Start → target position window)	
0x00000018	ReadWrite	Servo	**Write**			Read the "Stop information" (stop distance, stop time)	From TC 2.11 R2 Only port 500!
			REAL64	e.g. mm/s^2	≥ 0	Deceleration for axis stop	
			REAL64	e.g. mm/s^3	≥ 0	Jerk for axis stop	
			Read				
			REAL64	e.g. mm	≥ 0	Stop distance	
			REAL64	s	≥ 0	Stop time	
0x00n0001A	Read	every	REAL64	e.g. mm		Uncorrected set position	
0x00n0001D	Read	every	REAL64	1	[-1.0, 0.0, 1.0]	Uncorrected set direction	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n0001E	Read	every	REAL64	e.g. mm/s		Uncorrected set velocity	
0x00n0001F	Read	every	REAL64	e.g. mm/s^2		Uncorrected set acceleration	
0x00000020	Read	every	UINT32	1	s. ENUM	Coupling state (state)	
0x00000021	Read	every	UINT32	1	≥ 0	Coupling table index	
0x00000022	Read	Servo Master/Slave coupling Type: LINEAR, (&SPECIAL)	{			Reading the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: Gearing factor	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000023	Read	Servo Master/Slave coupling Type: LINEAR, (&SPECIAL)	REAL64	1	[±1000000.0]	reading the gearing factor (SERVO): Type: LINEAR	
0x00000024	Read	Servo	UINT32	1	≥ 0	Number/index of the active axis control loop (triple of encoder, controller and axis interfaces)	
0x00000025	Read	Servo	UINT16	1	0/1	External set value setting via axis interface PCLtoNC active?	
0x00000026	Read	Servo Master/Slave coupling Type: SYNCHRONIZING	REAL64 [64]	1	±∞	Read the characteristic values from the profile for synchronizing the slave axis Type: SYNCHRONIZING	
0x00000027	ReadWrite	Servo Master/Slave coupling Type: TABULAR, MF	**Write**			Read the "Tabular coupling information"	Only port 500!
			VOID or REAL64	e.g. mm	±∞	- No data for the "current information", - Optional for a certain "master axis position"	
			Read				
			REAL64 [32]		±∞	Read the tabular coupling information structure	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000028	ReadWrite	Servo Master/Slave coupling Type: MULTI-CAM (CamAddition)	**Write**			Read the "multi tabular (CamAddition) coupling information"	Only port 500!
			UINT32	1	≥ 0	Table ID to which the query relates	
			Read				
			96 bytes			Read the multi tabular coupling information structure	
0x00000029	Read	Servo	UINT32	1		Delayed error code (error prewarning) in case of a delayed error reaction (see bit *ErrorPropagationDelayed*)	From TC 2.11 R3 B2229
0x0000002A	Read	Servo	REAL64	e.g. mm	±∞	Position difference while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002B	Read	Servo	REAL64	e.g. mm/s	±∞	Relative velocity while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002C	Read	Servo	REAL64	e.g. mm/s ^2	±∞	Relative acceleration while fading from set position to actual position (fading part)	From TC 2.11 R2
0x0000002D	Read	Servo	UINT32	1	≥ 0	Counter for initialization command (InitializeCommandCounter)	
0x0000002E	Read	Servo	UINT32	1	≥ 0	Counter for reset command (ResetCommandCounter)	
0x00000050	Read	every	UINT32	1		Set drive phase (SWGenerator)	Cannot be traced by oscilloscope!
0x00000051	Read	every	UINT16	1		Is the axis deactivated?	Cannot be traced by oscilloscope!
0x00n00060	Read/Write	every (online set value structure)	{			AXIS SET VALUE STRUCTURE (NC/CNC)	Cannot be traced by oscilloscope!
			REAL64	e.g. mm		Set position	
			REAL64	e.g. mm/s		Set velocity	
			REAL64	e.g. mm/s^2		Set acceleration/deceleration	
			REAL64	1	[-1.0, 0.0, 1.0]	Set travel direction	
			REAL64	e.g. mm/s^3		Set jerk	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	Nm resp. N		Set torque resp. set force (new from TC V2.11 B1514)	
			}				
0x00n00061	Read/Write	every (online dynamic set value structure)	{			AXIS DYNAMIC SET VALUE STRUCTURE (NC/CNC)	
			REAL64	e.g. mm/s		Set velocity	
			REAL64	e.g. mm/s^2		Set acceleration/deceleration	
			REAL64	1	[-1.0, 0.0, 1.0]	Set travel direction	
			REAL64	e.g. mm/s^3		Set jerk	
			REAL64	Nm resp. N		Set torque resp. set force (new from TC V2.11 B1514)	
			}				
0x00n10002	Read	every (Encoder)	REAL64	e.g. mm		Actual position (charge with actual position compensation value) n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional)	*symbolic access:* *'ActPos"*
0x00n10003	Read	every (Encoder)	REAL64	e.g. degree		Modulo actual position	*symbolic access:* *'ActPosModulo'*
0x00n10004	Read	every (Encoder)	INT32	1		Modulo actual rotation	
0x00n10005	Read	every (Encoder)	REAL64	e.g. mm/s		Optional: actual velocity	*symbolic access:* *'ActVelo"*
0x00n10006	Read	every (Encoder)	REAL64	e.g. mm/s^2		Optional: actual acceleration	*symbolic access:* *'ActAcc"*
0x00n10007	Read	every (Encoder)	INT32	INC		Encoder actual increments	
0x00n10008	Read	every (Encoder)	INT64	INC		Software - actual increment counter	
0x00n10009	Read	every (Encoder)	UINT16	1	0/1	Reference flag ("calibrate flag")	
0x00n1000A	Read	every (Encoder)	REAL64	e.g. mm		Actual position correction value (measurement system error correction)	
0x00n1000B	Read	every (Encoder)	REAL64	e.g. mm		Actual position without actual position compensation value	Cannot be traced by oscilloscope!
0x00n10010	Read	every (Encoder)	REAL64	e.g. mm/s		Actual velocity without actual position compensation value	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n10012	Read	every (Encoder)	REAL64	e.g. mm		Unfiltered actual position (charge with actual position compensation value)	
0x00n10015	Read	every (Encoder)	REAL64	e.g. mm/s		Optional: Unfiltered actual velocity	Base Unit / s
0x00n10101	Read	INC (Encoder)	REAL64	e.g. mm		Read back of the position difference between activation of the internal hardware latch and the time when it becomes valid	Cannot be traced by oscilloscope!
0x00n20001	Read	R: every	INT32	1		Error state of the controller n = 0: standard controller of the axis n > 0: n-th controller of the axis (optional)	
0x00n20002	Read	R: every	REAL64	e.g. mm/s		Controller output in absolute units	*symbolic access:* *'CtrlOutput''*
0x00n20003	Read	R: every	REAL64	%		Controller output in percent	Cannot be traced by oscilloscope!
0x00n20004	Read	R: every	REAL64	V		Controller output in volts	Cannot be traced by oscilloscope!
0x00n2000D	Read	R: every	REAL64	e.g. mm		Following error position (without dead time compensation)	Base Unit
0x00n2000F	Read	R: every	REAL64	e.g. mm		Following error position (with dead time compensation)	*symbolic access:* *'PosDiff''*
0x00n20010	Read	R: every	REAL64	e.g. mm		Peak hold value for maximum negative following error of the position	
0x00n20011	Read	R: every	REAL64	e.g. mm		Peak hold value for minimum positive following error of the position	
0x00n20012	Read	R: every	REAL64	e.g. mm/s		Following error velocity	Not implemented!
0x00n20021	Read	R: every	REAL64	e.g. mm		Difference (deviation) between the following error from master and slave axis	*symbolic access:* *'PosDiffCouple'*

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						(master error minus slave error)	
0x00n20022	Read	R: every	REAL64	e.g. mm		PeakHold value for the maximum negative difference between master and slave following error of the position	Base Unit
0x00n20023	Read	R: every	REAL64	e.g. mm		PeakHold value for the maximum positive difference between master and slave following error of the position	Base Unit
0x00n20101	Read	R: P/PID (pos.)	REAL64	e.g. mm/s		P-part of the controller in absolute units	
0x00n20102	Read	R: PID (pos.)	REAL64	e.g. mm/s		I-part of the controller in absolute units	
0x00n20103	Read	R: PID (pos.)	REAL64	e.g. mm/s		D-part of the controller in absolute units	
0x00n20104	Read	R: PID (pos.)	UINT16	1	0/1	Limitation of the I-part active?	
0x00n20105	Read	R: PID (pos.)	UINT16	1	0/1	Limitation of the D-part active?	
0x00n20106	Read	R: PID (pos.)	UINT16	1	0/1	ARW measure for the I-part active? ARW: Anti Reset Windup	Not implemented!
0x00n20110	Read	R: PID (pos.)	REAL64	e.g. mm/s		Acceleration pre-control Yacc of the controller in absolute units Note: Function depends on controller type!	Acceleration pre-control
0x00n20111	Read	R: PP (Pos.)	REAL64	mm/s/ mm	≥0	Internal interpolated proportional gain kp or kv	PP controller
0x00n20201	Read	R: P,PID (velocity)	REAL64	e.g. mm/s		Velocity part of the controller	Base Unit / s
0x00n20202	Read	R: P,PID (velocity)	REAL64	%		Velocity part of the controller in percent	Cannot be traced by oscilloscope!
0x00n20203	Read	R: P,PID (velocity)	REAL64	V		Velocity part of the controller in volts	Cannot be traced by oscilloscope!
0x00n20201	Read	R: P/PID (velocity)	REAL64	e.g. mm/s		P-part of the controller in absolute units	
0x00n20202	Read	R: P/PID (velocity)	REAL64	e.g. mm/s		I-part of the controller in absolute units	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n20203	Read	R: P/PID (velocity)	REAL64	e.g. mm/s		D-part of the controller in absolute units	
0x00n20204	Read	R: P/PID (velocity)	UINT16	1	0/1	Limitation of the I-part active?	
0x00n20205	Read	R: P/PID (velocity)	UINT16	1	0/1	Limitation of the D-part active?	
0x00n20206	Read	R: P/PID (velocity)	UINT16	1	0/1	ARW measure for the I-part active?	ARW: Anti Reset Windup
0x00n2020A	Read	R: P/PID (velocity)	REAL64	e.g. mm/s		Total input size of the velocity controller	
0x00n20A00	Read	R: PID (MW)	REAL64	%	[-1.0...1.0]	Offsetting of the set velocity (pre-control)	Reserved parameters !
0x00n20A01	Read	R: PID (MW)	REAL64	e.g. mm/s		P-part of the controller in absolute units or percent (according to output weight)	Reserved parameters !
0x00n20A02	Read	R: PID (MW)	REAL64	e.g. mm/s		I-part of the controller in absolute units or percent (according to output weight)	Reserved parameters !
0x00n20A03	Read	R: PID (MW)	REAL64	e.g. mm/s		D-part of the controller in absolute units or percent (according to output weight)	Reserved parameters !
0x00n20A04	Read	R: PID (MW)	UINT16	1	0/1	Limitation of the I-part active?	Reserved parameters !
0x00n20A05	Read	R: PID (MW)	UINT16	1	0/1	Limitation of the D-part active?	Reserved parameters !
0x00n20A06	Read	R: PID (MW)	UINT16	1	0/1	ARW measure for the I-part active? ARW: Anti Reset Windup	Reserved parameters !
0x00n20A10	Read	R: PID (MW)	REAL64	e.g. mm/s		Acceleration pre-control Yacc of the controller in absolute units	Reserved parameters !
0x00n30001	Read	D: every	INT32	1		Error state of the drive	
0x00n30002	Read	D: every	REAL64	e.g. mm/s		Total output in absolute units	*Symbolic access:* *'DriveOutput'*
0x00n30003	Read	D: every	REAL64	%		Total output in percent	
0x00n30004	Read	D: every	REAL64	V		Total output in volts	Cannot be traced by oscilloscope!
0x00n30005	Read	D: every	REAL64	e.g. mm/s		PeakHold value for maximum negative total output	Base Unit / s
0x00n30006	Read	D: every	REAL64	e.g. mm/s		PeakHold value for maximum positive total output	Base Unit / s

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n30013	Read	D: every	REAL64	%		Total output in percent (according to non-linear characteristic !)	
0x00n30014	Read	D: every	REAL64	V		Total output in volts (according to non-linear characteristic !)	Cannot be traced by oscilloscope!
0x00n3011A	Read	D: Servo (Sercos, CANopen)	REAL64	e. g. mm		Optional Smoothingfilter: Filtered set position	NEW For Sercos, CANopen
0x00n3011E	Read	D: Servo (Sercos, CANopen)	REAL64	e. g. mm/s		Optional Smoothingfilter: Filtered set velocity	NEW For Sercos, CANopen
0x00n3011F	Read	D: Servo (Sercos, CANopen)	REAL64	e. g. mm/s^2		Optional Smoothingfilter: Filtered set acceleration/deceleration	NEW For Sercos, CANopen

5.4.1.4.3 "Index offset" specification for axis functions (Index group 0x4200 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000001	Write	every	VOID			Reset axis	For FIFO axes too!
0x00000002	Write	every	VOID			Stop axis	For FIFO axes too!
0x00000003	Write	every	VOID			Clear axis (task)	For FIFO axes too!
0x00000004	Write	every	{			Emergency Stop (Emergency stop with regulated ramp)	Only for PTP axes!
			REAL64	e.g. mm/s^2	> 0.0	Deceleration (must be greater than or equal to the original deceleration)	
			REAL64	e.g. mm/s^3	> 0.0	Jerk (must be greater than or equal to the original jerk)	
			}				
0x00000005	Write	PTP axis	{			Parametrized stop (with regulated ramp)	Only for PTP axes! Reserved function, no standard!
			REAL64	e.g. mm/s^2	> 0.0	Deceleration	
			REAL64	e.g. mm/s^3	> 0.0	Jerk	
			}				

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000009	Write	PTP axis	{			Oriented stop (oriented end position)	Only for PTP axes!
			REAL64	e.g. degree	≥ 0.0	Modulo end position (modulo target position)	
			REAL64	e.g. mm/s^2	> 0.0	Deceleration (not yet implemented)	
			REAL64	e.g. mm/s^3	> 0.0	Jerk (not yet implemented)	
			}				
0x00000010	Write	every	VOID			Reference axis ("calibration")	
0x00000011	Write	every	{			New end position axis	
			UINT32	ENUM	s. appendix	End position type (see appendix)	
			REAL64	e.g. mm	±∞	New end position (target position)	
			}				
0x00000012	Write	every	{			New end position and new velocity axis	
			UINT32	ENUM	s. appendix	Command type (s. appendix)	
			UINT32	ENUM	s. appendix	End position type (s. appendix)	
			REAL64	e.g. mm	±∞	new end position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	New end velocity (requested travelling speed)	
			REAL64	e.g. mm	±∞	Optional: Switching position from which the new travel profile is activated	
			}				
0x00000015	Write	every	{			New dynamic parameters for active positioning	
			REAL64	e.g. mm/s^2	> 0.0	Acceleration	
			REAL64	e.g. mm/s^2	> 0.0	Deceleration	
			REAL64	e.g. mm/s^3	> 0.0	Optional: Jerk (not yet implemented)	
			}				
0x00000016	**ReadWrite**	every SERVO	**Write** (76 bytes)			Universal axis start (UAS): merge of single commands, such as axis start, and online changes in combination	Always to SAF Port 501!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						with "Buffer Mode" (see TcMc2.lib)	
			{				
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			UINT32	1	≥ 0	Bit mask for checks and operation modes (Default value: 0)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity *Vrequ*	
			REAL64	e.g. mm/s^2	≥ 0.0	Optional: Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Optional: Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Optional: Jerk	
			UINT32	ENUM	s. appendix	Buffer mode (command buffer) [▶ 134]	
			REAL64	e.g. mm	±∞	Optional: Blending position (command blending position)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment start velocity *Vi* *(0 ≤ Vi ≤ Vrequ)*	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment end velocity Vf *(0 ≤ Vf ≤ Vrequ)*	
			}				
			Read				
			{				
			UINT16	1	≥ 0	Command number (job number)	
			UINT16	1	≥ 0	Command status	
			}				
0x00000017	ReadWrite	SERVO	Write (76 bytes)			"Master/slave decoupling" and "Universal axis start (UAS)": Merge of decoupling command of a slave axis (IdxOffset: 0x00000041) and the subsequent universal axis start (UAS) (IdxOffset: 0x00000016)	Not yet released!
			{				
			UINT32	ENUM	s. appendix	Start type (s. appendix)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			UINT32	1	≥ 0	Bit mask for checks and operation modes (Default value: 0)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity Vrequ	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			UINT32	ENUM	s. appendix	Buffer mode (command buffer) [▶ 134]	
			REAL64	e.g. mm	±∞	Optional: Blending position (command blending position)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment start velocity Vi (0 ≤ Vi ≤ Vrequ)	
			REAL64	e.g. mm/s	≥ 0.0	Optional: Segment end velocity Vf (0 ≤ Vf ≤ Vrequ)	
			}				
			Read				
			{				
			UINT16	1	≥ 0	Command number (job number)	
			UINT16	1	≥ 0	Command status	
			}				
0x00000018	Write	every	VOID			Release axis lock for motion commands (TcMc2)	
0x00000019	Write	every	UINT32	1	> 0	Set external axis error (runtime error)	Caution by using!
0x00n0001A	Write	every	{			Set actual axis position	**Caution by using!** For FIFO axes too! Always to SAF Port 501!
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	Actual position for axis n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional)	
			}				
0x00n0001B	Write	every	UINT32	1	0/1	Set reference flag ("calibrate flag") n = 0: standard encoder of the axis	**Caution by using!** For FIFO axes too!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						n > 0: n-th encoder for the axis (optional)	
0x00n0001C	Write	SERVO	{			Set only actual axis position without manipulating the set position (also for slave and with active process)	**Caution by using!**
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	Actual position for axis n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional) **Take care when using!**	
			}				
0x00n0001D	Write	every	{			Set actual value of the axis on the drive side (Position interface and encoder offset of null required!) n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional)	**Caution by using!** Only for CANopen
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	e.g. mm	±∞	Actual position for axis	
			}				
0x00n0001E	Write	every	{			Set a new encoder scaling factor on the fly (also when axis is in motion)	Caution by using! Always to SAF Port 501!
			UINT32	ENUM	1	Encoder scaling factor type 1: Absolute 2: Relative	
			REAL64	e.g. mm/INC	[1.0E-8 ... 100.0]	New encoder scaling factor n = 0: standard encoder of the axis n > 0: n-th encoder for the axis (optional)	
			}				
0x00n0001F	Write	every	{			Set actual axis position on the fly	**Caution by using!** Always to SAF Port 501!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						(also when axis is in motion)	
			UINT32	ENUM		Position type for setting actual value on the fly 1: Absolute 2: Relative	
			UINT32	1		Control DWord, e.g. for "clearing the position lag"	
			REAL64			Reserve	
			REAL64	e.g. mm	±∞	New actual axis position	
			UINT32			Reserve	
			UINT32			Reserve	
			}				
0x00000020	Write	every 1D start	{			Standard axis start:	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥0.0	Required velocity	
			}				
0x00000021	Write	every 1D start	{			Extended axis start (SERVO):	
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	e.g. mm/s	≥ 0.0	Required velocity	
			UINT32	0/1	0/1	Standard acceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			UINT32	0/1	0/1	Standard deceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			UINT32	0/1	0/1	Standard jerk?	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000022	Write	SERVO(MW)	{			Special axis start (SERVO):	Reserved start function, no standard!
			UINT32	ENUM	s. appendix	Start type (s. appendix)	
			REAL64	e.g. mm	±∞	End position (target position)	
			REAL64	mm/s	≥ 0.0	Required start velocity	
			REAL64	e.g. mm	±∞	Position for a new velocity level	
			REAL64	e.g. mm/s	≥ 0.0	New end velocity level	
			UINT32	0/1	0/1	Standard acceleration?	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			UINT32	0/1	0/1	Standard deceleration?	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			UINT32	0/1	0/1	Standard Jerk?	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			}				
0x00000023	Write	SERVO	{			Start external set value setting (setting by cyclic axis interface PLCtoNC)	
			UINT32	ENUM	1: Absolute 2: Relative	Start type	
			REAL64	e.g. mm	±∞	New end position (target position) optional!	
			}				
0x00000024	Write	SERVO	VOID			Stop/disable external set value setting (cyclic axis interface PLCtoNC)	
0x00000025	Write	SERVO	{			Start reversing operation for positioning (SERVO):	
			UINT32	ENUM	1	Start type (default: 1)	
			REAL64	e.g. mm	±∞	End position 1 (target position)	
			REAL64	e.g. mm	±∞	End position 2 (target position)	
			REAL64	0/1	0/1	Required velocity	
			REAL64	s	≥ 0.0	Idle time	
			}				
0x00000026	Write	every	{			Start drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000027	Write	every	VOID			Start drive output	
0x00000028	Write	every	{			Change the drive output:	
			UINT32	ENUM	s. appendix	Output type (s. appendix)	
			REAL64	e.g. %	±∞	Required output value (e.g. %)	
			}				
0x00000029	Write	every	VOID			Instantaneously adopt current override value and freeze until next override change!	Reserved function, no standard!

0x0000002A	Write	every	{
			32 bytes
			}
						Calculate and set encoder offset	Reserved function, no standard!

0x0000002B	ReadWrite	every	WriteData: s. 'UAS' ReadData: s. 'UAS'			Stop external setpoint generator and continuous endless motion ('UAS': Universal axis start)	Reserved function, no standard!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000030	Write	SERVO	{			Start section compensation (SERVO)	
			UINT32	ENUM	s. appendix	Compensation type (s. appendix)	
			REAL64	e.g. mm/s^2	≥ 0.0	Max. acceleration increase	
			REAL64	e.g. mm/s^2	≥ 0.0	Max. deceleration increase	
			REAL64	e.g. mm/s	> 0.0	Max. increase velocity	
			REAL64	e.g. mm/s	> 0.0	Base velocity for the process	
			REAL64	e.g. mm	±∞	Path difference to be compensated	
			REAL64	e.g. mm	> 0.0	Path distance for compensation	
			}				
0x00000030	ReadWrite	SERVO returns the real active values	{ **READ+WRITE:**			Start section compensation (SERVO) Note: contained only in "TcMc2.lib"	
			UINT32	ENUM	s. appendix	Compensation type (s. appendix)	
			REAL64	e.g. mm/s^2	≥ 0.0	=> Max. acceleration increase <= Returns the implemented acceleration increase (new in "TcMc2.lib")	
			REAL64	e.g. mm/s^2	≥ 0.0	=> Max. Deceleration increase <= Returns the implemented deceleration increase (new in "TcMc2.lib")	
			REAL64	e.g. mm/s	> 0.0	=> Requested max. velocity of increase <= Returns the implemented velocity of increase	
			REAL64	e.g. mm/s	> 0.0	Base velocity for the process	
			REAL64	e.g. mm	±∞	=> Requested distance difference to be compensate <= Returns the implemented distance difference	
			REAL64	e.g. mm	> 0.0	=> Requested max. distance for compensation	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						<= Returns implemented distance	
			UINT32	1	≥ 0	<= Returns Warning ID (e.g. 0x4243)	
			}				
0x00000031	Write	SERVO	VOID			Stop section compensation (SERVO)	
0x00000032	Write	SERVO	{			Start reversing operation with velocity steps (SERVO): (can be used to determine the velocity step response)	
			UINT32	ENUM	1	Start type (default: 1)	
			REAL64	e.g. mm/s	±∞	Required velocity 1 (negative values also allowed)	
			REAL64	e.g. mm/s	±∞	Required velocity 2 (negative values also allowed)	
			REAL64	s	> 0.0	Travel time for velocity 1 and 2	
			REAL64	s	≥ 0.0	Idle time	
			UINT32	1	0, 1, 2, 3...	Optional: number of repetitions, Default "0": unlimited in time	
			}				
0x00000033	Write	SERVO	{			Sinus Oscillation Sequence - used as single sinus oscillation (sinus generator) - used as sinus oscillation sequence (e.g. for bode plot)	
			UINT32	ENUM	1	Start type (fixed to start type 1 yet)	
			REAL64	e.g. mm/s	> 0.0	Base amplitude (e.g. 2.5 mm/s)	
			REAL64	Hz	[0.0 .... 10.0]	Base frequency (e.g. 1.953125 Hz)	
			REAL64	e.g. mm/s	≥ 0.0	Start amplitude at begin (e.g. 0.0 mm/s)	
			REAL64	e.g. mm/REV	> 0.0	Feed constant motor (per motor turn) (e.g. 10.0 mm/REV)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	Hz	≥ 1.0	Frequency range: start frequency (e.g. 20.0 Hz)	
			REAL64	Hz	≤ 1/(2*dT)	Frequency range: stop frequency (e.g. 500.0 Hz)	
			REAL64	s	> 0.0	Step duration (e.g. 2.048s)	
			UINT32	1	[1 ... 200]	Number of measurements (step cycles) (e.g. 20)	
			UINT32	1		Number of parallel measurements (e.g. 1) not used yet!	
			}				
0x00000034	Write	SERVO	{			Phasing - Start Phasing - Stop Phasing	
			UINT32	ENUM	1	Phasing Type: 1: ABSOLUTE 2: RELATIVE 4096: STOP	
			UINT32	1	≥ 0	Control Mask	
			UINT32	1	≥ 0	Master axis ID (multi master)	
			UINT32			Reserve	
			REAL64	e.g. mm	> 0.0	Phase shift	
			REAL64	e.g. mm/s	> 0.0	Velocity	
			REAL64	e.g. mm/s^2	≥ 0.0	Acceleration	
			REAL64	e.g. mm/s^2	≥ 0.0	Deceleration	
			REAL64	e.g. mm/s^3	≥ 0.0	Jerk	
			REAL64[4]			Reserve	
			UINT32			Reserve	
			UINT32	1	ENUM	Buffer mode (NOT IMPLEMENTED)	
			REAL64	e.g. mm	±∞	Blending position (NOT IMPLEMENTED)	
			}				
0x00000040 (0x00n00040)	Write	Master/Slave coupling: (SERVO)	{			Master/slave coupling (SERVO):	Extension for "flying saw"!
			UINT32	ENUM	s. appendix	Slave type/coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis/ group	
			UINT32	1	[0...8]	Sub-index n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Sub-index n of the slave axis (default: value: 0)	
			REAL64	1	[±1000000.0]	Parameter 1:	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Linear: gearing factor FlySawVelo: Reserve FlySaw: abs. synchron position master [mm]	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve FlySawVelo: Reserve FlySawPos: abs. synchron position slave [mm]	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve FlySawVelo: angle of inclination in [DEGREE] FlySawPos: angle of inclination in [DEGREE]	Angle > 0.0 and £ 90.0 degree (parallel saw: 90.0 degree)
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve FlySawVelo: gearing factor FlySawPos: gearing factor	
			}				
0x00000041	Write	Master/slave decoupling (SERVO)	VOID			Master/slave decoupling (SERVO)	
0x00000041	Write	Master/slave decoupling with configurable follow-up function (SERVO)	{			Master/slave decoupling with configurable follow-up function (e.g. new end position, new velocity, stop, E-stop) (SERVO)	Not yet released!
			UINT32	ENUM	s. appendix [▶ 135]	Decoupling type (see appendix)	
			REAL64	e.g. mm	±∞	Optional: New end position	
			REAL64	e.g. mm/s	> 0.0	Optional: New requested velocity	
			REAL64	e.g. mm/s^2	≥ 0.0 (0: Default)	Optional: Acceleration for new end position, new velocity and emergency stop (E-stop)	
			REAL64	e.g. mm/s^2	≥ 0.0 (0: Default)	Optional: Deceleration for new end position, new veloc-	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						ity and emergency stop (E-stop)	
			REAL64	e.g. mm/s^3	≥ 0.0 (0: Default)	Optional: Jerk for new end position, new velocity and emergency stop (E-stop)	
			}				
0x00000042	Write	Master/Slave coupling Type: LINEAR (&SPECIAL)	{			Change of the coupling parameters (SERVO):	
			REAL64	1	[±1000000.0]	Parameter 1: Linear: gearing factor	
			REAL64	1	[±1000000.0]	Parameter 2: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 3: Linear: Reserve	
			REAL64	1	[±1000000.0]	Parameter 4: Linear: Reserve	
			}				
0x00000043	Write	Master/slave table coupling Type: TABULAR	{			Change of the table coupling parameters (SERVO):	
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			}				
0x00000043	Write	Master/slave table coupling Type: TABULAR and "Motion Function"	{			Change of the table coupling parameters (SERVO):	Also for "Motion Function"
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			REAL64	1	±∞ (<> 0.0)	Slave position scaling	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			}				
0x00000043	Write	Master/slave table coupling Type: TABULAR	{			Change of the table coupling parameters (SERVO):	
			REAL64	mm	±∞	Slave position offset	
			REAL64	mm	±∞	Master position offset	
			REAL64	1	±∞ (<> 0.0)	Slave position scaling	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			REAL64	e.g. mm	±∞	Absolute master activation position	
			}				
0x00000044	Write	Slave-Stop (SERVO)	VOID			Stop the "flying saw" (SERVO)	Only for "flying saw"

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00000045 (0x00n00045)	Write	Master/slave table coupling (SERVO)	{			Master/slave table coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/coupling type (s. appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
						SOLO TABLE SECTION	
			REAL64	mm	±∞	Slave position offset (type: TABULAR)	
			REAL64	mm	±∞	Master position offset (type: TABULAR)	
			UINT32	1	[0,1]	Slave position absolute (type: TABULAR)	
			UINT32	1	[0,1]	Master position absolute (type: TABULAR)	
			UINT32	1	[1...255]	Table ID of the coupling table (type: TABULAR)	
						MULTI-TABLE SECTION	
			UINT16	1	[0...8]	Number of tables (type: MULTITAB) Note: misused as interpolation type for solo tables	
			UNIT16	1	[0...8]	Number of profile tables (type: MULTITAB)	
			UNIT32[8]	1	[1...255]	Table IDs of the coupling tables (type: MULTITAB)	
			}				
0x00000046	Write	Master / slave multi-tables	UINT32	1	[1...255]	Activation of correction table correction table ID	
0x00000046	Write	Master / slave multi-tables	{			Activation of correction table	
			UINT32	1	[1...255]	Correction table ID	
			REAL64	e.g. mm	±∞	Absolute master activation position	
			}				
0x00000047	Write	Master/Slave Multi-tables	UINT32	1	[1..255]	Deactivation of profile table at end of cycle	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Table ID of the current mono-cyclic profile table	
0x00000048	ReadWrite	Master/Slave Multi-tables	Write: UINT32	1	[1..255]	Read the last correction offset:+ Table ID of the correction table	
			Read: REAL32	e.g. mm	±∞	Offset by departing the correction table with the according table ID	
0x00000049	Write	Master/slave table coupling Type: TABULAR	REAL64	1	±∞	Change the slave table scaling Scaling factor of the slave table column (Default value: 1.0)	
0x0000004A (0x00n0004A)	Write	Master/slave universal table coupling (SERVO)	{			Master/slave solo table coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/coupling type (see appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
			UINT32	1	1...255]	Table ID of the coupling table (type: TABULAR)	
			UINT32	1		Tabular interpolation type	
			REAL64	mm	±∞	Slave position offset (type: TABULAR)	
			REAL64	mm	±∞	Master position offset (type: TABULAR)	
			REAL64	mm	±∞	Slave position scaling (type: TABULAR)	
			REAL64	mm	±∞	Master position scaling (Type: TABULAR)	
			UINT32	1	[0,1]	Slave position absolute? (Type: TABULAR)	
			UINT32	1	[0,1]	Master position absolute? (Type: TABULAR)	
			UINT32	ENUM	s. appendix	Activation type of the change (NEW)	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'	
			REAL64	mm	±∞	Activation position (NEW)	
			UINT32	ENUM	s. appendix	Master scaling type (NEW) 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type (NEW) 0: user defined (default) 1: scaling with auto offset 2: off	
			}				
0x0000004B (0x00n0004B)	Write	Master / slave universal flying saw (SERVO)	{			Master/slave synchronizing coupling (SERVO):	
			UINT32	ENUM	s. appendix	Slave type/coupling type (s. appendix)	
			UINT32	1	[1...255]	Axis ID of the master axis	
			UINT32	1	[0...8]	Subindex n of the master axis (default: value: 0)	
			UINT32	1	[0...8]	Subindex n of the slave axis (default: value: 0)	
			REAL64	1	±∞ (<> 0.0)	Gearing factor	
			REAL64	mm	±∞	Master synchron position	
			REAL64	mm	±∞	Slave synchron position	
			REAL64	mm/s	≥ 0.0	Slave velocity (optional)	
			REAL64	mm/s^2	≥ 0.0	Slave acceleration (optional)	
			REAL64	mm/s^2	≥ 0.0	Slave deceleration (optional)	
			REAL64	mm/s^3	≥ 0.0	Slave jerk (optional)	
			UINT32	1	≥ 0	Bit mask for checks and operation modes (Default value: 0)	
			}				

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x0000004D (0x00n0004D)	Write	Master/slave table coupling Type: TABULAR and MF	{			Change of the table scaling (SERVO):	Also for MF Reserved function, no standard!
			UINT32	ENUM	s. appendix	Activation type of the change 0: 'instantaneous' (default) 1: 'at master cam position' 2: 'at master axis position' 3: 'next cycle'	
			REAL64	e.g. mm	±∞	Activation position	
			UINT32	ENUM	s. appendix	Master scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			REAL64	e.g. mm	±∞	Master position offset	
			REAL64	e.g. mm	±∞	Slave position offset	
			REAL64	1	±∞ (<> 0.0)	Master position scaling	
			REAL64	1	±∞	Slave position scaling	
			}				
0x00000050	Write	every	VOID			Deactivate complete axis (disable)	
0x00000051	Write	every	VOID			Activate complete axis (enable)	
0x00000052	Write	SERVO	{			Change of the active axis control loop (triple from encoder, controller and axis interfaces) with/without external set value setting:	
			UINT32	1	≥ 0	Number/index of the axis control loop (Default value: 0)	
			UINT32	ENUM	see appendix (>0)	Switch type for synchronization behavior 1: 'Standard'	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
			REAL64	1	±∞	Synchronization value for switching (optional)	
			UINT32	0/ 1	0/1	External set value setting by means of axis interface? Note: not used so far!	
			}				
0x00000060	Write	every	VOID			Deactivate drive output (disable)	
0x00000061	Write	every	VOID			Activate drive output (enable)	
0x00000062	Write	high/low	UINT16	1	0/1	Release parking brake? 0: automatic activation (default) 1: mandatorily always released! Note: reset to '0' when resetting the axis!	
0x00000070	Write	every	VOID			Return of the axis from, e.g. a 3D group to its own PTP group	

5.4.1.4.4 "Index offset" specification for cyclic axis process data (Index group 0x4300 + ID)

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note

0x00n00000	Read/Write	every (PLC→NC)	{
			128 bytes
			}
					STRUCT
					see axis interface
						AXIS STRUCTURE (PLC→NC)
						n = 0: standard axis interface
						n > 0: n-th axis interface (optional)
							Write command only optional!
							Consider safety aspects!

0x00n00001	Read/Write	every (PLC→NC)	UINT32	1	>0	Control double word	Write command only optional!
0x00n00002	Read/Write	every (PLC→NC)	UINT16	1	0/1	Enable controller	Cannot be traced by oscilloscope!
0x00n00003	Read/Write	every (PLC→NC)	UINT16	1	0/1	Feed enable plus	Cannot be traced by oscilloscope!
0x00n00004	Read/Write	every (PLC→NC)	UINT16	1	0/1	Feed enable minus	Cannot be traced by oscilloscope!
0x00n00007	Read/Write	every (PLC→NC)	UINT16	1	0/1	Referencing cam	Cannot be traced by oscilloscope!
0x00n00021	Read/Write	every (PLC→NC)	UINT32	%	0...1000000	Speed override (1000000 == 100%)	Write command only optional!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n00022	Read/Write	every (PLC→NC)	UINT32	1	ENUM	Operation mode axis	Write command only optional!
0x00n00025	Read/Write	every (PLC→NC)	REAL64	e.g. mm		Actual position correction value (measurement system error correction)	Write command only optional!
0x00n00026	Read/Write	every (PLC→NC)	REAL64	e.g. mm/s		External controller part (position controller part)	Write command only optional!
0x00n00027	Read/Write	every (PLC→NC)	{			External setpoint generation	Write command only optional!
			REAL64	e.g. mm	±∞	External set position	
			REAL64	e.g. mm/s	±∞	External set velocity	
			REAL64	e.g. mm/s^2	±∞	External set acceleration	
			INT32	1	+1, 0, -1	External set travel direction	
			}				

0x00n00080	Read	every (PLC→NC)	{
			128 bytes
			}
					STRUCT
					see axis interface
						AXIS STRUCTURE (NC→PLC)
						n = 0: standard axis interface
						n > 0: n-th axis interface (optional)
							NCTOPLC_AXLESTRUCT

0x00n00071	Read	every (PLC→NC)	UINT8	1	>0	Status double word: byte 1	
0x00n00072	Read	every (PLC→NC)	UINT8	1	>0	Status double word: byte 2	
0x00n00073	Read	every (PLC→NC)	UINT8	1	>0	Status double word: byte 3	
0x00n00074	Read	every (PLC→NC)	UINT8	1	>0	Status double word: byte 4	
0x00n00081	Read	every (PLC→NC)	UINT32	1	>0	Status double word (complete)	
0x00n00082	Read	every (PLC→NC)	UINT16	1	0/1	Axis is ready for operation	Cannot be traced by oscilloscope!
0x00n00083	Read	every (PLC→NC)	UINT16	1	0/1	Axis has been referenced	Cannot be traced by oscilloscope!
0x00n00084	Read	every (PLC→NC)	UINT16	1	0/1	Axis in protected operation mode (e.g. slave axis)	Cannot be traced by oscilloscope!
0x00n00085	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in rapid mode	Cannot be traced by oscilloscope!
0x00n00088	Read	every (PLC→NC)	UINT16	1	0/1	Axis has invalid IO data	Cannot be traced by oscilloscope!
0x00n00089	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in a fault state	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
0x00n0008A	Read	every (PLC→NC)	UINT16	1	0/1	Axis moving to larger values	Cannot be traced by oscilloscope!
0x00n0008B	Read	every (PLC→NC)	UINT16	1	0/1	Axis moving to smaller values	Cannot be traced by oscilloscope!
0x00n0008C	Read	every (PLC→NC)	UINT16	1	0/1	Axis is at a logical standstill (only set values are considered) (position controller?)	Cannot be traced by oscilloscope!
0x00n0008D	Read	every (PLC→NC)	UINT16	1	0/1	Axis is being referenced	Cannot be traced by oscilloscope!
0x00n0008E	Read	every (PLC→NC)	UINT16	1	0/1	Axis is in position window	Cannot be traced by oscilloscope!
0x00n0008F	Read	every (PLC→NC)	UINT16	1	0/1	Axis is at target position (target position reached)	Cannot be traced by oscilloscope!
0x00n00090	Read	every (PLC→NC)	UINT16	1	0/1	Axis has constant velocity or rotary speed	Cannot be traced by oscilloscope!
0x00n0009A	Read	every (PLC→NC)	UINT16	1	0/1	Operation mode not executed (busy)	Cannot be traced by oscilloscope!
0x00n0009B	Read	every (PLC→NC)	UINT16	1	0/1	Axis has instructions, is carrying instructions out	Cannot be traced by oscilloscope!
0x00n000B1	Read	every (PLC→NC)	UINT32	1	≥0	Axis error code	
0x00n000B2	Read	every (PLC→NC)	UINT32	1	ENUM	Present state of the axis movement	
0x00n000B3	Read	every (PLC→NC)	UINT32	1	ENUM	Axis operation mode (feedback from the NC)	
0x00n000B4	Read	every (PLC→NC)	UINT32	1	ENUM	Axis referencing status	
0x00n000B5	Read	every (PLC→NC)	UINT32	1	ENUM	Axis coupling status	
0x00n000B6	Read	every (PLC→NC)	UINT32	1	≥0	SVB entries/ tasks of the axis (PRE table)	
0x00n000B7	Read	every (PLC→NC)	UINT32	1	≥0	SAF entries/ tasks of the axis (EXE table)	
0x00n000B8	Read	every (PLC→NC)	UINT32	1	≥0	Axis ID	
0x00n000B9	Read	every (PLC→NC)	UINT32	1	≥0	Operation modes status double word: Bit 0: position range monitoring active? Bit 1: target position window monitoring active?	

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						Bit 2: loop travel active? Bit 3: physical movement monitoring active? Bit 4: PEH time monitoring active? Bit 5: backlash compensation active? Bit 6: NEW: delayed error reaction mode active? Bit 7: NEW: modulo operation mode active (modulo axis)? Bit 16: trailing separation monitoring position active? Bit 17: following error monitoring speed active? Bit 18: end position monitoring min. active? Bit 19: end position monitoring max. active? Bit 20: actual position correction active?	
0x00n000BA	Read	every (PLC→NC)	REAL64	e.g. mm		Actual position (calculated absolute value)	
0x00n000BB	Read	every (PLC→NC)	REAL64	e.g. mm		Modulo actual position	
0x00n000BC	Read	every (PLC→NC)	INT32	1		Modulo rotations	
0x00n000BD	Read	every (PLC→NC)	REAL64	e.g. mm/s		Actual velocity (optional)	
0x00n000BE	Read	every (PLC→NC)	REAL64	e.g. mm		Following error position	
0x00n000BF	Read	every (PLC→NC)	REAL64	e.g. mm		Set position	
0x00n000C0	Read	every (PLC→NC)	REAL64	e.g. mm/s		Set velocity	
0x00n000C1	Read	every (PLC→NC)	REAL64	e.g. mm/s^2		Set acceleration	
0x00n000C2	Read	every (PLC→NC)	REAL64			Reserve 2	
0x00n000C3	Read	every (PLC→NC)	REAL64			Reserve 3	
0x00n000C4	Read	every (PLC→NC)	REAL64			Reserve 4	

0x00n10000	Read/Write	Encoder: every (NC→IO)	{
			12 bytes
			}
					STRUCT
					s. encoder-interface
						ENCODER-OUTPUT-STRUCTURE	Write command only optional!
							Consider safety aspects!

Index offset (Hex)	Access	Axis type	Data type	Phys. unit	Definition range	Description	Note
						(NC→IO, 12 bytes)	

0x00n10000	Read/Write	Encoder: every (NC→IO)	{
			40 bytes
			}
					STRUCT
					s. new encoder-interface
						ENCODER-EXTENDED-OUTPUT-STRUCTURE (new)
						(NC→IO, 40 bytes)
							Write command only optional!
							Consider safety aspects!

0x00n10080	Read	Encoder: every (IO→NC)	{
			12 bytes
			}
					STRUCT
					s. encoder-interface
						ENCODER-INPUT-STRUCTURE
						(IO→NC, 12 bytes)	

0x00n10080	Read	Encoder: every (IO→NC)	{
			40 bytes
			}
					STRUCT
					s. new encoder-interface
						ENCODER-EXTENDED-INPUT-STRUCTURE (new)
						(IO→NC, 40 bytes)	

0x00n30000	Read/Write	Drive: every (NC→IO)	{
			12 bytes
			}
					STRUCT
					see drive interface
						DRIVE-OUTPUT-STRUCTURE
						(NC→IO, 12 bytes)
							Write command only optional!
							Consider safety aspects!

0x00n30000	Read/Write	Drive: every (NC→IO)	{
			40 bytes
			}
					STRUCT
					See new drive interface
						DRIVE-EXTENDED-OUTPUT-STRUCTURE (new)
						(NC→IO, 40 bytes)
							Write command only optional!
							Consider safety aspects!

0x00n30080	Read	Drive: every (IO→NC)	{
			12 bytes
			}
					STRUCT
					see drive interface
						DRIVE-INPUT-STRUCTURE
						(NC→IO, 12 bytes)	

0x00n30080	Read	Drive: every (IO→NC)	{
			40 bytes
			}
					STRUCT
					See new drive interface
						DRIVE-EXTENDED-INPUT-STRUCTURE (new)
						(NC→IO, 40 bytes)	

5.4.1.5 Specification Encoder

5.4.1.5.1 "Index offset" specification for encoder parameter (Index group 0x5000 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT32	1	[1 ... 255]	Encoder ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Encoder name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Encoder type [▶ 137]	
0x00000004	Read / Write	every	UINT32	1	Byte offset	Input address offset (IO-Input-Image)	Change I/O address
0x00000005	Read / Write	every	UINT32	1	Byte offset	Output address offset (IO-Output-Image)	Change I/O address

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000006	Read / Write	every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Resulting scaling factor (numerator / denominator) Note: from TC 3.0 the scaling factor consists of two components – numerator and denominator (default: 1.0).	
0x00000007	Read / Write	every	REAL64	e.g. mm	[±1.0E+9]	Position offset	
0x00000008	Read / Write	every	UINT16	1	[0,1]	Encoder count direction	
0x00000009	Read / Write	every	REAL64	e.g. mm	[0.001 ... 1.0E+9]	Modulo factor	
0x0000000A	Read / Write	every	UINT32	1	s. ENUM (>0)	Encoder mode [▶ 138]	s. appendix
0x0000000B	Read / Write	every	UINT16	1	0/1	Soft end min. monitoring?	
0x0000000C	Read / Write	every	UINT16	1	0/1	Soft end max. monitoring?	
0x0000000D	Read / Write	every	REAL64	mm		Soft end position min.	
0x0000000E	Read / Write	every	REAL64	mm		Soft end position max.	
0x0000000F	Read / Write	every	UINT32	1	s. ENUM (≥0)	Encoder evaluation direction [▶ 138] (enable for log. counting direction)	s. appendix
0x00000010	Read / Write	every	REAL64	s	[0.0...60.0]	Filter time for actual position value in seconds (P-T1)	
0x00000011	Read / Write	every	REAL64	s	[0.0...60.0]	Filter time for actual velocity value in seconds (P-T1)	
0x00000012	Read / Write	every	REAL64	s	[0.0...60.0]	Filter time for actual acceleration value in seconds (P-T1)	
0x00000013	Read / Write	every	UINT8[10+1]	1		Physical unit	Not implemented !
0x00000014	Read/Write	every	UINT32	1		Interpretation of the units (position, velocity, time) Bit 0: velocity in x/min instead of x/s Bit 1: position in thousands of the base unit	Not implemented! bit array
0x00000015	Read	every	UINT32	INC	[0x0... 0xFFFFFFFF]	Encoder mask (maximum value of the encoder actual value in increments) Note: From TwinCAT 2.11 R2 the encoder mask can be any numerical value (e.g.	Read-only parameter s. parameter "Encoder Sub Mask"

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
						3600000) and no longer needs to be a continuous sequence of binary ones (2^n-1).	
0x00000016	Read/Write	every	UINT16	1	0/1	Actual position correction (measurement system error correction)?	
0x00000017	Read/Write	every	REAL64	s	[0.0...60.0]	Filter time for actual position correction in seconds (P-T1)	
0x00000018	Read/Write	every	UINT32	1	[0x0... 0xFFFFFFFF]	Filter mask for raw incremental value (0x0: full passage)	
0x00000019	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder absolute dimensioning system [▶ 139]	s. appendix
0x0000001A	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder position initialization [▶ 139]	Not implemented!
0x0000001B	Read/Write	every	REAL64	e.g. mm	[≥0, modulo factor/2]	Tolerance window for modulo-start	
0x0000001C	Read	every	UINT32	1	s. ENUM (≥0)	Encoder sign interpretation (data type) [▶ 138]	
0x0000001D	Read	every	UINT16	1	0/1	Incremental or absolute encoder? 0: incremental encoder type 1: absolute encoder type	
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0)	Encoder dead time compensation mode 0: Off (default) 1: On (with velocity) 2: On (with velocity and acceleration)	
0x00000021	Read/Write	every	UINT32	1		Control double word (32 bits) for the encoder dead time compensation: Bit 0 = 0: relative IO times (default) Bit 0 = 1: absolute IO times	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Sum of the parameterized time shifts for the encoder dead time com-	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
						pensation (typically positive numerical values)	
0x00000023	Read/Write	every	REAL64	e.g. mm/INC	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: numerator (=> scaling factor numerator / scaling factor denominator)	From TC 3.0
0x00000024	Read/Write	every	REAL64	1	[1.0E-12 ... 1.0E+30]	Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator) Default: 1.0	From TC 3.0
0x00000025	Read/Write	every	{ REAL64 REAL64 } 16 bytes	e.g. mm/INC 1	[1.0E-12 ... 1.0E+30] [1.0E-12 ... 1.0E+30]	Component of the scaling factor: numerator Component of the scaling factor: denominator (=> scaling factor numerator / scaling factor denominator)	From TC 3.0
0x00000030	Read/Write	every	UINT32	1		Internal encoder control double word for specifying the operating modes and properties	From 211R3 B2227
0x00000101	Read/Write	INC	UINT16	1	[0,1]	Inverse search direction for ref.cam?	
0x00000102	Read/Write	INC		1	[0,1]	Inverse search direction for sync pulse?	
0x00000103	Read/Write	INC	REAL64	e.g. mm	[±1.0E+9]	Reference position	
0x00000104	Read/Write	INC	UINT16	1	[0,1]	Distance monitoring between Ref. cams and sync pulse active?	Not implemented!
0x00000105	Read/Write	INC	UINT32	INC	[0 ...65536]	Minimum distance between Ref. cams and sync pulse in increments	Not implemented!
0x00000106	Read/Write	INC	UINT16	1	[0,1]	External sync pulse?	
0x00000107	Read/Write	INC	UINT32	1	s. ENUM (>0)	Reference mode [▶ 139]	s. appendix
0x00000108	Read/Write	INC	UINT32	1	[0x0000000F... 0xFFFFFFFF] Binary mask: (2^n - 1)	Encoder sub mask (maximum value of the absolute range of the encoder actual value in increments)	s. parameter "Encoder Mask"

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
						Used, for example, as a reference mark for the referencing mode "Software Sync" and for the NC Retain Data "ABSOLUTE (MODULO)", "INCREMENTAL (SINGLETURN ABSOLUTE)". Note 1: The Encoder Sub Mask must be smaller than or equal to the Encoder Mask. Note 2: The Encoder Mask must be an integer multiple of the Encoder Sub Mask. Note 3: The Encoder Sub Mask must be a continuous sequence of binary ones (2^n-1), e.g. 0x000FFFFF.	
0x00000110	Read/Write	INC (encoder simulation)	REAL64	1	[0.0 ... 1000000.0]	Scaling/weight of the noise part for the simulation encoder	

5.4.1.5.2 "Index offset" specification for encoder state (Index group 0x5100 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	INT32			Error state encoder	
0x00000002	Read	every	REAL64			Actual position (charge with actual position compensation value)	Symbolic access possible! *'fPosIst'*
0x00000003	Read	every	REAL64			Modulo actual position	Symbolic access possible! *'FModuloPosIst'*
0x00000004	Read	every	INT32			Modulo actual rotation	Symbolic access possible! *'nModuloTurns'*
0x00000005	Read	every	REAL64			Optional: actual velocity	Base Unit / s *symbolic access possible!* *'fVeloIst'*
0x00000006	Read	every	REAL64			Optional: actual acceleration	Base Unit / s^2 *symbolic access possible!*

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
							'fAccIst'
0x00000007	Read	every	INT32			Encoder actual increments	Symbolic access possible! *'nHardIncs'*
0x00000008	Read	every	INT64			Software - actual increment counter	Symbolic access possible! *'nSoftIncs'*
0x00000009	Read/Write	every	UINT16			Reference flag ("calibrate flag")	
0x0000000A	Read	every	REAL64			Actual position correction value (measurement system error correction)	
0x0000000B	Read	every	REAL64			Actual position without actual position compensation value	
0x0000000C	Read	every	REAL64	e.g. mm		Actual position compensation value due to the dead time compensation	
0x0000000D	Read	every	REAL64	s		Sum of the time shifts for encoder dead time compensation (parameterized and variable dead time) Note: a dead time is specified in the system as a positive value.	
0x0000000E	Read	every	REAL64	e.g. mm		Internal position offset as a correction value for a value reduction to the base period (modulo range)	
0x00000010	Read	every	REAL64	e.g. mm/s		Actual velocity without actual position compensation value	
0x00000012	Read	every	REAL64	e.g. mm		Unfiltered actual position (charge with actual position compensation value)	
0x00000015	Read	every	REAL64	e.g. mm/s		Optional: Unfiltered actual velocity	Base Unit / s
0x00000016	Read	every	**READ** (16 byte * N)			Read the actual position buffer	From TC 2.11 R3
			{				
			UINT32	ns	≥0	DcTimeStamp with 32 bits	
			UINT32			Reserve	
			REAL64	e.g. mm	±∞	Actual position for the associated DC time stamp	
			} [N]				

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000101	Read	INC	REAL64	e.g. mm		Read back the position difference between the hardware latch being activated and becoming valid	Cannot be traced by oscilloscope!
0x00000200	**Read Write**	**Function group "TouchProbeV2":** **- SERCOS (SoE),** **- EtherCAT (CANopen DS402)** **- SoftDrive (TCom)**	**WRITE** (24 byte)			Read "Touch Probe" state (state of external latch)	TC 2.11 Build 1547 only for SAFport 501
			{				
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32[5]			Reserved	
			}				
			READ (64 byte)				
			{				
			UINT32	1	[0/1]	Touch probe rising edge active?	
			UINT32	1	[0/1]	Touch probe rising edge became valid?	
			REAL64	e.g. mm		Touch probe rising edge position value	
			UINT32	1	≥0	Touch probe rising edge counter (continuous mode)	
			UINT32			Reserved	
			UINT32	1	[0/1]	Touch probe falling edge active?	
			UINT32	1	[0/1]	Touch probe falling edge became valid?	
			REAL64	e.g. mm		Touch probe falling edge position value	
			UINT32	1	≥0	Touch probe falling edge counter (continuous mode)	
			UINT32[5]			Reserved	
			}				
0x00000201	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" active? or "Measuring sensor function" active ? *(edge-independent)*	Cannot be traced by oscilloscope!
0x00000201	Read	**CANopen**	UINT32[4]	1	[0,1]	"External latch functions 1 to 4" active ? or Measuring sensor functions 1 to 4" active ?	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000202	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT16	1	[0,1]	External latch value became valid? or measuring sensor latched ? *(edge-independent)*	cf. Axis interface NcToPlc (state double word)
0x00000202	Read	**CANopen**	UINT32[4]	1	[0,1]	External latch values 1 to 4 became valid? or measuring sensors 1 to 4 latched ?	cf. Axis interface NcToPlc (state double word)
0x00000203	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT32	INC		External / measuring sensor hardware incremental latch value	
0x00000204	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT64	INC		External / measuring sensor software incremental latch value	
0x00000205	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	REAL64	e.g. mm		External / measuring sensor position latch value	Base Unit
0x00000205	Read	**CANopen**	REAL64 [4]	e.g. mm		External measuring sensor values / position latch values	Base Unit
0x00000206	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT32	INC		Difference hardware incremental latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope!
0x00000207	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	UINT64	INC		Difference software incremental latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope!
0x00000208	Read	**KL5101, SERCOS, AX2xxx, ProviDrive**	REAL64	e.g. mm		Difference software incremental latch values (NewLatch - LastLatch)	Cannot be traced by oscilloscope! Base Unit
0x00000210	Read	**KL5101, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" for *rising edge* active ? or "Measuring sensor function" active for *rising edge* ?	**Extension for KL5101 (3E), AX2xxx (3.51) and ProviDrive (3.1)** Cannot be traced by oscilloscope!
0x00000210	Read	**CANopen**	UINT16[4]	1	[0,1]	"External latch function" for *rising edge* active ? or "Measuring sensor function" active for *rising edge* ?	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000211	Read	**KL5101, AX2xxx, ProviDrive**	UINT16	1	[0,1]	"External latch function" for *falling edge* active ? or "Measuring sensor function" active for *falling edge* ?	**Extension for KL5101 (3E), AX2xxx (3.51) and ProviDrive (3.1)** Cannot be traced by oscilloscope!
0x00000211	Read	**CANopen**	UINT16[4]	1	[0,1]	"External latch function" for *falling edge* active ? or "Measuring sensor function" active for *falling edge* ?	Cannot be traced by oscilloscope!

5.4.1.5.3 "Index offset" specification for encoder functions (Index group 0x5200 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x0000001A	Write	every	{			Set actual position encoder/ axis	Base Unit
			UINT32	ENUM	s. appendix	Actual position type (s. appendix)	
			REAL64	mm	±∞	Actual position for encoder/axis **caution by using!**	
			}				
0x0000001B	Write	every	void			Re-initialization of the actual encoder position Note: Only takes effect for reference system “ABSOLUTE (with single overflow)”	From TC 2.11R3 Build 2261
0x00000200	Write	Function group "TouchProbeV2": - SERCOS (SoE), - EtherCAT (CANopen DS402) - SoftDrive (TCom)	{			Activate "Touch Probe" (external latch)	From TC 2.11 Build 154 Only for SAFport 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32	1	[1,2]	Probe mode (1=single, 2=continuous, ...)	
			UINT32			Reserved (e.g. signal source)	
			UINT32			Reserved	
			UINT32			Reserved	
			} 24 bytes				

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000201	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Activate "External latch" or Activate "Measuring sensor function" *(typically rising edge)*	
0x00000201	Write	CANopen	UINT32[4]			Activate "External Latch" 1 to 4 or Activate "Measuring sensor function" 1 to 4 *(typically rising edge)*	
0x00000202	Write	KL5101, AX2xxx, PROFIDrive	VOID			Activate "External Latch" or Activate "Measuring sensor function" *(falling edge)*	
0x00000202	Write	CANopen	UINT32[4]			Activate "External Latch" 1 to 4 or Activate "Measuring sensor function" 1 to 4 *(falling edge)*	
0x00000205	Write	Function group "TouchProbeV2": - SERCOS (SoE), - EtherCAT (CANopen DS402) - SoftDrive (TCom)	{			Deactivate "Touch Probe" (external latch)	From TC 2.11 Build 1547 only for SAFport 501
			UINT32	1	[1,2,3,4]	Probe unit (probe 1, 2, 3, 4)	
			UINT32	1	[0,1]	Signal edge (0=rising edge, 1=falling edge)	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
			UINT32			Reserved	
			} 24 byte				
0x00000205	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	VOID			Deactivate "External Latch" or Deactivate "Measuring sensor function"	
0x00000205	Write	CANopen	UINT32[4]			Deactivate "External Latch" or Deactivate "Measuring sensor function"	
0x00000210	Write	KL5101, SERCOS, AX2xxx, PROFIDrive	REAL64	e.g. mm	±∞	Set "External latch event" and "External latch position"	Only for EtherCAT:

5.4.1.5.4 "Index offset" specification for cyclic encoder process data (Index group 0x5300 + ID)

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000000	Read/Write	every (NC→IO)	{		STRUCT see encoder interface or see extended encoder interface	ENCODER-OUTPUT-STRUCTURE (NC→IO, 12 bytes) or ENCODER-OUTPUT-STRUCTURE (NC→IO, 40 bytes)	Write command only optional! Consider safety aspects!
			INT32	INC	≥ 0	nOutData1	
			INT32	INC	≥ 0	nOutData2	
			UINT8	1	≥ 0	nControl1	
			UINT8	1	≥ 0	nControl2	
			UINT8	1	≥ 0	nControl3	
			UINT8	1	≥ 0	nControl4	
			Optional:				
			INT32	INC	≥ 0	nOutData3	
			INT32	INC	≥ 0	nOutData4	
			INT32	INC	≥ 0	nOutData5	
			INT32	INC	≥ 0	nOutData6	
			UINT8	1	≥ 0	nControl5	
			UINT8	1	≥ 0	nControl6	
			UINT8	1	≥ 0	nControl7	
			UINT8	1	≥ 0	nControl8	
			INT32		≥ 0	reserved	
			INT32		≥ 0	reserved	
			}				
0x00000001	Write	every (IO→NC)	{		STRUCT s. encoder interface	Bitwise access to ENCODER-OUTPUT-STRUCT (NC→IO, 40 Byte) NCENC-STRUCT_OUT 2	Write command only optional! Consider safety aspects! From TC2.11 R3 B2303
			UINT32	1	[0 … 39]	ByteOffset Relative address offset [0..39] in output structure. E.G.: To write “nControl1” the ByteOffset must be 8.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	BitSelectMask (BSM) The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	Value	

Index offset (Hex)	Access	Group type	Data type	Phys. unit	Definition range	Description	Remarks
						Only those bits in value are overwritten where BSM equals 1.	
			}				
0x00000080	Write	every (IO→NC)	{		STRUCT see encoder interface or see extended encoder interface	ENCODER-INPUT-STRUCTURE (IO→NC, 12 bytes) or optional ENCODER-INPUT-STRUCTURE (IO→NC, 40 bytes)	
			INT32	INC	≥ 0	nInData1	
			INT32	INC	≥ 0	nInData2	
			UINT8	1	≥ 0	nStatus1	
			UINT8	1	≥ 0	nStatus2	
			UINT8	1	≥ 0	nStatus3	
			UINT8	1	≥ 0	nStatus4	
			Optional:				
			INT32	INC	≥ 0	nInData3	
			INT32	INC	≥ 0	nInData4	
			INT32	INC	≥ 0	nInData5	
			INT32	INC	≥ 0	nInData6	
			UINT8	1	≥ 0	nStatus5	
			UINT8	1	≥ 0	nStatus6	
			UINT8	1	≥ 0	nStatus7	
			UINT8	1	≥ 0	nStatus8	
			INT32		≥ 0	Reserved	
			INT32		≥ 0	Reserved	
			}				

5.4.1.6 Specification Controller

5.4.1.6.1 "Index offset" specification for controller parameter (Index group 0x6000 + ID)

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT32	1	[1 ... 255]	Controller ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Controller name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Controller type	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Controller mode	
0x0000000B	Read/Write	every	REAL64	%	[0.0 ... 1.0]	Weight of the velocity pre-control (standard value: 1.0 == 100%)	
0x00000010	Read/Write	every	UINT16	1	0/1	Following error monitoring position?	
0x00000011	Read/Write	every	UINT16	1	0/1	Following error monitoring velocity?	
0x00000012	Read/Write	every	REAL64	mm		Max. following error position	

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000013	Read/Write	every	REAL64	s		Max. following error time position	
0x00000014	Read/Write	every	REAL64	mm/s		Max. following error velocity	
0x00000015	Read/Write	every	REAL64	s		Max. following error time velocity	
0x00000021	Read/Write	every	REAL64	1	[0.0...1000000.0]	Scaling factor (multiplier) for the difference between the following error of the master and that of the slave (Conversion in the same coordinate system of the master)	Reserved function, no standard!
0x00000100	Read/Write	P/PID (Pos., (velocity)	REAL64	1	[0.0...1.0]	Maximum output limitation (±) for controller total output	(Standard value: 0.5 == 50%)
0x00000102	Read/Write	P/PID (Pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Proportional gain kp or kv respectively	base unit / s / base unit Position control
0x00000103	Read/Write	PID (Pos.)	REAL64	s	[0.0 ... 60.0]	Integral action time T_n	Position control
0x00000104	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Derivative action time T_v	Position control
0x00000105	Read/Write	PID (pos.)	REAL64	s	[0.0 ... 60.0]	Damping time T_d	Position control
0x00000106	Read/Write	PP (Pos.)	REAL64	(mm/s) / mm	[0.0...1000.0]	Additional proportional gain, k_p or k_v respectively, that applies above a limit velocity in percent.	base unit / s / base unit Position control
0x00000107	Read/Write	PP (Pos.)	REAL64	%	[0.0...1.0]	Threshold velocity in percent above which the additional proportional gain, k_p or k_v respectively, applies	(Standard value: 0.01 == 1%)
0x00000108	Read/Write	P/PID (Acc.)	REAL64	s	[0.0 ... 100.0]	Proportional gain k_a	Acceleration pre control
0x0000010A	Read/Write	every	UINT32	1	ENUM	Filter for maximum increase in the set speed (acceleration-limited): 0: Off, 1: Velo, 2: Pos+Velo	Reserved function, no standard!
0x0000010B	Read/Write	every	REAL64	mm/s^2		Filter for maximum increase in the set speed (maximum acceleration)	Reserved function, no standard!
0x0000010D	Read/Write	P/PID	REAL64	mm	[0.0 ... 10000.0]	"dead band" for position error (control deviation)	Reserved function

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
						(for P/PID controllers with velocity or torque interface)	
0x0000010F	Read/Write	P/PP/PID (Pos.) slave control	REAL64	(mm/s) / mm	[0.0...1000.0]	Slave coupling difference control: proportional gain k_{cp}	Slave coupling difference control
0x00000110	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: active/passive	
0x00000111	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: hold mode	
0x00000112	Read/Write	P (Pos.)	UINT16	1	0/1	Automatic offset calibration: fading mode	
0x00000114	Read/Write	P (Pos.)	REAL64	%	[0.0 ... 1.0]	Automatic offset calibration: pre-control limit	(Standard value: 0.05 == 5%)
0x00000115	Read/Write	P (Pos.)	REAL64	s	[0.1 ... 60.0]	Automatic offset calibration: time constant	
0x00000116	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I-part in percent (default setting: 0.1 == 10%)	
0x00000117	Read/Write	PID (Pos.)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D-part in percent (default setting: 0.1 == 10%)	
0x00000118	Read/Write	PID (Pos.)	UINT16	1	0/1	Deactivation of the I-component during an active positioning process (assuming I-component active)? (default setting: 0 = FALSE)	
0x00000120	Read/Write	PID (Pos.)	REAL64	s	≥0	PT-1 filter value for position error (pos. control difference)	Reserved function, no standard!
0x00000202	Read/Write	P/PID (velocity)	REAL64	1	[0.0...1000.0]	Proportional gain k_p or k_v respectively	Velocity control
0x00000203	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Integral action time T_n	Velocity control
0x00000204	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Derivative action time T_v	Velocity control
0x00000205	Read/Write	PID (velocity)	REAL64	s	[0.0 ... 60.0]	Damping time T_d	Velocity control
0x00000206	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for I-part in percent (default setting: 0.1 == 10%)	Velocity control

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000207	Read/Write	PID (velocity)	REAL64	%	[0.0...1.0]	Maximum output limitation (±) for D-part in percent (default setting: 0.1 == 10%)	Velocity control
0x0000020D	Read/Write	P/PID (velocity)	REAL64	mm/s	[0.0 ... 10000.0]	"dead band" for velocity error (control deviation) (for P/PID controllers with velocity or torque interface)	Reserved function
0x00000220	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-2 filter value for velocity error (vel. control difference)	Velocity control, no standard!
0x00000221	Read/Write	P/PID (velocity)	REAL64	s	≥0	PT-1 filter value for velocity error (vel. control difference)	Reserved function, no standard!
0x00000250	Read/Write	P/PI (observer)	UINT32	1	s. ENUM (≥0)	Observer mode [▶ 136] for controller with torque interface 0: OFF (default) 1: LUENBERGER	From TC 2.10 Build 1320
0x00000251	Read/Write	P/PI (observer)	REAL64	Nm / A	>0.0	Motor: Torque constant K_T	
0x00000252	Read/Write	P/PI (observer)	REAL64	kg m^2	>0.0	Motor: Moment of inertia J_M	
0x00000253	Read/Write	P/PI (observer)	REAL64	Hz	[100.0 ... 2000.0] Default: 500	Bandwidth f_0	
0x00000254	Read/Write	P/PI (observer)	REAL64	1	[0.0 ... 2.0] Default: 1.0	Correction factor k_c	
0x00000255	Read/Write	P/PI (observer)	REAL64	s	[0.0 ... 0.01] Default: 0.001	Velocity filter (1^{st} order): Time constant T	
0x00000A03	Read/Write	PID (MW)	REAL64	cm^2	[0.0 ...1000000]	Cylinder area A_A of the A side in cm^2	
0x00000A04	Read/Write	PID (MW)	REAL64	cm^2	[0.0 ...1000000]	Cylinder area A_B of the B side in cm^2	
0x00000A05	Read/Write	PID (MW)	REAL64	cm^3/s	[0.0 ...1000000]	Nominal volume flow Q_{nom} in cm^3/s	
0x00000A06	Read/Write	PID (MW)	REAL64	bar	[0.0 ...1000000]	Rated pressure or valve pressure drop, P_{nom} in bar	
0x00000A07	Read/Write	PID (MW)	UINT32	1	[1 ... 255]	Axis ID for the system pressure P_o	

5.4.1.6.2 "Index offset" specification for controller state (Index group 0x6100 + ID)

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	INT32			Error status controller	Symbolic access possible! *'nErrState'*
0x00000002	Read	every	REAL64	e.g. mm/s		Controller output in absolute units	Base Unit / s *symbolic access possible! 'fOutput'*
0x00000003	Read	every	REAL64	%		Controller output in percent	Cannot be traced by oscilloscope!
0x00000004	Read	every	REAL64	V		Controller output in volts	Cannot be traced by oscilloscope!
0x0000000D	Read	every	REAL64	mm		Following error position (without dead time compensation)	Base Unit
0x0000000E	Read	every	REAL64	mm		Following error position (without set position correction)	Base Unit
0x0000000F	Read	every	REAL64	mm		Following error position (with set position correction and with dead time compensation)	Base Unit *symbolic access possible! 'fPosDiff'*
0x00000010	Read	every	REAL64	mm		Peak hold value for maximum negative following error of the position	Base Unit
0x00000011	Read	every	REAL64	mm		Peak hold value for minimum positive following error of the position	Base Unit
0x00000012	Read	every	REAL64	mm/s		Following error velocity	Base Unit / s
0x00000021	Read	every	REAL64	mm		Difference (deviation) between the following error from master and slave axis (master error minus slave error)	Base Unit *Symbolic access possible via axis! 'fPosDiffCouple'*
0x00000022	Read	every	REAL64	mm		PeakHold value for the maximum negative difference between master and slave following error of the position	Base Unit
0x00000023	Read	every	REAL64	mm		PeakHold value for the maximum positive	Base Unit

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
						difference between master and slave following error of the position	
0x00000101	Read	P/PID (pos.)	REAL64	e.g. mm/s		P-part of the controller in absolute units	
0x00000102	Read	PID (pos.)	REAL64	e.g. mm/s		I-part of the controller in absolute units	
0x00000103	Read	PID (pos.)	REAL64	e.g. mm/s		D-part of the controller in absolute units	
0x00000104	Read	PID (pos.)	UINT16	1	0/1	Limitation of the I-part active?	
0x00000105	Read	PID (pos.)	UINT16	1	0/1	Limitation of the D-part active?	
0x00000106	Read	PID (pos.)	UINT16	1	0/1	ARW measure for the I-part active?	ARW: Anti Reset Windup
0x00000110	Read	PID (pos.)	REAL64	e.g. mm/s		Acceleration pre-control Y_{acc} of the controller in absolute units Note: Function depends on controller type!	acceleration pre control
0x00000111	Read	PP (Pos.)	REAL64	mm/s/ mm	≥0	Internal interpolated proportional gain kp or kv	PP controller
0x0000011A 0x0000011B 0x0000011C 0x0000011D 0x0000011E 0x0000011F 0x00000120 0x00000121 0x00000122 0x00000123 0x00000124	Read	P (Pos.)	UINT32 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64 REAL64	1 mm mm/s mm/s mm/s^2 mm mm mm/s mm/s^2 mm/s mm/s^2		Set velocity filter: InternalPhase InternalPosSollError! TestVeloSoll InternalLimitedVeloSoll InternalAccSollRel InternalPosSollRel PosSollCorrected! VeloSollCorrected! AccSollCorrected! TestVeloSollCorrected TestAccSollCorrected	Reserved function, no standard!
0x00000201	Read	P,PID (velocity)	REAL64	e.g. mm/s		Velocity part of the controller	Base Unit / s
0x00000202	Read	P,PID (velocity)	REAL64	%		Velocity part of the controller in percent	Cannot be traced by oscilloscope!
0x00000203	Read	P,PID (velocity)	REAL64	V		Velocity part of the controller in volts	Cannot be traced by oscilloscope!

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000201	Read	P/PID (velocity)	REAL64	e.g. mm/s		P-part of the controller in absolute units	
0x00000202	Read	P/PID (velocity)	REAL64	e.g. mm/s		I-part of the controller in absolute units	
0x00000203	Read	P/PID (velocity)	REAL64	e.g. mm/s		D-part of the controller in absolute units	
0x00000204	Read	P/PID (velocity)	UINT16	1	0/1	Limitation of the I-part active?	
0x00000205	Read	P/PID (velocity)	UINT16	1	0/1	Limitation of the D-part active?	
0x00000206	Read	P/PID (velocity)	UINT16	1	0/1	ARW measure for the I-part active?	ARW: Anti Reset Windup
0x0000020A	Read	P/PID (velocity)	REAL64	e.g. mm/s		Total input size of the velocity controller	
0x00000250	Read	P/PI (observer)	REAL64	e.g. mm		Observer: position difference (actual position – observer position)	
0x00000251	Read	P/PI (observer)	REAL64	e.g. mm		Observer: position	
0x00000252	Read	P/PI (observer)	REAL64	e.g. mm/s		Observer: velocity 2 (for P-part)	
0x00000253	Read	P/PI (observer)	REAL64	e.g. mm/s		Observer: velocity 1 (for I-part)	
0x00000254	Read	P/PI (observer)	REAL64	e.g. mm/s^2		Observer: acceleration	
0x00000255	Read	P/PI (observer)	REAL64	A		Observer: actual motor current	
0x00000256	Read	P/PI (observer)	UINT16	1	0/1	Observer: Limitation of the I-part active?	
0x00000A00	Read	PID (MW)	REAL64	%	[-1.0...1.0]	Calculation of the set speed (pilot control) in percent	
0x00000A01	Read	PID (MW)	REAL64	e.g. mm/s		P-part of the controller in absolute units or percent (according to output weight)	
0x00000A02	Read	PID (MW)	REAL64	e.g. mm/s		I-part of the controller in absolute units or percent (according to output weight)	
0x00000A03	Read	PID (MW)	REAL64	e.g. mm/s		D-part of the controller in absolute units or percent (according to output weight)	
0x00000A04	Read	PID (MW)	UINT16	1	0/1	Limitation of the I-part active?	
0x00000A05	Read	PID (MW)	UINT16	1	0/1	Limitation of the D-part active?	

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000A10	Read	PID (pos.)	REAL64	e.g. mm/s		Acceleration pre-control Y_{acc} of the controller in absolute units	acceleration pre control

5.4.1.6.3 "Index offset" specification for controller functions (Index group 0x6200 + ID)

Index offset (Hex)	Access	Controller type	Data type	Phys. unit	Definition range	Description	Remarks

5.4.1.7 Specification Drive

5.4.1.7.1 "Index offset" specification for drive parameter (Index group 0x7000 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	UINT32	1	[1 ... 255]	Drive ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Drive name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Drive type [▶ 140]	
0x00000004	Read/Write	every	UINT32	1	Byte offset	Input address offset (IO-Input-Image)	Change I/O address
0x00000005	Read/Write	every	UINT32	1	Byte offset	Output address offset (IO-Output-Image)	Change I/O address
0x00000006	Read/Write	every	UINT16	1	[0,1]	Motor polarity	
0x0000000A	Read/Write	every	UINT32	1	s. ENUM (>0)	Drive mode	
0x0000000B	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Minimum output limit (output limitation) (default setting: -1.0 == -100%)	
0x0000000C	Read/Write	every	REAL64	%	[-1.0 ... 1.0]	Maximum output limit (output limitation) (default setting: 1.0 == 100%)	
0x0000000D	Read	every	UINT32	INC		Maximum number of output increments (output mask)	
0x00000010	Read/Write	every	UINT32	1		Internal Drive Control double word to determine the drive operation modes	Reserved!
0x00000011	Read/Write	every	UINT32	1	≥ 5	Internal Drive Reset Counter (time in NC cycles for enable and reset)	Reserved!

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000020	Read/Write	every	UINT32	1	s. ENUM (≥0)	Drive dead time compensation mode 0: Off (default) 1: On (with velocity) 2: On (with velocity and acceleration)	s. appendix
0x00000021	Read/Write	every	UINT32	1		Control double word (32 bits) for the drive dead time compensation: Bit 0 = 0: relative IO times (default) Bit 0 = 1: absolute IO times	
0x00000022	Read/Write	every	INT32	ns	[±1.0E+9]	Sum of the parameterized time shifts for the drive dead time compensation (typically positive numerical values)	
0x00000101	Read/Write	Servo	REAL64	e.g. mm/s	>0.0	Reference velocity at reference output (velocity precontrol)	Base Unit / s
0x00000102	Read/Write	Servo	REAL64	%	[0.0 ... 5.0]	Reference output in percent	
0x00000103	Read	Servo	REAL64	e.g. mm/s	>0.0	Resulting velocity at 100% output	Base Unit / s
0x00000104	Read/Write	Servo	REAL64	e.g. mm/s	±∞	Velocity offset (DAC offset) for drift calibration (offset calibration) of the axis	Base Unit / s
0x00000105	Read/Write	Servo (Sercos, Profi Drive, AX200x, CANopen)	REAL64	1	[0.0 ... 100000000.0]	Velocity scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x00000106	Read/Write	Profi Drive DSC	UINT32	0.001 * 1/s	≥ 0	Profibus/Profi Drive DSC: position control gain Kpc	Only for Profi Drive DSC
0x00000107	Read/Write	Profi Drive DSC	REAL64	1	≥ 0.0	Profibus/Profi Drive DSC: scaling for calculating 'XERR' (Default: 1.0)	Only for Profi Drive DSC
0x00000109	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Position scaling (scaling factor to react to the weight in the drive)	For Sercos, CANopen
0x0000010A	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Acceleration scaling (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x0000010B	Read/Write	Servo	REAL64	1	[0.0 ... 100000000.0]	Torque scaling (rotary motor) or force scaling (linear motor) (scaling factor to react to the weight in the drive)	For Sercos, Profi Drive, AX200x, CANopen
0x0000010D	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive velocity output	For Sercos, CANopen
0x0000010E	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive acceleration output	For Sercos, CANopen
0x0000010F	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Damping time for drive torque output or force output	For Sercos, CANopen
0x00000110	Read/Write	Servo (Sercos, CANopen)	UINT32	1	s. ENUM (≥0)	Optional output filtering of the set position value: Filter type 0: OFF (Default) 1: Moving Average 2: P-Tn	For Sercos, CANopen
0x00000111	Read/Write	Servo (Sercos, CANopen)	REAL64	s	[0.0 ... 1.0]	Optional output filtering of the set position value: Filter time remark: the maximum filter time depends on the NC cycle time and is limited to the following maximum value: 1 ms => 64 ms 2 ms => 128 ms 3 ms => 192 ms	For Sercos, CANopen
0x00000112	Read/Write	Servo (Sercos, CANopen)	UINT32	1	[0 ... 10]	Optional output filtering of the set position value: Filter order 'n' (for P-Tn type only)	For Sercos, CANopen
0x00000120	Read/Write	Servo/Hydraulic/	UINT32	1	≥ 0	Table ID (0: no table)	Only for KL4xxx, M2400, Universal
0x00000121	Read/Write	Servo/Hydraulic	UINT32	1	≥ 0	Interpolation type 0: Linear 2: Spline	Only for KL4xxx, M2400, Universal
0x00000122	Read/Write	Servo/Hydraulic	REAL64	%	[-1.0 ... 1.0]	Output offset in percent Note:	Only for KL4xxx, M2400, Universal

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
						Acts according to the characteristic evaluation!	
0x00000151	Read/Write	Servo/non-linear	REAL64	1	[0.0 ... 100.0]	Quadrant equalizing factor (relation between quadrants I and III)	
0x00000152	Read/Write	Servo/non-linear	REAL64	1	[0.01 ... 1.0]	Velocity reference point in percent (1.0 == 100 %)	
0x00000153	Read/Write	Servo/non-linear	REAL64	1	[0.01 ... 1.0]	Output reference point in percent (1.0 == 100%)	
0x00000301	Read/Write	Stepper motor	UINT8			Bit mask: cycle 1	
0x00000302	Read/Write	Stepper motor	UINT8			Bit mask: cycle 2	
0x00000303	Read/Write	Stepper motor	UINT8			Bit mask: cycle 3	
0x00000304	Read/Write	Stepper motor	UINT8			Bit mask: cycle 4	
0x00000305	Read/Write	Stepper motor	UINT8			Bit mask: cycle 5	
0x00000306	Read/Write	Stepper motor	UINT8			Bit mask: cycle 6	
0x00000307	Read/Write	Stepper motor	UINT8			Bit mask: cycle 7	
0x00000308	Read/Write	Stepper motor	UINT8			Bit mask: cycle 8	
0x00000310	Read/Write	Stepper motor	UINT8			Bit mask: holding current	

5.4.1.7.2 "Index offset" specification for drive state (Index group 0x7100 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000001	Read	every	INT32			Error state drive	Symbolic access possible! *'nErrState'*
0x00000002	Read	every	REAL64	e.g. mm/s		Total output in absolute units	Base Unit / s *symbolic access possible!* *'fOutput'*
0x00000003	Read	every	REAL64	%		Total output in percent	
0x00000004	Read	every	REAL64	V		Total output in volts	Cannot be traced by oscilloscope!
0x00000005	Read	every	REAL64	e.g. mm/s		PeakHold value for maximum negative total output	Base Unit / s
0x00000006	Read	every	REAL64	e.g. mm/s		PeakHold value for maximum positive total output	Base Unit / s

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x0000000C	Read	every	REAL64	e.g. mm		Set position correction value for drive output due to dead time compensation	
0x0000000D	Read	every	REAL64	s		Sum of the time shifts for drive dead time compensation (parameterized and variable dead time) Note: a dead time is specified in the system as a positive value.	
0x00000013	Read	every	REAL64	%		Total output in percent (according to non-linear characteristic!)	
0x00000014	Read	every	REAL64	V		Total output in volts (according to non-linear characteristic!)	Cannot be traced by oscilloscope!
0x0000011A	Read	Servo (Sercos, CANopen)	REAL64	e. g. mm		Optional Smoothingfilter: Filtered set position	NEW For Sercos, CANopen
0x0000011E	Read	Servo (Sercos, CANopen)	REAL64	e. g. mm/s		Optional Smoothingfilter: Filtered set velocity	NEW For Sercos, CANopen
0x0000011F	Read	Servo (Servo, CANopen)	REAL64	e. g. mm/s^2		Optional Smoothingfilter: Filtered set acceleration /deceleration	NEW For Sercos, CANopen

5.4.1.7.3 "Index offset" specification for drive functions (Index group 0x7200 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000102	Write	SERVO	{			Remove and delete the characteristic drive table	Only for SAF-Port 501!
			ULONG	1	>0	See Axis function with index offset 0x00000012	
			}				

5.4.1.7.4 "Index offset" specification for cyclic drive process data (Index group 0x7300 + ID)

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000000	Read/Write	every (NC→IO)	{		STRUCT see drive interface or see extended drive interface	DRIVE-OUTPUT-STRUCTURE (NC→IO, 12 bytes) or DRIVE-OUTPUT-STRUCTURE (NC→IO, 40 bytes)	Write command only optional! Consider safety aspects!
			INT32	INC	≥ 0	nOutData1	
			INT32	INC	±2^31	nOutData2	
			UINT8	1	≥ 0	nControl1	
			UINT8	1	≥ 0	nControl2	
			UINT8	1	≥ 0	nControl3	
			UINT8	1	≥ 0	nControl4	
			Optional:				
			INT32	INC	≥ 0	nOutData3	
			INT32	INC	≥ 0	nOutData4	
			INT32	INC	≥ 0	nOutData5	
			INT32	INC	≥ 0	nOutData6	
			UINT8	1	≥ 0	nControl5	
			UINT8	1	≥ 0	nControl6	
			UINT8	1	≥ 0	nControl7	
			UINT8	1	≥ 0	nControl8	
			INT32	1	≥ 0	reserved	
			INT32	1	≥ 0	reserved	
			}				
0x00000001	Write	every (IO→NC)	{		STRUCT s. drive interface	Bitwise access to DRIVE-OUTPUT-STRUKTUR (NC→IO, 40 Byte) *NC-DRIVESTRUCT_OUT2*	Write command only optional! Consider safety aspects! From TC 2.11 R3 B2303
			UINT32	1	[0 … 39]	ByteOffset Relative address offset [0..39] in output structure. E.G.: To write “nControl1” the ByteOffset must be 8.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	BitSelectMask (BSM) The mask defines write enabled bits in a DWORD. Zero bits are protected and remain unaffected.	
			UINT32	1	[0x00000000… 0xFFFFFFFF]	Value Only those bits in value are overwritten where BSM equals 1.	
			}				

Index offset (Hex)	Access	Drive type	Data type	Phys. Unit	Definition range	Description	Note
0x00000080	Write	every (IO→NC)	{		STRUCT see drive interface or see extended drive interface	DRIVE-INPUT-STRUCTURE (IO→NC, 12 bytes) or DRIVE-INPUT-STRUCTURE (IO→NC, 40 bytes)	
			INT32	INC	≥ 0	nInData1	
			INT32	INC	±2^31	nInData2	
			UINT8	1	≥ 0	nStatus1	
			UINT8	1	≥ 0	nStatus2	
			UINT8	1	≥ 0	nStatus3	
			UINT8	1	≥ 0	nStatus4	
			Optional:				
			INT32	INC	≥ 0	nInData3	
			INT32	INC	≥ 0	nInData4	
			INT32	INC	≥ 0	nInData5	
			INT32	INC	≥ 0	nInData6	
			UINT8	1	≥ 0	nStatus5	
			UINT8	1	≥ 0	nStatus6	
			UINT8	1	≥ 0	nStatus7	
			UINT8	1	≥ 0	nStatus8	
			INT32	1	≥ 0	Reserved	
			INT32	1	≥ 0	Reserved	
			}				

5.4.1.8 Specification Tables

5.4.1.8.1 "Index offset" specification for table parameter (Index group 0xA000 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000001	Read	every	UINT32	1	[1 ... 255]	Table ID	
0x00000002	Read	every	UINT8[30+1]	1	30 characters	Table name	
0x00000003	Read	every	UINT32	1	s. ENUM (>0)	Table sub types [▶ 143]	
0x00000004	Read	every	UINT32	1	s. ENUM (>0)	Table main types [▶ 142]	
0x00000010	Read	every	UINT32	1	[0... 16777216]	Number of lines (n)	
0x00000011	Read	every	UINT32	1	[0... 16777216]	Number of columns (m)	
0x00000012	Read	every	UINT32	1	≥0	Number of total elements (n*m)	
0x00000013	Read	equidistant table	REAL64	e.g. mm	≥0.0	Step size (position delta) (equidistant tables)	Base Unit
0x00000014	Read	cyclical table	REAL64	e.g. degree	≥0.0	Master period (cyclical tables)	Base Unit

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
0x00000015	Read	cyclical table	REAL64	e.g. degree	≥0.0	Slave difference per master period (cyclic tables)	Base Unit
0x0000001A	Read/Write	"Motion Function" (motion laws)	{			Activation type for online changes of table data (MF only)	
			UINT32	ENUM	s. appendix	Activation mode 0: 'instantaneous' (default) 1: 'master cam pos.' 2: 'master' axis pos.' 3: 'next cycle' 4: 'next cycle once' 5: 'as soon as possible' 6: 'off' 7: 'delete queued data'	
			REAL64	e.g. mm	±∞	Activation position	
			UINT32	ENUM	s. appendix	Master scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			UINT32	ENUM	s. appendix	Slave scaling type 0: user defined (default) 1: scaling with auto offset 2: off	
			}				
0x00000020	Read/Write	every	{			Write single value [n,m]:	
			UINT32	1	[0 ... 16777216]	n-th line	
			UINT32	1	[0 ... 16777216]	m-th column	
			REAL64	e.g. mm	±∞	Single value	
			}				
0x00000021	ReadWrite	every	*REAL64	e.g. mm	±∞	Read slave position for the specified master position (related to the "raw values" in the table)	
0x00000022	ReadWrite	"Motion Function" (motion laws)	**Write**			Read the "Motion Function" as a "point cloud"	Only possible on a line-by-line basis! (multiple integer)
			{				

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
			UINT16	1	0/1	Prompt consistent data adoption?	
			UINT16	1	Bitmask (≥0)	Select bit mask (number of columns m is master position plus number of bits): Bit 0: Pos (Slave) Bit 1: Velo (Slave) Bit 2: Acc (Slave) Bit 3: Jerk (Slave)	
			REAL64	e.g. mm	±∞	Start position (Master)	
			REAL64	e.g. mm	> 0.0	Step size	
			}				
			Read				
			{				
			REAL64[x*m]	e.g. mm	±∞	Read x rows starting from the master start position: (x*m)-values (one or more lines)	
			}				
0x00000023	ReadWrite	every	**Write**			Read slave values for the specified master position (related to the "raw values" in the table)	
			REAL64	e.g. mm	±∞	Master position	
			Read				
			{				
			REAL64	e.g. mm	±∞	Slave position	
			REAL64	mm/s	±∞	Slave velocity	
			REAL64	mm/s^2	±∞	Slave acceleration	
			}				
0x00000050	Read/Write	every	REAL64 [64]	1	±∞	Characteristic values in the table	
0x00000050	ReadWrite	every	**Write**			Read the characteristic values in a table in relation to the nominal master velocity	
			REAL64 [64]	...	±∞	Optional nominal master reference velocity "fMasterVeloNom" (standardized => 1.0 mm/s), the remaining elements are not evaluated	
			Read				

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
			REAL64 [64]	...	±∞	Read the characteristic values in a table	
0x00000115	Write	monotonic linear, monotonic cycl.,	{			Set /change the table scaling:	
			REAL64	1	[±1000000.0]	Original weighting of the table	
			REAL64	e.g. mm	[±1000000.0]	Master column position offset	
			REAL64	1	[±1000000.0]	Scaling of the master column	
			REAL64	e.g. mm	[±1000000.0]	Position offset of the slave column	
			REAL64	1	[±1000000.0]	Scaling of the slave column	
			REAL64	e.g. mm	[±1000000.0]	Lower range limit (starting position)	
			REAL64	e.g. mm	[±1000000.0]	Upper range limit (end position)	
			}				

0x01000000 +n-th start line	Read/ Write[<=16777216]	every	{
			REAL64[x*m]
			}
				e.g. mm	±∞	Read/Write x lines starting from the n-th line:
						(x*m)-values (one or more lines)
						Value range n: [0 ... 16777216]
							Only possible on a line-by-line basis!
							(multiple integer)

0x02000000 +m-th start column	Read/ Write[<=16777216]	every	{ REAL64[x*n] }	e.g. mm	±∞	Read/Write x columns starting from the m-th column: (x*n)-values (one or more lines) Value range m: [0 ... 16777216]	Only possible on a column-by-column basis! (multiple integer)
0x05000000 +n-th start line	Read/ Write[<=16777216]	"Motion Function" (motion laws) Data: STRUCT[x*m]	{			Read/Write x lines starting from the n-th line: (x*m)-structures (one or more lines) Value range n: [0 ... 16777216]	Only possible on a line-by-line basis! (multiple integer)
			UINT32	1		Abs. point index (not evaluated)	
			UINT16	ENUM		Function type 1: Polynomial 1 15: Polynomial 5	
			UINT16	ENUM		Point type 0: default 1: ignore	

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
			INT32	1		Rel. address index at target point (Default: 1)	
			REAL64	mm		Master position	
			REAL64	mm		Slave position	
			REAL64	mm/s		Slave velocity	
			REAL64	mm/s^2		Slave acceleration	
			REAL64	mm/s^3		Slave jerk	
			}				
0x06000000 +m-th start column	Read/ Write[<=16777216]	"Motion Function" (motion laws) Data: STRUCT[x*n]	{			Read/Write x columns starting from the m-th column: (x*n)-structures (one or more lines) Value range m: [0 ... 16777216]	Only possible on a column-by-column basis! (multiple integer)
			UINT32	1		Abs. point index (not evaluated)	
			UINT16	ENUM		Function type 1: Polynomial 1 15: Polynomial 5	
			UINT16	ENUM		Point type 0: default 1: ignore	
			INT32	1		Rel. address index at target point (Default: 1)	
			REAL64	mm		Master position	
			REAL64	mm		Slave position	
			REAL64	mm/s		Slave velocity	
			REAL64	mm/s^2		Slave acceleration	
			REAL64	mm/s^3		Slave jerk	
			}				

5.4.1.8.2 "Index offset" specification for table state (Index group 0xA100 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
0x0000000A	Read	every	INT32	1	≥ 0	'User Counter' (Number of users of this table)	Cannot be traced by oscilloscope!

5.4.1.8.3 "Index offset" specification for table functions (Index group 0xA200 + ID)

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks

Index offset (Hex)	Access	Table type	Data type	Phys. unit	Definition range	Description	Remarks
0x00010000	Write	Cam plate	{			Generates a cam plate table with dimension (n*m)	Table types: ,2,3,4 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00010001	Write	Characteristic curve	{			Generates characteristic table with dimensions (n*m):	Table types: 1,3 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00010010	Write	"Motion Function" (motion laws)	{			Generates "Motion Function" table with dimension (n*m):	Table types: 3.4 Dimension: at least 2x1
			UINT32	1	s. ENUM (>0)	Table type (see appendix)	
			UINT32	1	[2...16777216]	Number of lines	
			UINT32	1	[1...16777216]	Number of columns	
			}				
0x00020000	Write	every	VOID			Deletes table with dimension (n*m)	Table types: 1,2,3,4
0x00030000	Write	every	VOID			Initializes table Initialization is no longer needed, because now it happens automatically in the following cases. a) when coupling by means of a table b) when reading out the slave position (see table para.)	

5.4.1.9 Appendix

Enum Channel types

Define	Channel types
1	Standard
2	Interpreter
3	FIFO

Define	Channel types
4	Kinematic transformation

Enum Interpreter types

Define	Interpreter types
0	NOT DEFINED
1	reserved
2	DIN 66025 (Siemens dialect)

Enum Interpreter Operation modes

Define	interpreter/channel operation mode
0x0	Default (deactivates the other modes)
0x1	Single block mode in the NC core (Block execution task/SAF)
0x1000	reserved
0x2000	reserved
0x4000	Single block mode in the interpreter

Enum Interpolation load log mode

Define	Load log mode
0	Loader log off
1	Source only
2	Source & Compiled

Enum Interpolation Trace mode

Define	Trace mode
0	Trace off
1	Trace line numbers
2	Trace Source

Enum Interpreter state

See Interpreter state

Enum Group types

Define	Group types
0	NOT DEFINED
1	PTP-Group + x Slave
2	1D-Group + x Slave
3	2D-Group + x Slave
4	3D-Group + x Slave
5	High/low speed + x Slave
6	Low cost stepper motor (dig. IO) + x Slave
7	Table Group + x Slave
9	Encoder Group + x Slave
11	FIFO Group + x Slave
12	Kinematic Transformation Group + x Slave

Enum Curve velocity reduction method

See Curve velocity reduction method

Enum Axis types

Define	Axis types
0	NOT DEFINED
1	Continuous axis (Servo)
2	Discrete axis (high/low speed)
3	Continuous axis (stepper motor)

Enum Stepper motor operation mode

Define	Stepper motor operation mode
0	NOT DEFINED
1	2-phase excitation (4 cycles)
2	1-2-phase excitation (6 cycles)
3	Power section

Enum Override types for PTP axes (velocity override)

Define	Override types
1	Reduced Old variant, replaced by "(3) Reduced (iterated)"
2	Original Old variant, replaced by "(4) Original (iterated)"
3	Reduced (iterated) Default value: the override value is related to the velocity which is internally reduced in a special case. This results in a directly proportional velocity (=> linear relationship) for the entire override range from 0 to 100%.
4	Original (iterated) The override value is always referred to the velocity programmed by the user. If this velocity cannot be driven, however, then a maximum override value results from which no higher velocity can be reached (=> limitation).

Enum Group/axis start types

Define	Group/axis start types
0	NOT DEFINED
1	Absolute start
2	Relative start
3	Continuous start positive
4	Continuous start negative
5	Modulo start (OLD)
261	Modulo start on the shortest distance
517	Modulo start in positive direction (with modulo tolerance window)
773	Modulo start in negative direction (with modulo tolerance window)

Define	Group/axis start types
4096	Stop and lock (axis locked for motion commands)
8192	Halt (without motion lock)

Enum Command buffer types (buffer mode) for universal axis start (UAS)

Define	Buffer mode
0	ABORTING (default) (instantaneous, aborts current movement and deletes any buffered commands)
1	BUFFERED (stored in command buffer to be executed after an active movement)
18	BLENDING LOW (buffered, no stop, runs through intermediate target position at the lowest velocity of two commands)
19	BLENDING PREVIOUS (buffered, no stop, runs through intermediate target position at the velocity of the active command)
20	BLENDING NEXT (buffered, no stop, runs through intermediate target position at the velocity of the buffered command)
21	BLENDING HIGH (buffered, no stop, runs through intermediate target position at the highest velocity of two commands)

Enum End position types (new end position)

Define	End position types
0	NOT DEFINED
1	Absolute position
2	Relative position
3	Continuous position positive
4	Continuous position negative
5	Modulo position

Enum Command types for new end position with new velocity (new end position and/or new velocity)

Define	Command types for new end position with new velocity
0	NOT DEFINED
1	Position (instantaneous)
2	Velocity (instantaneous)
3	Position and velocity (instantaneous)
9	Position (switching position)
10	Velocity (switching position)
11	Position and velocity (switching position)

Enum Actual position types (set actual position)

Define	Actual position types
0	NOT DEFINED
1	Absolute position
2	Relative position

Define	Actual position types
5	Modulo position

Enum Compensation types (path compensation or superimposed)

Define	Compensation types
0	NOT DEFINED
1	VELOREDUCTION_ADDITIVEMOTION The max. velocity VelocityDiff is reduced. The path over which the compensation trip is effective consists of length + distance.
2	VELOREDUCTION_LIMITEDMOTION The max. velocity VelocityDiff is reduced. The path over which the compensation trip is effective is defined by the Length parameter.
3	LENGTHREDUCTION_ADDITIVEMOTION The max. available path is reduced and consists of length + distance. The system tries to use the max. velocity VelocityDiff.
4	LENGTHREDUCTION_LIMITEDMOTION The max. available path is reduced and is limited by the Length parameter. The system tries to use the max. velocity VelocityDiff.

Enum Slave types

Define	Slave types
0	NOT DEFINED
1	Linear
2	Flying saw (velocity, jerk restricted profile)
3	Flying saw (position and velocity, jerk restricted profile)
5	Synchronization generator (velocity, jerk restricted profile)
6	Synchronization generator (position and velocity, jerk restricted profile)
10	Tabular
11	Multi-tabular
13	'Motion Function' (MF)
15	Linear with cyclic gearing factor change (ramp filter for acceleration limits)
100	Specific

Enum Slave decoupling types (for subsequent axis command)

Define	Slave decoupling types (for subsequent axis command)
0	Stop, E-stop or P-stop (default) (STOP)
1	Oriented stop (O-stop) (ORIENTEDSTOP)
2	Reduce any acceleration to 0 (force-free) and continue to endless target position

Define	Slave decoupling types (for subsequent axis command)
	(ENDLESS)
3	Continue to endless target position at new requested velocity (ENDLESS_NEWVELO)
4	New end position (NEWPOS)
5	New end position and new requested velocity (NEWPOSANDVELO)
6	Logical decoupling and stopping of axis immediately without velocity ramp (INSTANTANEOUSSTOP)

Enum Controller types

Define	Controller types
0	NOT DEFINED
1	P-controller (standard) (Position)
2	PP-controller (with ka) (Position)
3	PID-controller (with ka) (Position)
5	P-controller (Velocity)
6	PI-controller (Velocity)
7	High/low speed controller (Position)
8	Stepper motor controller (Position)
9	SERCOS controller (Position in the drive)
10	RESERVED
11	RESERVED
12	RESERVED
13	RESERVED
14	TCom Controller (Soft Drive) (Position in the drive)

Enum Controller Observer mode

Define	Controller observer mode
0	No observer active (default)
1	"Luenberger" observer (classic observer design)

Requires control loop with torque interface

Enum Encoder types

Define	Encoder types
0	NOT DEFINED
1	Simulation Encoder (Incremental)
2	M3000 Encoder (Multi/Single-Turn) (Absolute)
3	M31x0 / M2000 Encoder (Incremental)
4	MDP 511 Encoder: EL7041, EL7342, EL5101, EL5151, EL2521, EL5021, IP5101 (Incremental)
5	MDP 500/501 Enc.: EL5001, IP5009, KL5001 (SSI) (Absolute)
6	MDP 510 Encoder: KL5051, KL2502-30K Encoder (BiSSI) (Incremental)
7	KL30xx Encoder (Analog) (Absolute)
8	SERCOS and EtherCAT SoE (Position) (Incremental)
9	SERCOS and EtherCAT SoE (Position and velocity) (Incremental)
10	Binary encoder (0/1) (Incremental)
11	M2510 Encoder (Absolute)
12	FOX50 Encoder (Absolute)
14	AX2000 (Lightbus) (Incremental)
15	Provi-Drive MC (Simodrive 611U) (Incremental)
16	Universal encoder (variable bit mask) (Incremental)
17	NC back plane (Incremental)
18	Special CANopen type (e.g. Lenze Drive 9300) (Incremental)
19	MDP 513 (DS402): CANopen and EtherCAT CoE (AX2xx-B1x0/B510, EL7201) (Incremental)
20	AX2xx-B900 (Ethernet) (Incremental)
21	KL5151 Encoder (Incremental)
24	IP5209 Encoder (Incremental)

Define	Encoder types
25	KL2531/KL2541 Encoder (Stepper Motor) (Incremental)
26	KL2532/KL2542 Encoder (DC motor), KL2535/ KL2545 (PWM current terminal) (Incremental)
27	Time base encoder (Time Base Generator) (Incremental)
28	TCom Encoder (Soft Drive) (Incremental)

Enum Encoder mode

Define	Encoder mode
0	NOT DEFINED
1	Determination of position
2	Determination of position and velocity
3	Determination of position, velocity and acceleration

Enum Encoder evaluation direction (log. counting direction)

Define	Encoder evaluation direction (log. counting direction)
0	Evaluation in positive and negative counting direction (default configuration, i.e. compatible with the previous state)
1	Evaluation only in positive counting direction
2	Evaluation only in negative counting direction
3	Evaluation neither in positive nor in negative counting direction (evaluation blocked)

Not for all encoder types; only for KL5101, KL5151, KL2531, KL2541, IP5209, Universal encoder, etc.

	Encoder types		
Encoder evaluation direction (Log. counting direction)	KL5101, ...	Universal Encoder	other types
0: positive and negative	√	√	—
1: only positive	√	√	—
2: only negative	√	√	—
3: disabled	√	√	—

Enum Encoder sign interpretation (data type)

Define	Sign interpretation (data type) of the encoder actual increments
0	NOT DEFINED (default configuration, i.e. compatible with the previous state)
1	UNSIGNED: unsigned interpretation of the encoder actual increments

Define	Sign interpretation (data type) of the encoder actual increments
2	SIGNED: signed interpretation of the encoder actual increments

For KL30xx/KL31xx only for the time being

Enum Encoder absolute dimensioning system

Define	Encoder absolute dimensioning system
0	INC: Incremental absolute dimension system with underflow and overflow offset (default, i.e. compatible with the previous state)
1	ABS: Absolute dimension system without underflow and overflow offset (no underflow or overflow of the encoder allowed)
2	ABS MODULO: Conditionally absolute dimension system, since it has underflow and overflow offset (absolute value that modulo (endless) continues)

Not for all encoder types; only for Profi Drive MC, M3000, KL5001/EL5001, IP5009, SERCOS, UNIVERSAL, etc.

Enum Encoder position initialization

Define	Encoder position initialization
0	Direct adoption of the position increments without further logic (default configuration, i.e. compatible with the previous state)
1	With underflow and overflow offset logic (direct adoption, or underflow or overflow offset)

For SERCOS only for the time being

Enum Reference mode for incremental encoder

Define	Reference mode for incremental encoder
0	NOT DEFINED (default configuration, i.e. compatible with the previous state)
1	Latch event: shutdown of the PLC cam (falling edge)
2	Latch event: Hardware sync pulse (zero track)
3	Latch event: External hardware latch with rising edge (measuring sensor or, respectively, measurement on the fly with rising edge)
4	Latch event: External hardware latch with falling edge (measuring sensor or, respectively, measurement on the fly with falling edge)
5	Latch event: Synthetically emulated software sync pulse (software zero track); REQUIREMENT: absolute per motor revolution, e.g. resolver!

Define	Reference mode for incremental encoder
6	Latch event: Hardware latch event defined in the drive with rising edge (e.g. for SoftDrive) (NEW)
7	Latch event: Hardware latch event defined in the drive with falling edge (e.g. for SoftDrive) (NEW)
20	Application defined Homing sequence (PLC code): Application defined Homing request is signaled to the PLC with the ApplicationRequest-Bit (NEW)

	Referencing mode: Latch event					
Encoder types	**0: not defined**	**1: PLC cam (falling edge)**	**2: Hardware Sync pulse (zero-/C-track)**	**3: External hardware Latch with rising edge**	**4: External hardware Latch with falling edge**	**5: Software Sync pulse (Software zero track)**
AX2xxx-B200 (Lightbus)	—	√	√	√	√	√ (only resolver)
AX2xxx-B510 (CANopen)	—	√	—	—	—	√ (only resolver) (see "Reference mask" parameter)
AX2xxx-B1x0 (EtherCAT)	—	√	√	√	√	√ (only resolver) (fixed 20-bit)
AX2xxx-B900 (Ethernet)	—	√	√	√	√	√ (only resolver)
Sercos	—	√	√	√ (AX5xxx specific implemented)	√	√ (see "Reference mask" parameter)
Profi Drive	—	√	√	√	√	√
KL5101 IP5109	—	√	√	√	√	√
KL5111	—	√	√	—	—	√
KL5151	—	√	√	√	√	√ (not useful)
IP5209	—	√	√	—	—	√ (not useful)
CANopen (e.g. Lenze)	—	√	—	√ (Input In1)	√ (Input In2)	√ (only resolver) (fixed 16-bit)
others Types	—	—	—	—	—	—

Enum Drive types

Define	Drive types
0	NOT DEFINED

Define	Drive types
1	Analog Servo Drive: M2400 DAC 1 (Analog)
2	Analog Servo Drive: M2400 DAC 2 (Analog)
3	Analog Servo Drive: M2400 DAC 3 (Analog)
4	Analog Servo Drive: M2400 DAC 4 (Analog)
5	MDP 252 Drive: Analog Servo Drive: KL4xxx, KL2502-30K (Analog)
6	MDP 252 Drive: Analog Servo Drive (non-linear): KL4xxx, KL2502-30K (Analog)
7	High/low speed drive (Digital)
8	Stepper motor drive (Digital)
9	SERCOS-Drive (Digital)
10	MDP 510 Drive: KL5051 (BiSSI-Interface) (Digital)
11	AX2000 (Lightbus) (Digital)
12	Provi-Drive MC (Simodrive 611U) (Digital)
13	Universal Drive (Analog)
14	NC back plane (Analog)
15	Special CANopen type (e.g. Lenze Drive 9300) (Digital)
16	MDP 742 (DS402): CANopen and EtherCAT CoE (AX2xx-B1x0/B510) (Digital)
17	AX2xx-B900 Drive (Ethernet) (Digital)
20	KL2531/KL2541 Encoder (Stepper Motor) (Digital)
21	KL2532/KL2542 Encoder (DC motor), KL2535/ KL2545 Encoder (PWM current terminal) (Digital)
22	TCom Drive (Soft Drive) (Digital)
23	MDP 733 Drive: Profile MDP 733 (EL7332, EL7342, EP7342) (Digital)

Define	Drive types
24	MDP 703 Drive: Profile MDP 703 (EL7031, EL7041, EP7041) (Digital)

Enum Drive-Output-Start types

Define	Enum Drive-Output-Start types
0	NOT DEFINED
1	Output value in percent
2	Output as velocity, e.g. m/min

Enum Moving phases / Movement state for master axes

Define	Moving phases / Movement state (distinction between internal and external setpoint generation)
Internal setpoint generation	
0	Setpoint generator not active (INACTIVE)
1	Setpoint generator active (RUNNING)
2	Velocity override is zero (OVERRIDE_ZERO)
3	Constant velocity (PHASE_VELOCONST)
4	Acceleration phase (PHASE_ACCPOS)
5	Deceleration phase (PHASE_ACCNEG)
External setpoint generation:	
41	External setpoint generation active (EXTSETGEN_MODE1)
42	Internal and external setpoint generation active (EXTSETGEN_MODE2)

Enum Moving phases / Movement state for slave axes

Define	Moving phases / Movement state
0	Slave generator not active (INACTIVE)
11	Slave is in a movement pre-phase (PRE-PHASE)
12	Slave is synchronizing (SYNCHRONIZING)
13	Slave is synchronized and moves synchronously (SYNCHRON)

Only for slaves of the type synchronization generator for the time being

Enum Table main types

Define	Table main types
1	(n*m) Cam plate tables (Camming)
10	(n*m) Characteristic curves tables (Characteristics) (e.g. hydraulic valve characteristics) Only non-cyclic table sub-types (1, 3) are supported!
16	(n*m) "Motion Function" tables (MF) Only non-equidistant table sub-types (3, 4) are supported!

Enum Table sub-types

Define	Table sub types
1	(n*m) Table with equidistant master positions and no cyclic continuation of the master profile (equidistant linear)
2	(n*m) Table with equidistant master positions and cyclic continuation of the master profile (equidistant cyclic)
3	(n*m) Table with non-equidistant, but strictly monotonously increasing master positions and a non-cyclic continuation of the master profile (monotonously linear)
4	(n*m) Table with non-equidistant, but strictly monotonously increasing master positions and a cyclic continuation of the master profile (monotonously cyclic)

Enum Table interpolation types

Define	Table interpolation types between the reference points
0	Linear interpolation (NC_INTERPOLATIONTYPE_LINEAR) (Standard)
1	4-point interpolation (NC_INTERPOLATIONTYPE_4POINT) (for equidistant table types only)
2	Cubic spline interpolation of all reference points ("global spline") (NC_INTERPOLATIONTYPE_SPLINE
3	sliding cubic spline interpolation of n reference points ("local spline") (NC_INTERPOLATIONTYPE_SLIDINGSPLINE)(from TC V2.11 B1514)

Structure of the tables (CAM) coupling information

Tables		(CAM) coupling information
nTableID;	1.	cam table ID
nTableMainType;	2.	e.g. CAMMING, CHARACTERISTIC, MOTIONFUNCTION
nTableSubType;	3.	e.g. EQUIDIST_LINEAR, EQUIDIST_CYCLE, NONEQUIDIST_LINEAR, NONEQUIDIST_CYCLE
nInterpolationType;	4.	e.g. LINEAR, 4POINT, SPLINE
nNumberOfRows;	5.	number of rows/elements
nNumberOfColumns;	6.	number of columns
fRawMasterPeriod;	7.	master period/cycle (raw value, not scaled)
fRawSlaveStroke;	8.	slave difference per master period/ cycle (raw value, not scaled)
fMasterOffset;	9.	total master offset
fSlaveOffset;	10.	total slave offset
fMasterScaling;	11.	total master scaling

Tables		(CAM) coupling information
fSlaveScaling;	12.	total slave scaling
fActualMasterAxisPos;	13.	actual master axis setpos (absolute)
fActualSlaveAxisPos;	14.	actual slave axis setpos (absolute)
fActualMasterCamPos;	15.	actual master cam setpos
fActualSlaveCamPos;	16.	actual master cam setpos

Structure of the characteristic values

Characteristic values		
fMasterVeloNom;	1.	master nominal velocity (normed: => 1.0)
fMasterPosStart;	2.	master start position
fSlavePosStart;	3.	slave start position
fSlaveVeloStart;	4.	slave start velocity
fSlaveAccStart;	5.	slave start acceleration
fSlaveJerkStart;	6.	slave start jerk
fMasterPosEnd;	7.	master end position
fSlavePosEnd;	8.	slave end position
fSlaveVeloEnd;	9.	slave end velocity
fSlaveAccEnd;	10.	slave end acceleration
fSlaveJerkEnd;	11.	slave end jerk
fMPosAtSPosMin;	12.	master pos. at slave min. position
fSlavePosMin;	13.	slave minimum position
fMPosAtSVeloMin;	14.	master pos. at slave min. velocity
fSlaveVeloMin;	15.	slave minimum velocity
fMPosAtSAccMin;	16.	master pos. at slave min. acceleration
fSlaveAccMin;	17.	slave minimum acceleration
fSVeloAtSAccMin;	18.	slave velocity at slave min. acceleration
fSlaveJerkMin;	19.	slave minimum jerk
fSlaveDynMomMin;	20.	slave minimum dynamic momentum (NOT SUPPORTED YET!)
fMPosAtSPosMax;	21.	master pos. at slave max. position
fSlavePosMax;	22.	slave maximum position
fMPosAtSVeloMax;	23.	master pos. at slave max. velocity
fSlaveVeloMax;	24.	slave maximum velocity
fMPosAtSAccMax;	25.	master pos. at slave max. acceleration
fSlaveAccMax;	26.	slave maximum acceleration
fSVeloAtSAccMax;	27.	slave velocity at slave max. acceleration
fSlaveJerkMax;	28.	slave maximum jerk
fSlaveDynMomMax;	29.	slave maximum dynamic momentum (NOT SUPPORTED YET!)
fSlaveVeloMean;	30.	slave mean absolute velocity
fSlaveAccEff;	31.	slave effective acceleration

Enum Axis control loop switch types

Define	Axis control loop switch types
0	NOT DEFINED
1	Simple switching (similar to an axis reset) (STANDARD)
2	Switching/synchronization by means of I/D-part of the controller to an internal initial value (jerk-free/smooth)
3	Switching/synchronization by means of I/D-part of the controller to a parameterisable initial value

5.5 TwinCAT ADS Device CAM

The TwinCAT camshaft controller can be described as a virtual field device (automation device). It makes a TwinCAT ADS (Automation Device Specification) interface available for other communication partners (such as other virtual field devices or Windows programs), through which it can be parameterised or queried. The use of ADS standardises the access to the camshaft controller, placing it in the group of available virtual field devices.

The READ and WRITE operations on the camshaft controller interface occur via two indices, as specified through ADS: the index group [▶ 145] (32bit) and the index offset (32bit).
The following pages provide a more detailed description of the group and offset indices of the camshaft controller ADS interface.

5.5.1 ADS index groups of the camshaft controller

The four global areas of a camshaft controller are shown as four sections in the index groups as follows:

Index-Group (0x = hex)	Index Group description
0x00000000 0x00000FFF	reserved
0x00001000	Cam controller 1
0x00001001	Encoder of camshaft controller 1
0x00001101 0x00001180	Parameters of cams 1 - 128 of camshaft controller 1
0x00002000 0x00004000	Camshaft controllers 2 - 4 (optional)
0x0000F000 0x0000FFFF	General TwinCAT ADS system services

Via an additional index offset, a READ or WRITE operation can access an element from the selected area. All services are addressed via port 900.
The description of the individual areas, their elements and the respective index offsets are described in the following sections.

5.5.2 ADS Index tables cam shaft controller

5.5.2.1 Index table encoder

Index group	Index Offset	Access	Data type	Phys. unit	Def. Range	Description
0x00001001	0x00000200	R/W	UINT8[n]	-		Read resp. write the total encoder data set
0x00001001	0x00000201	R/W	UINT8[n]	1/10 degree		Zero point compensatio n , Machine configuration
0x00001001	0x00000202	R/W	UINT8[n]	1/10 degree		Dynamic positionoffset (e.g. OT shift)
0x00001001	0x00000210	R/W	UINT8[n]	-	1/2	Box address of the encoder

5.5.2.2 Index table Camshaft controller parameter

Index group	Index Offset	Access	Data type	Phys. unit	Def. range	Description
0x00001000	0x00000000	R/W	UINT8[n]	-		Read resp. write the total encoder data set
0x00001000	0x00000001	R	UINT32	-		quantitiy of current parameterise d cams
0x00001000	0x00000002	R/W	UINT8	-	0/1	Boolean flag to activate the cam calculation
0x00001000	0x00000010	R/W	UINT8	-	0/1	Boolean flag for reversal of rotation direction aktiv(ieren) active/ activate
0x00001000	0x00000011	R/W	UINT8	-	0/1	Boolean flag for encoder monitoring active/ activate
0x00001000	0x00000012	R/W	UINT8	-	0/1	Boolean flag for shaft break monitoring active/ activate

Index group	Index Offset	Access	Data type	Phys. unit	Def. range	Description
0x00001000	0x00000013	R/W	UINT8	-	0/1	Boolean flag for stroke rate monitoring active/ activate
0x00001000	0x00000014	R/W	UINT8	-	0/1	Boolean flag for state "coupling"
0x00001000	0x00000020	R/W	UINT8	-	0/1	Boolean flag for encoder error + reset
0x00001000	0x00000021	R/W	UINT8	-	0/1	Boolean flag for shaft break + reset
0x00001000	0x00000022	R/W	UINT8	-	0/1	Boolean flag for rotary error + reset
0x00001000	0x00000031	R/W	UINT32	1/min	0-2000	Tolerance for stroke rate deviation
0x00001000	0x00000032	R/W	UINT32	Ms	0-65.535 s	Tolerance time range for stroke rate deviation
0x00001000	0x00000033	R/W	UINT32	1/min	0-2000	Current stroke rate
0x00001000	0x00000034	R/W	UINT32	1/min	0-2000	Stroke rate set value
0x00001000	0x00000035	R/W	UINT32	Ms	0-65.535 s	Tolerance time range for shaft break error
0x00001000	0x00000100	R/W	UINT32	-		Or mask to force the 32 cam bits
0x00001000	0x00000101	R/W	UINT32	-		And mask to force 32 cam bits
0x00001000	0x00000301	R/W	UINT32	-		Count value of counter 1
0x00001000	0x00000302	R/W	UINT32	-		Count value of counter 2
0x00001000	0x00000303	R/W	UINT32	-		Count value of counter 3
0x00001000	0x00000304	R/W	UINT32	-		Count value of counter 4
0x00001000	0x00000305	R/W	UINT32	-		Count value of counter 5

5.5.2.3 Index table cam parameter

Index group	Index Offset	Access	Data type	Phys. unit	Def. range	Description
0x00001101	0x00000000	R/W	UINT8[n]	-		Read resp. write the total encoder data set
0x00001101	0x00000001	R	UINT8	-	0-3	Cam type: 0-None, 1-Way cam, 2-time cam, 3-Brake cam, 4-Multi cycle cam
0x00001101	0x00000002	R/W	UINT32	degree	0-359	Initial value (Way cam)
0x00001101	0x00000003	R/W	UINT32	degree	0-359	Final value (Way cam)
0x00001101	0x00000004	R/W	UINT32	-	0-31	Cam track
0x00001101	0x00000005	R/W	UINT32	0.1 ms	0-6553.5 ms	Hold-up time for cam dynamic
0x00001101	0x00000006	R/W	UINT32	1/min	1-65535	Count value for count function, Cam is every n-th stroke active; **At multi cycle cam**: quantitiy of cycles in which the cam has the value "0".
0x00001101	0x00000007	R/W	UINT32	Incr	0-8192	Encoder increment quantitiy for hysteresis functions
0x00001101	0x00000008	R/W	UINT32	0.1 ms	0.1-6553.5 ms	Duration (Time cam)
0x00001101	0x00000009	R/W	UINT32	0.1 ms	0.-6553.5 ms	final value (Brake cam)
0x00001101	0x00000010	R/W	UINT8	-	0/1	Boolean flag for forward active
0x00001101	0x00000011	R/W	UINT8	-	0/1	Boolean flag for backward active
0x00001101	0x00000012	R/W	UINT8	-	0/1	Boolean flag for dynamic calculation
0x00001101	0x00000013	R/W	UINT8	-	0/1	Boolean flag to load counter
0x00001101	0x00000014	R/W	UINT8	-	0/1	Boolean flag to activate hysteresis

Index group	Index Offset	Access	Data type	Phys. unit	Def. range	Description
0x00001101	0x00000015	R	UINT8	-	0/1	State of external brake enable
0x00001101	0x00000016	R/W	UINT8	-	0/1	Force the brake bits
0x00001101	0x00000017	R/W	UINT32	-	2-65535	**Only at multi cycle cam**: quantitiy of cycles in which the cam has the state "1".

5.5.2.4 Index tables of the TwinCAT ADS system services

Index-Group	Index offset	Access	Data type	Phys. unit	Def. range	Description
0x0000F003	0x00000000	R&W	W:UINT8[n] R: UINT32			GET_SYMHANDLE_BYNAME The name contained in the write data is assigned a handle (characteristic value) and returned to the caller as a result.
0x0000F004	0x00000000	R&W W R	 UINT8[n] UINT8[n]			READ_SYMVAL_BYNAME The value of a named variable is read from the device. The variable names are specified while this write and read service is called. The response contains the variable value.

Index-Group	Index offset	Access	Data type	Phys. unit	Def. range	Description
0x0000F005	0x00000000-0xFFFFFFFF =symHandle	R/W	UINT8[n]			READ_/WRITE_SYMVAL_BYHANDLE Reading of the value of the variable identified by 'symHdl', or allocating a value to the variable. Prior to this, ‚symHdl' must have been determined via the GET_SYMHNADLE_BYNAME service.
0x0000F006	0x00000000-0xFFFFFFFF =symHandle	W	UINT32			RELEASE_SYMHANDLE The characteristic number contained in the write data (handle) for a named variable to be queried is enabled.

5.5.3 Process image cam shaft controller

Input range:

Adress-offset	Data type	Phys. unit	Def. range	Description
0x0000	UINT32			Alternative encoder input (cam shaft has no private fieldbus)
0x0004	UINT32		0/1	Rotary direction inverted
0x0008	UINT32		0/1	State: "Engaged"
0x000C	UINT32			Set stroke count
0x0010	UINT32			Maximum allowed stroke count difference to error monitoring
0x0014	UINT32		0/1	Reset I/O error,
0x0018	UINT32		0/1	Reset encoder error
0x001C	UINT32		0/1	Reset shaft break error
0x0020	UINT32		0/1	Reset rotary error

Adress-offset	Data type	Phys. unit	Def. range	Description
0x0024	UINT32		0/1	Counter enable
0x0028	UINT32			Set value counter 1
0x002C	UINT32			Set value counter 2
0x0030	UINT32			Set value counter 3
0x0034	UINT32			Set value counter 4
0x0038	UINT32			Set value counter 5
0x003C+n	UINT32		0/1	External brake enable (1 Bit each cam track), n=(output groups-1)
0x0040+n	UINT32		0/1	"AND" mask for cam outputs
0x0044+n	UINT32		0/1	"OR" maske for cam outputs

Output range:

Adress-offset	Data type	Phys. unit	Def. Range	Description
0x0000	UINT32			Unconverted encoder value
0x0004	UINT32			Encoder value in 0.1 degree
0x0008	UINT32			Current stroke count
0x000C	UINT32			Quantitiy of configurated cams
0x0010	UINT32			Quantitiy of II/O error
0x0014	UINT32		0/1	Encoder error
0x0018	UINT32		0/1	Shaft break error
0x001C	UINT32		0/1	Rotary error
0x0020	UINT32			Timeout error 1 at private fieldbus
0x0024	UINT32			Timeout error 2 at private fieldbus
0x0028	UINT32			Counter 1
0x002C	UINT32			Counter 2
0x0030	UINT32			Counter 3
0x0034	UINT32			Counter 4
0x0038	UINT32			Counter 5
0x003C+n	UINT32		0/1	Inputs at private fieldbus
0x0040+n	UINT32		0/1	Cam outputs , n=(output groups-1)

6 How to...

6.1 ADS device identification

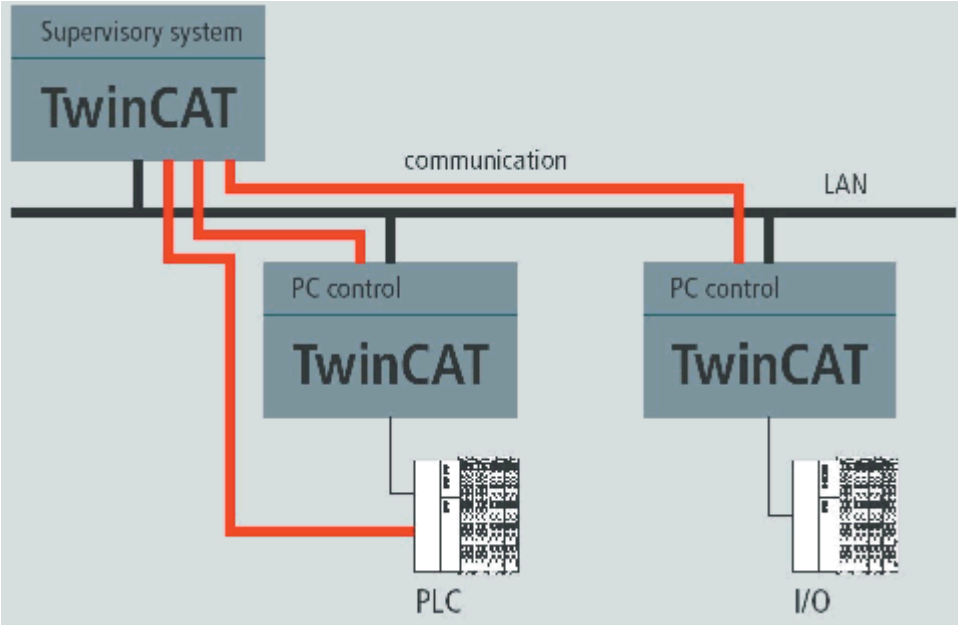
The unique identification of ADS devices is implemented with the aid of two identifiers:

- AdsAmsNetId
- AdsPortNr

(Note: In terms of a telecommunications analogy, the AdsAmsNetId corresponds to the "STD code" and the ADS-PortNr to the "subscriber number ".)

ADS-AmsNetId

It is not only possible to exchange data between TwinCAT modules on one PC, it is even possible to do so by ADS methods between multiple TwinCAT PC's on the network.



Every PC on the network can be uniquely identified by a TCP/IP address, such as "172.1.2.16". The AdsAmsNetId is an extension of the TCP/IP address and identifies a TwinCAT message router, e.g. "172.1.2.16.1.1". TwinCAT message routers exist on every TwinCAT PC, and on every Beckhoff BCxxxx bus controller (e.g. BC3100, BC8100, BC9000, ...).

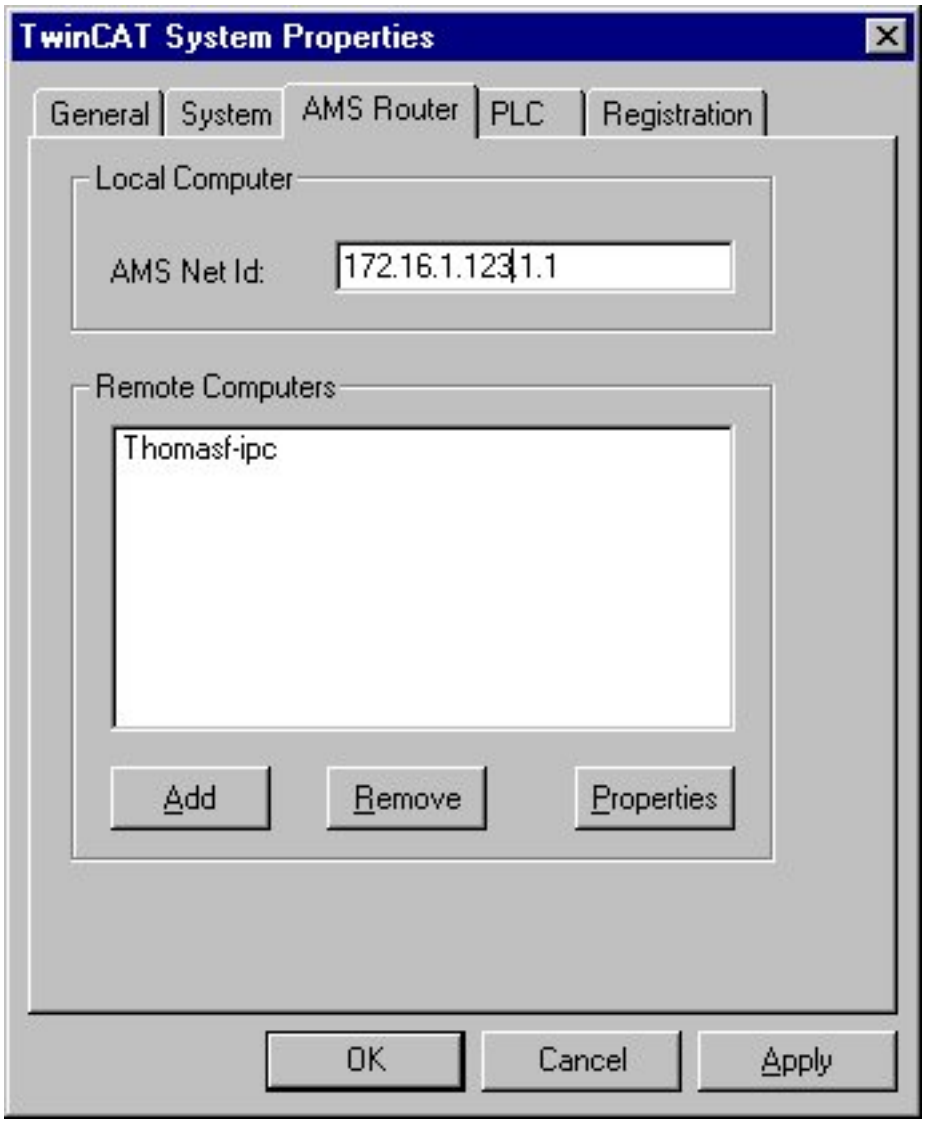
Configuration:

The ADS-AmsNetId of a TwinCAT-PC can be set in the TwinCAT system service: A menu appears after the system service icon has been selected. By selecting "Properties" you reach the "TwinCAT System Properties" dialogue.

After the "AMS router" page has been selected, the desired identifier can be read or modified in the "AMS Net Id" field. By default, the TwinCAT installation constructs the AMSNetId from the TCP/IP address of the PC with an extension of ".1.1" (as if it were a "sub-net mask" for field busses, target bus controllers,..).

The AMSNetId can in fact be freely chosen.

When the services of an ADS device in the network are called on, its message router, or, more precisely, its AMSNetId, must be known. This is achieved through insertion of the target PC, because of which TwinCAT can establish the connection between the TCP/IP address of the target PC and the AMSNetId of the target message router address.



ADS-Port

The ADS devices in a TwinCAT message router are uniquely identified by a number referred to as the ADS-PortNr. For ADS devices this has a fixed specification, whereas pure ADS client applications (e.g. a visualization system) are allocated a variable ADS port number when they first access the message router.

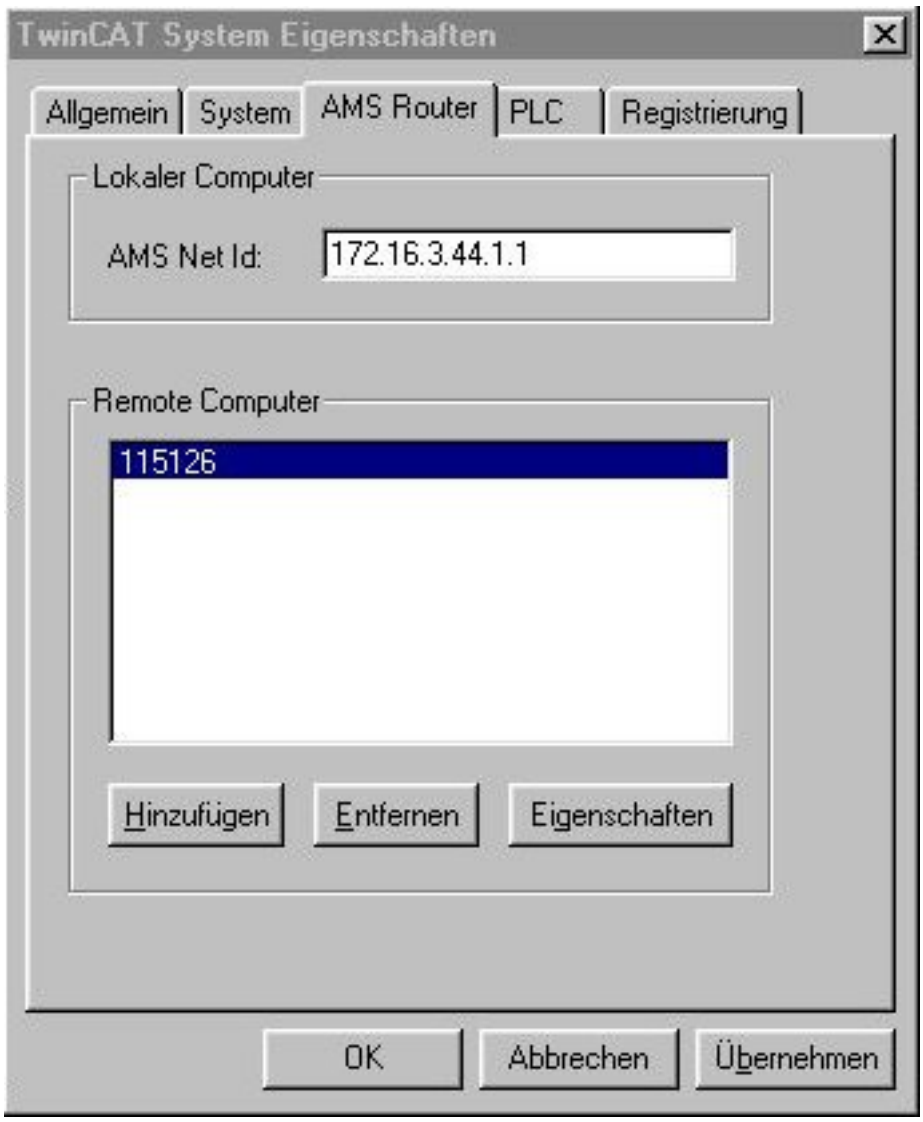
The following ADS port numbers are already assigned:

ADS-PortNr	ADS device description
100	**Logger (only NT-Log)**
110	**Eventlogger**
300	**IO**
301	**additional Task 1**
302	**additional Task 2**

ADS-PortNr	ADS device description
500	**NC**
801	**PLC RuntimeSystem 1**
811	**PLC RuntimeSystem 2**
821	**PLC RuntimeSystem 3**
831	**PLC RuntimeSystem 4**
900	**Camshaft controller**
10000	**System Service**
14000	**Scope**

6.2 Setting Up an ADS Remote Connection

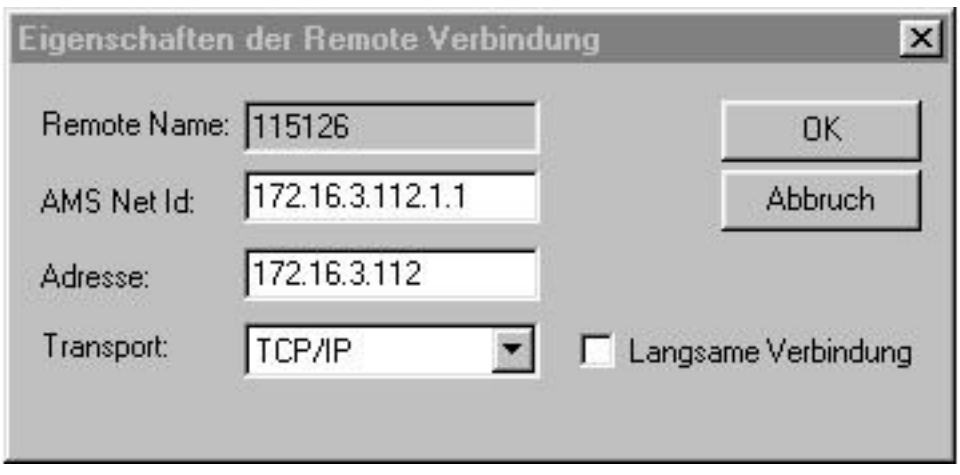
In order to create an AMS connection between a number (n) of computers, these computers need to be fitted with network cards. The network must be set up under Windows NT. If the computer is being operated in a LAN network, then it is possible to obtain the TCP/IP address of the computer via the DHCP server. If, for instance, the remote connection is being used to service equipment that is in operation, then it is recommended that a fixed TCP/IP address be assigned.



Assigning a fixed address also prevents the event display being filled up with the following error message: 'DHCP could not obtain a lease for the IP address for the card with network address 00E0F40FC80F. The following error occurred: The time limit for the semaphore was reached.'.

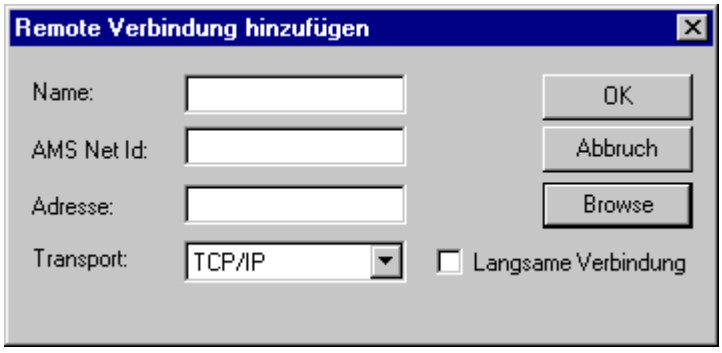
The network connection between the n computers should be tested using the command "ping <PCI/IP-address of the target computer>". If the network test is successful, it is possible to begin the settings in TwinCAT. The remote computers must first be made known to one another.

Entries necessary for this purpose are made under TwinCAT+-> Properties +-> AMS Router.



The AMS Net ID is composed of the TCP/IP of the local computer plus the suffix ".1.1". The AMS Net ID is based on the TCP/IP address, but the relationship is not entirely fixed.

Add should be pressed in order to include a remote computer in the connection.



It is possible to enter the parameters manually in his window, or to select them using Browse. This procedure is repeated n-1 times. Take care to see that the data in the window for adding a remote connection agree with the data for the remote computer. The host computer is added in the same way at the remote computer.

When the settings have been entered, they are confirmed with the Return key. The n computers can be restarted at this point.

In PLC Control, under Online +-> Select target system, one of the n computers can be selected as the PLC platform for the current PLC project.

6.3 Cookbook "How to implement an ADS client"

0. Overview

Always good to start with overview

Following samples are based on C++, however they work identically for C#

See device concept [▶ 8]

1. Get started and open/connect to ADS router

Overview C++ samples

The easy (*but not recommended way*) to read data: Read flag synchronously from the PLC by absolute adress information
See sample

Disadvantage: If PLC code or PLC variable declaration is changing the ADS-client will not notice this in any case.
WE HIGHLY RECOMMEND to work on handles of variables (see Agenda 3)

2. Which variables available

By compiling PLC code the PLC-Editor "PLC-Control" creates an XML-based export file <projectname.tpy> containing all information about symbols (variables) and their properties (datatype etc).
At HMI design time this file can be imported to Scada editor to configure mapping between variable and HMI-object.

We are preparing sample code to parse type on demand, will be available here

3. Get handles of variables

ADS-List-command: Get/Release multiple handles with one ADS call

Instead looping through a routine to get handles we recommend, to save protocol transport time by requesting multiple handles with one single ADS call

See sample C++

NOTE:

Devices with previous TwinCAT-versions might not support this call (ADS return code “Service not supported”). In this case collect handles one by one.

4. Cyclic collection of life data

ADS Sum Command : Read

Works also for write direction

The Sub-IndexGroup should always be 0xF005 in combination with the Sub-IndexOffset filled with the handle of variable.

See sample C++

5. Release handles

ADS-List-command: Get/Release multiple handles with one ADS call
Instead looping through a routine to release handles we recommend, to save protocol transport time by releasing multiple handles with one single ADS call

See sample C++

Previous TwinCAT-versions

Devices with previous TwinCAT-versions might not support this call (ADS return code “Service not supported”) In this case release handles one by one.

6. Close ADS channel

Always close ADS channel PortClose() – be aware to release handles before !

7. In case of ADS error

Always good to take care on return codes

See ADS return codes
In case of return code “Invalid handle” somebody downloaded a new PLC code or restarted the TwinCAT machine.

Release existing handles and request new handles, after this continue collection data.

6.4 How to... "determine router memory"

In cases with high ADS message volume the router memory should be increased.

Calculation

The calculation of the router memory depends on several factors. The following rule of thumb can be used for guidance:

Router memory = packet size per notification* + 500 kB message buffer + 200 kB management information

*The packet sizes of the notification depends on the selected parameters (Cycletime and Maxdelay). The minimum size if 1 kB per notification.

Example*:

10 controls, each with 20 notifications, are used (DINT values, Cycle Time = 1 ms, MaxDelay = 100 ms).

In this example each notification message requires 1220 bytes (rounded up to 2 kB, since the memory is specified in kB).

Data field	Size	Description
Length	4 bytes	Data size in bytes.
Stamps	4 bytes	Number of elements of type AdsStampHeader.
TimeStamp	8 bytes	Timestamp. The timestamp is encoded according to the Windows FILETIME format. This means that the value contains the number of nanoseconds that have passed since 1.1.1601. The local time difference is not taken into account. The timestamp is in Universal Coordinated Time (UTC) format.
Samples	4 bytes	Number of elements of type AdsNotificationSample.

The MaxDelay parameter ensures that 100 values are put in the buffer:

Notification Handle	100 x 4 bytes	100 notification handle.
Sample Size	100 x 4 bytes	Size of the data area in bytes.
Data	100 x 4 bytes	DINT data.

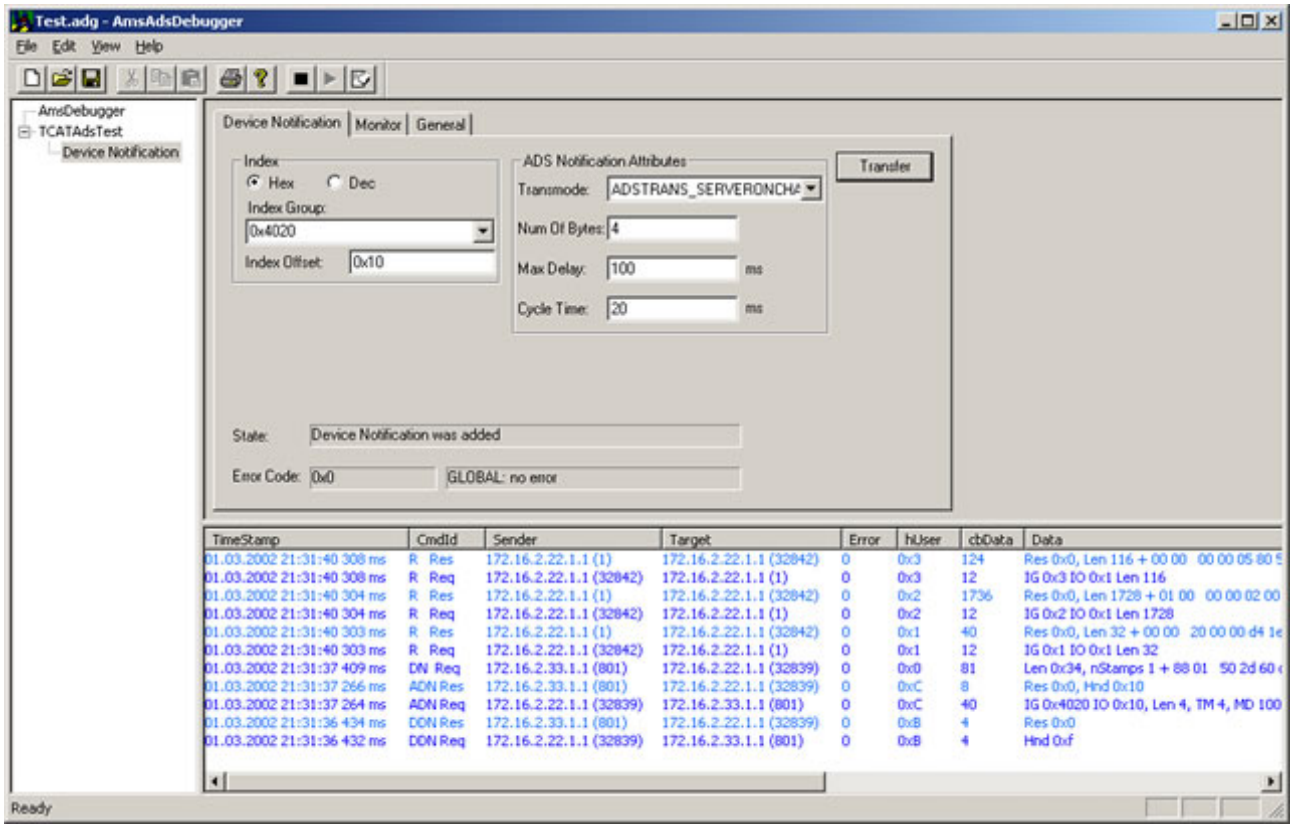
Required router memory* = 10 controllers x 20 notifications/controller x 2 kB/notifications + 500 kB message buffer + 200 kB management information = 1100 kB

* The calculation refers to a case with maximum memory usage. The memory allocation may differ during normal operation.

6.5 Diagnostic aid

6.5.1 TwinCAT AMS/ADS Debugger

The *TwinCAT AMS/ADS Debugger* logs all data traffic via the TwinCAT Message Router. This also affects the communication between ADS devices if they are located on the same computer. E.g. the data exchange between TwinCAT NC and TwinCAT PLC or between the TwinCAT cam controller and the fieldbus. The *TwinCAT AMS/ADS Debugger* is located on the TwinCAT CD in the directory *Unsupported Utilities*.



Not an official Beckhoff product

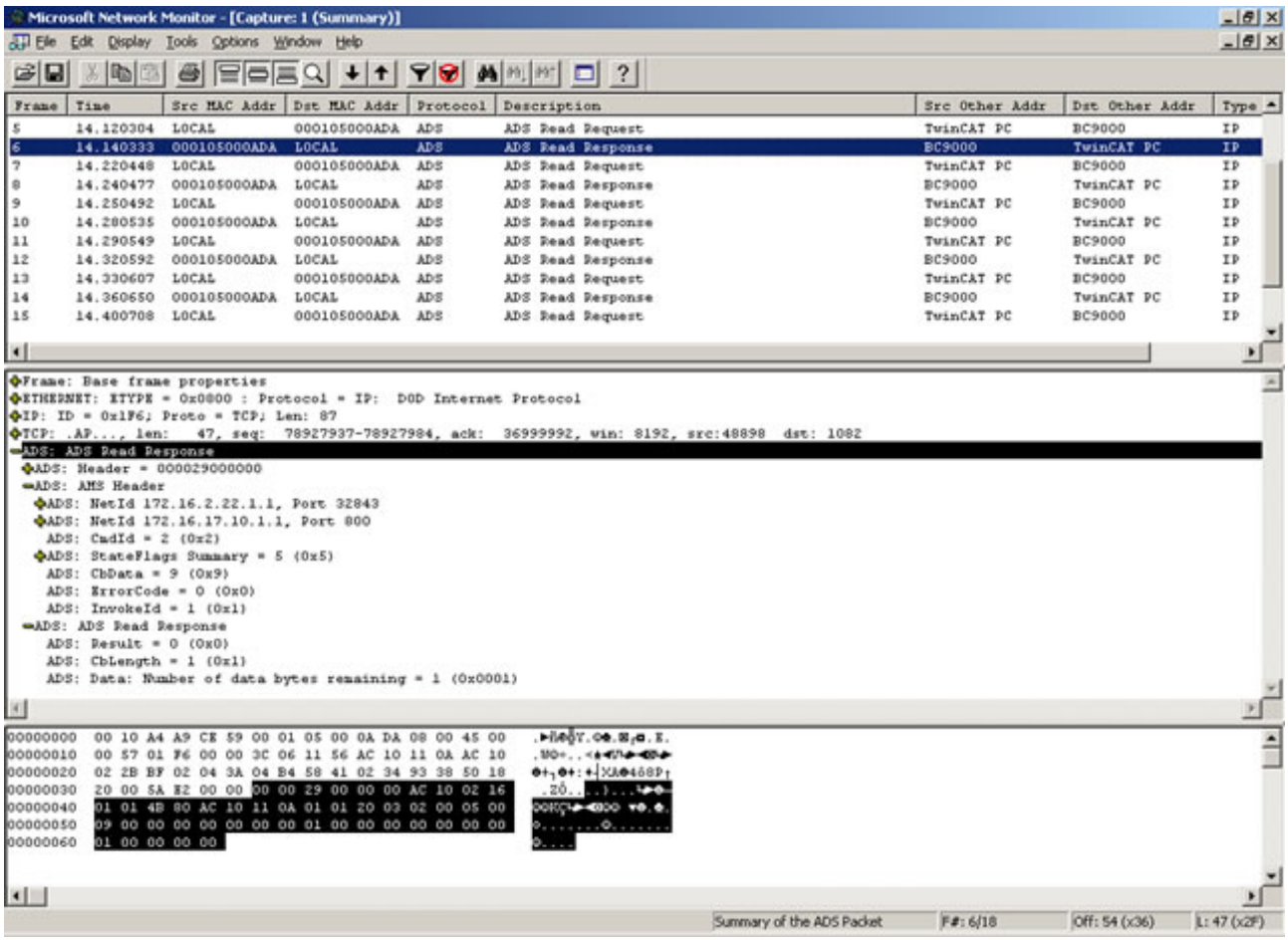
It must be pointed out at this point that the TwinCAT AMS/ADS Debugger is not an official product of the Beckhoff company, so that we do not assume any warranty or support for its use. However, we are grateful for information on corresponding errors.

6.5.2 Microsoft Network Monitor

The *MS Network Monitor 2* can be extended by further protocols using appropriate parser DLLs. For the AMS/TCP protocol such an extension DLL is delivered with TwinCAT.

This DLL is located on the TwinCAT CD in the directory *Unsupported Utilities* and must be copied to the subdirectory *Parsers* of the *MS Network Monitor*.

With the help of the *MS Network Monitor* only the communication that takes place via the Ethernet interface can be logged. Recording with other fieldbuses (Lightbus, Profibus, ...) is not possible.



Not a Beckhoff product

It must be pointed out at this point that the MS Network Monitor is not a Beckhoff product and therefore no support can be provided for it. Also, the parser DLL is not an official product, so we do not provide any warranty for its use. However, we are grateful for information on corresponding errors.

7 ADS Return Codes

Grouping of error codes:
Global error codes: ADS Return Codes [▶ 161]... (0x9811_0000 ...)
Router error codes: ADS Return Codes [▶ 161]... (0x9811_0500 ...)
General ADS errors: ADS Return Codes [▶ 162]... (0x9811_0700 ...)
RTime error codes: ADS Return Codes [▶ 163]... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started or is not reachable.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low –TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.

Hex	Dec	HRESULT	Name	Description
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	License period too long.

Hex	Dec	HRESULT	Name	Description
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Service contains an invalid parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes			

More Information:
www.beckhoff.com/automation

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