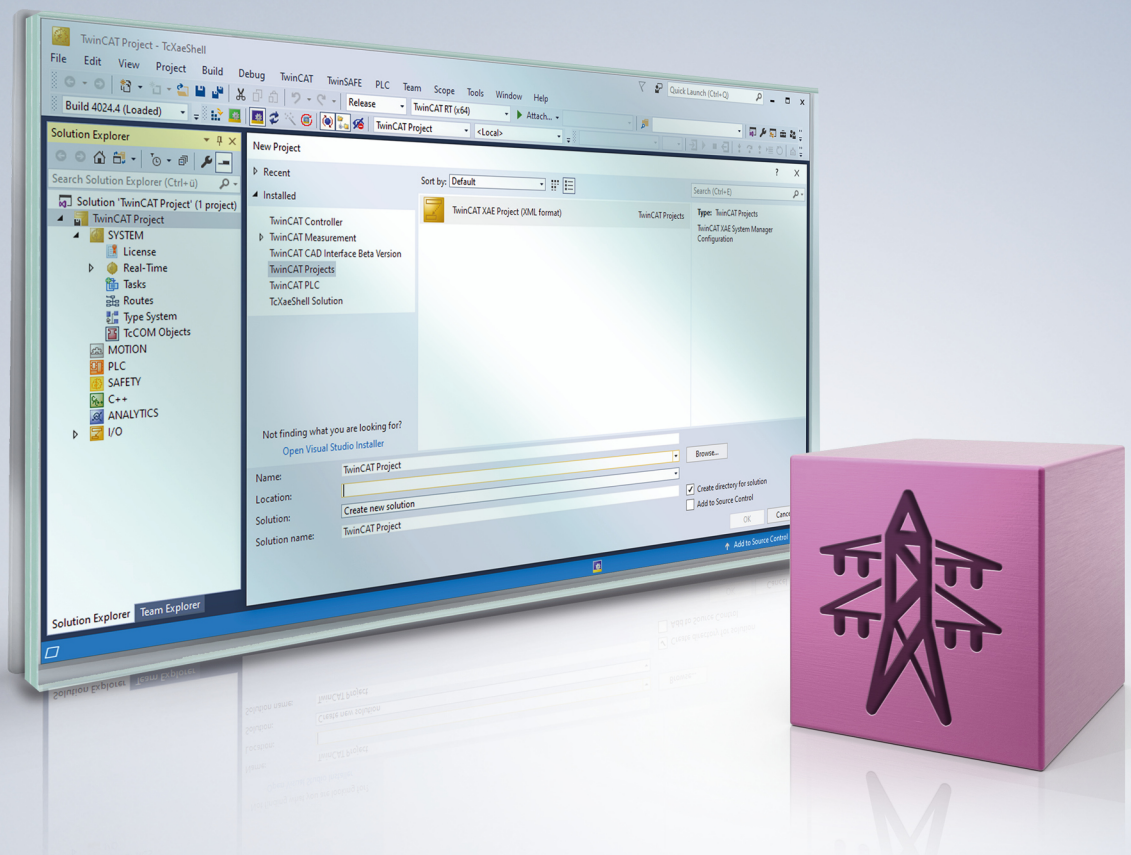
BECKHOFF New Automation Technology

Manual | EN

TF6500

TwinCAT 3 | IEC 60870-5-10x



2025-07-16 | Version: 1.6.2

Table of contents

1 Foreword 9
1.1 Notes on the documentation 9
1.2 For your safety 9
1.3 Notes on information security 11

2 Overview 12
2.1 Product description 12
2.2 Interoperability check list 13
2.2.1 Interoperability list according to IEC 60870-5-101 (master) 13
2.2.2 Interoperability list according to IEC 60870-5-101 (slave) 30
2.2.3 Interoperability list according to IEC 60870-5-102 (master) 47
2.2.4 Interoperability list according to IEC 60870-5-103 (master) 56
2.2.5 Interoperability list according to IEC 60870-5-104 (master) 67
2.2.6 Interoperability list according to IEC 60870-5-104 (slave) 85

3 Installation 103
3.1 System requirements 103
3.2 Installation 103
3.3 Installing TwinCAT/BSD 106
3.4 Installation Windows CE 106
3.5 Licensing 108
3.6 Migration from TwinCAT 2 110

4 Technical reference 112
4.1 IEC60870-5-101 112
4.1.1 IEC 60870-5-101 telegram structure 113
4.1.2 ASDU object description 117
4.2 IEC60870-5-102 239
4.2.1 IEC 60870-5-102 telegram structure 240
4.2.2 ASDU object description 242
4.3 IEC60870-5-103 330
4.3.1 IEC 60870-5-103 telegram structure 331
4.3.2 ASDU object description 333
4.4 IEC60870-5-104 394
4.4.1 IEC 60870-5-104 telegram structure 395
4.4.2 ASDU object description 399

5 PLC Libraries 488
5.1 Function blocks 488
5.1.1 High Level 488
5.1.2 Low Level 502
5.1.3 FB_IEC870_PartyLineCtrl 515
5.1.4 FB_IEC870_SerialLineCtrl 519
5.1.5 FB_IEC870_DebugLogFifo 521
5.1.6 FB_IEC870_5_101ErrorFifo 522
5.2 Functions 523
5.2.1 ASDU101COT_TO_STRING 523

5.2.2 ASDU102COT_TO_STRING ... 524
5.2.3 ASDU103COT_TO_STRING ... 524
5.2.4 ASDU101TYPEID_TO_STRING ... 525
5.2.5 ASDU102TYPEID_TO_STRING ... 526
5.2.6 ASDU103TYPEID_TO_STRING ... 526
5.2.7 CP40Time2a_TO_SYSTEMTIME ... 527
5.2.8 CP56Time2b_TO_CP40Time2a ... 528
5.2.9 CP56Time2b_TO_SYSTEMTIME ... 528
5.2.10 SYSTEMTIME_TO_CP56Time2b ... 529
5.2.11 CP56Time2a_TO_SYSTEMTIME ... 530
5.2.12 SYSTEMTIME_TO_CP56Time2a ... 530
5.2.13 CP56Time2a_TO_CP32Time2a ... 531
5.2.14 CP56Time2a_TO_CP24Time2a ... 532
5.2.15 CP24IOA_TO_DWORD ... 532
5.2.16 DWORD_TO_CP24IOA ... 533
5.2.17 IEC101ASDU_TO_STRING ... 533
5.2.18 IEC102ASDU_TO_STRING ... 534
5.2.19 IEC103ASDU_TO_STRING ... 534
5.2.20 INT_TO_NORMF16 ... 535
5.2.21 QOI_TO_COT ... 536
5.2.22 RQT_TO_COT ... 536
5.2.23 F_iecInitAOEntry ... 537
5.2.24 F_iecSetAOQuality ... 539
5.2.25 F_iecGetAOQuality ... 540
5.2.26 F_iecGetAOTimeTag ... 541
5.2.27 F_iecIncVTI ... 543
5.2.28 F_iecDecVTI ... 543
5.2.29 F_iecCopyBufferToStream ... 544
5.2.30 F_iecCopyBufferToStreamByOffset ... 545
5.2.31 F_iecCopyStreamToBuffer ... 546
5.2.32 F_iecCopyStreamToBufferByOffset ... 548
5.2.33 F_iecCopyStreamToStream ... 549
5.2.34 F_iecMoveStreamToBuffer ... 551
5.2.35 F_iecMoveStreamToStream ... 552
5.2.36 F_iecResetStream ... 553
5.2.37 F_iecCreateTableHnd ... 554
5.2.38 F_iecAddTableEntry ... 555
5.2.39 F_iecGetPosOfTableEntry ... 558
5.2.40 F_iecLookupTableEntry ... 560
5.2.41 F_iecRemoveTableEntry ... 562
5.2.42 F_iecCmpAddrOctets ... 563
5.2.43 F_iecGetSPI ... 564
5.2.44 F_iecGetDPI ... 565
5.2.45 F_iecGetSCS ... 566
5.2.46 F_iecGetDCS ... 566
5.2.47 F_iecSetSPI ... 567

5.2.48 F_iecSetDPI ... 568
5.2.49 F_iecSetSCS ... 569
5.2.50 F_iecSetDCS ... 570
5.2.51 F_iecChangeLinkLayerMode ... 570
5.2.52 F_iecApdu101ToAsduLen ... 573
5.2.53 F_iecApdu102ToAsduLen ... 573
5.2.54 F_iecApdu103ToAsduLen ... 574
5.2.55 F_iecApdu104ToAsduLen ... 575
5.2.56 IEC870_5_102_DEFAULT_ASDUFMTPARAMS ... 575
5.2.57 IEC870_5_104_DEFAULT_ASDUFMTPARAMS ... 576
5.2.58 Obsolete ... 577
5.3 Data types ... 582
5.3.1 ST_IEC870_5_101ExSystemInterface ... 582
5.3.2 ST_IEC870_5_101SystemInterface ... 582
5.3.3 ST_IEC870_5_101SystemServices ... 583
5.3.4 ST_IEC870_5_101TBuffer ... 583
5.3.5 ST_IEC870_5_101AODBEntry ... 584
5.3.6 ST_IEC870_5_101AOGen ... 585
5.3.7 ST_IEC870_5_101AOCfg ... 585
5.3.8 ST_IEC870_5_101DataUnit_Ident ... 586
5.3.9 ST_IEC870_5_101AOInfoObj ... 587
5.3.10 ST_IEC870_5_101Stream ... 587
5.3.11 ST_IEC870_5_101SystemParams ... 588
5.3.12 ST_IEC870_5_101DeviceInterface ... 591
5.3.13 ST_IEC870_5_101AsduFmtParams ... 591
5.3.14 ST_IEC870_5_101ErrorFifoEntry ... 592
5.3.15 ST_IEC870_5_101AcquisitionParams ... 593
5.3.16 ST_IEC870_5_101TestPollParams ... 595
5.3.17 ST_IEC870_5_101ClockPollParams ... 596
5.3.18 ST_IEC870_5_101GenroPollParams ... 596
5.3.19 ST_IEC870_5_101CoroPollParams ... 597
5.3.20 ST_IEC870_5_101GenCmdPollParams ... 597
5.3.21 ST_IEC870_5_101DelayPollParams ... 598
5.3.22 ST_IEC870_5_101HashTableKey ... 599
5.3.23 E_IEC870_5_101AcquisitionMode ... 600
5.3.24 E_IEC870_5_101TcTypeID ... 601
5.3.25 E_IEC870_5_102TypeID ... 605
5.3.26 E_IEC870_5_103MTypeID ... 607
5.3.27 E_IEC870_5_103CTypeID ... 607
5.3.28 E_IEC870_5_101IOMappingType ... 608
5.3.29 E_IEC870_5_101AsduAddrSize ... 608
5.3.30 E_IEC870_5_101COTSize ... 609
5.3.31 E_IEC870_5_101LinkAddrSize ... 609
5.3.32 E_IEC870_5_101ObjAddrSize ... 610
5.3.33 E_IEC870_5_101ErrorSourceID ... 610
5.3.34 E_IEC870_5_101ClassType ... 611

5.3.35 E_IEC870_5_101COTType ... 611
5.3.36 E_IEC870_5_102COTType ... 614
5.3.37 E_IEC870_5_103MCOT ... 615
5.3.38 E_IEC870_5_103CCOT ... 616
5.3.39 E_IEC870_5_101FifoDbgFlags ... 616
5.3.40 E_IEC870_5_101AODBType ... 617
5.3.41 E_IEC870_5_101InitSeqStep ... 617
5.3.42 E_IEC870_5_101FBufferState ... 622
5.3.43 E_IEC870_5_101SCS ... 622
5.3.44 E_IEC870_5_101DCS ... 622
5.3.45 E_IEC870_5_101COI ... 623
5.3.46 E_IEC870_5_101QOI ... 623
5.3.47 E_IEC870_5_101QL ... 624
5.3.48 E_IEC870_5_101FRZ ... 624
5.3.49 E_IEC870_5_101RQT ... 625
5.3.50 E_IEC870_5_101QRP ... 625
5.3.51 E_IEC870_5_101QU ... 626
5.3.52 E_IEC870_5_101ES ... 626
5.3.53 E_IEC870_5_101KPA ... 627
5.3.54 E_IEC870_5_101QPA ... 627
5.3.55 E_IEC870_5_101RCS ... 628
5.3.56 E_IEC870_5_101SPI ... 628
5.3.57 E_IEC870_5_101DPI ... 629
5.3.58 E_IEC870_5_101FrameType ... 629
5.3.59 E_IEC870_5_101LinkMode ... 629
5.3.60 E_IEC870_5_101SerialLinkState ... 630
5.3.61 E_IEC870_5_101PartyLineMode ... 630
5.3.62 E_IEC870_5_101LinkReset ... 631
5.3.63 E_IEC870_DEVICE_TYPE ... 631
5.3.64 E_IEC870_5_104DataTransferState ... 632
5.3.65 T_HSERIALCTRL ... 632
5.3.66 T_HAODBTable ... 633
5.3.67 T_CP56Time2a ... 633
5.3.68 T_CP56Time2b ... 634
5.3.69 T_CP40Time2a ... 635
5.3.70 T_CP32Time2a ... 635
5.3.71 T_CP24Time2a ... 636
5.3.72 T_CP16Time2a ... 636
5.3.73 T_CP24IOA ... 637
5.3.74 T_IEC870_5_101COTBits ... 637
5.3.75 ST_IEC870_5_101FBufferCfg ... 638
5.3.76 ST_IEC870_5_101FBufferStatus ... 639
5.3.77 ST_IEC870_5_101ProtocolParams ... 640
5.3.78 ST_IEC870_5_102TBuffer ... 645
5.3.79 ST_IEC870_5_102AOGen ... 646
5.3.80 ST_IEC870_5_102AOInfoObj ... 647

5.3.81 ST_IEC870_5_102DataUnit_Ident ... 647
5.3.82 ST_IEC870_5_103TBuffer ... 648
5.3.83 ST_IEC870_5_103AOGen ... 649
5.3.84 ST_IEC870_5_103AOInfoObj ... 649
5.3.85 ST_IEC870_5_103DataUnit_Ident ... 650
5.3.86 ST_IEC870_5_104ProtocolParams ... 651
5.3.87 ST_IEC870_5_104ExSystemInterface ... 653
5.3.88 ST_IEC870_5_104ServerConnection ... 653
5.3.89 ST_IEC870_5_104GrpStatus ... 654
5.3.90 ST_IEC870_5_104SystemInterface ... 654
5.4 Global constants ... 655
5.4.1 Group configuration flags ... 655
5.4.2 Quality flags ... 657
5.4.3 IEC 60870-5-10x error codes ... 658
5.4.4 Parameter list ... 660
5.4.5 Library version ... 661

6 Samples ... 662
6.1 Configuration of serial interfaces ... 662
6.2 IEC 60870-5-101 ... 665
6.2.1 Control station ... 665
6.2.2 Controlled station ... 693
6.3 IEC 60870-5-102 ... 724
6.3.1 Low-Level Interface ... 724
6.4 IEC 60870-5-103 ... 727
6.4.1 Low-Level Interface ... 727
6.5 IEC 60870-5-104 ... 731
6.5.1 Central station ... 731
6.5.2 Controlled station ... 755

7 Appendix ... 784
7.1 IEC 60870-5-101 ... 784
7.1.1 TwinCAT IEC 60870-5-101 error codes ... 784
7.1.2 Troubleshooting/diagnostics ... 788
7.1.3 Debug messages in TwinCAT->Error List ... 789
7.2 IEC 60870-5-102 ... 793
7.2.1 TwinCAT IEC 60870-5-102 error codes ... 793
7.2.2 Troubleshooting/diagnostics ... 797
7.2.3 Debug messages in TwinCAT->Error List ... 797
7.3 IEC 60870-5-103 ... 799
7.3.1 TwinCAT IEC 60870-5-103 error codes ... 799
7.3.2 Troubleshooting/diagnostics ... 803
7.3.3 Debug messages in TwinCAT->Error List ... 803
7.4 IEC 60870-5-104 ... 805
7.4.1 TwinCAT IEC 60870-5-104 error codes ... 805
7.4.2 Troubleshooting/diagnostics ... 864
7.4.3 Debug messages in TwinCAT->Error List ... 866

7.4.4 Firewall settings 868
7.5 Glossary 869
7.6 FAQ – frequently asked questions and answers 869

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
The documentation and the following notes and explanations must be complied with when installing and commissioning the components.
The trained specialists must always use the current valid documentation.

The trained specialists must ensure that the application and use of the products described is in line with all safety requirements, including all relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been compiled 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 notice.
Claims to modify products that have already been supplied may not be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of the designations or trademarks contained in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.
The distribution and reproduction of this document, as well as the use and communication of its contents without express authorization, are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

Third-party trademarks

Trademarks of third parties may be used in this documentation. You can find the trademark notices here: https://www.beckhoff.com/trademarks.

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are 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 technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product



This information includes, for example:
recommendations for action, assistance or further information on the product.

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 Overview

2.1 Product description

In telecontrol applications, devices from different manufacturers have to communicate with each other. Based on the five basic standards IEC 60870-5-1 (transmission frame formats), -2 (transmission procedures/ link layer), -3 (structures/application data), -4 (information elements) and -5 (basic application functions), the IEC 60870-5-family defines the application-related standards IEC 60870-5-101, -102, -103 and -104. The internationally standardized telecontrol protocols IEC 60870-5-101/-102/-103 enable implementation of serial data transmission, while the telecontrol protocol IEC 60870-5-104 is used for TCP/IP-based data transmission.

Manufacturers are not obliged to implement the complete standard in their devices. For this reason there may be incompatibilities between devices during commissioning. In order to prevent this, manufacturers offer a suitable compatibility list for each device. In this list the implemented functions are listed or marked. The compatibility list can be used to compare the required functionality of two devices beforehand. Please also refer to the compatibility lists for the TwinCAT IEC 60870-5-10x libraries.

The PLC functions and function blocks can be used to realize central stations and/or substations (master/ slave) in the TwinCAT PLC, based on the IEC60870-5-101/102/103/104 standard. Some of the PLC libraries have two software interfaces ("Low Level" and "High Level"). The end application is imposed on one of these interfaces. High Level is suitable for fast applications with lower requirements. Low Level offers many degrees of freedom and is ideal for sophisticated applications. Which interface is used depends thus on the final application.

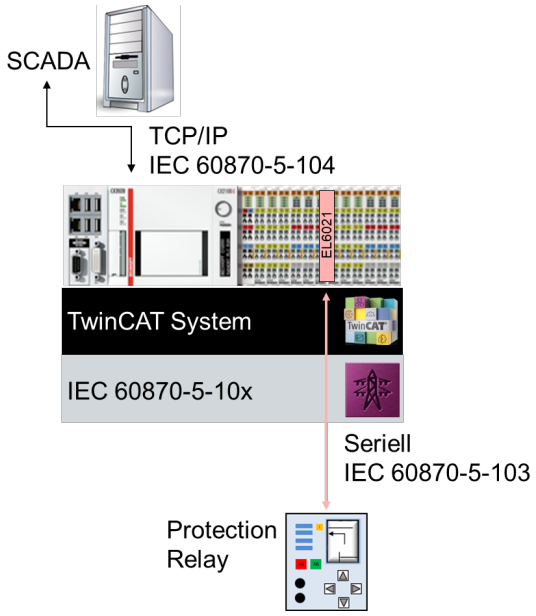
Product components

The function TF6500 | IEC 60870-5-10x consists of the following components, which are automatically installed during setup:

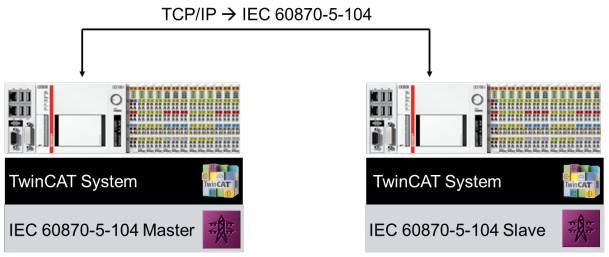
- **Functions** : TF6310 and TF6340.
- **PLC library:** Tc2_IEC60870_5_10x (Tc2_TcpIp, Tc2_SerialCom, installed with TF6310 and TF6340).
- **Background program:** TwinCAT TCP/IP Connection Server (for communication, installed with TF6310).

Operating principle and application scenarios:

IEC 60870-5-104 slave with subordinate -103 master for connecting protective devices.



One controller as IEC 60870-5-104 master, and one controller as connected -104 slave



Next steps:

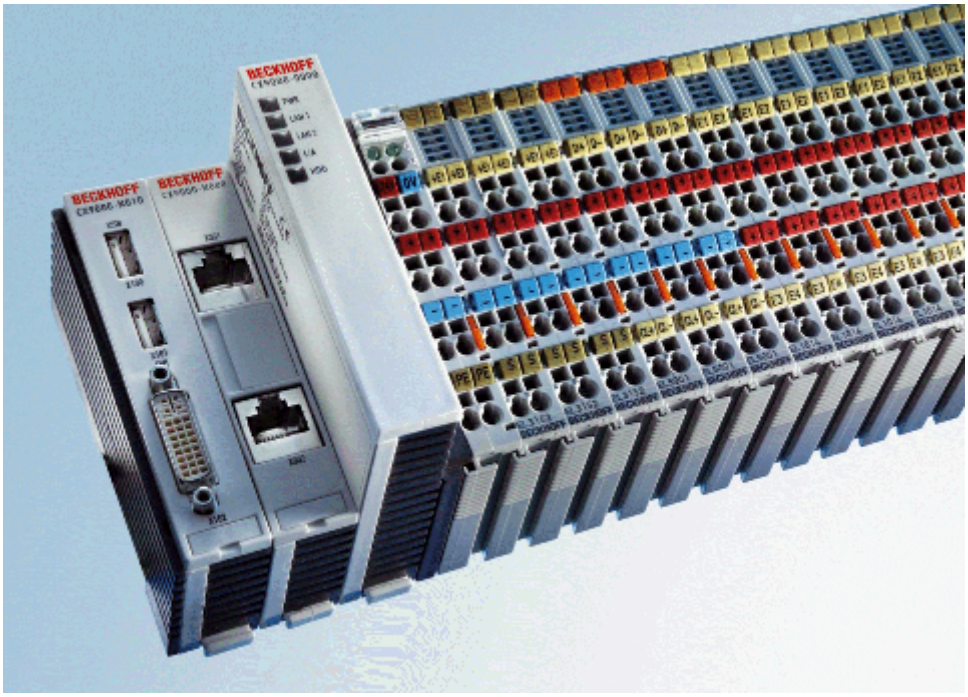
Selection of the interface - High or Low Level:

- Technical introduction IEC 60870-5-101 [▶ 112]
- Technical introduction IEC 60870-5-102 [▶ 239]
- Technical introduction IEC 60870-5-103 [▶ 330]
- Technical introduction IEC 60870-5-104 [▶ 394]
- Examples [▶ 662] of the different protocols.

2.2 Interoperability check list

2.2.1 Interoperability list according to IEC 60870-5-101 (master)

für TwinCAT SPS Bibliothek: IEC 60870-5-101 Zentralstation (master).



Version: 1.2
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984583435.zip.

Table of contents:

- General information [▶ 14]
  - Project information [▶ 14]

- Compatibility [▶ 15]
  - System or device [▶ 15]
  - Network configuration [▶ 15]
  - Physical layer [▶ 15]
  - Link layer [▶ 15]
  - Application layer [▶ 16]
  - Selection of standardised ASDUs

Process information in monitoring direction

Process information in control direction

System information in monitoring direction

System information in control direction

Parameter in control direction

File transfer

- Basic application functions [▶ 25]
  - Station initialization
  - Cyclic data transfer
  - Read procedure
  - Spontaneous transmission
  - Double transmission
  - Station interrogation
  - Clock synchronisation
  - Command transmission
  - Counter interrogation
  - Parameter loading
  - Parameter activation
  - Test procedure
  - File transfer
  - Background scan
  - Telegram run time acquisition
- Appendix [▶ 28]
  - Beckhoff support and service [▶ 28]

General Information

This compatibility list refers to functions supported within IEC 870-5-101.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Projekt No.:

_____

Project:

_____

Responsible:

_____

Date:

_____

Compatibility

The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as specified (preferred option)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in regular and reverse mode

System or device

☐	System definition
☒	Specifications for the control station (master)
☐	Specifications for the controlled station

Network configuration

☒	Point-to-point	☒	Multipoint-partyline
☒	Multiple point-to-point	☐	Multipoint-star

Physical layer

Transmission speed:

☐	100 bps	☒	2400 bps	☐	56000 bps
☐	200 bps	☒	4800 bps	☐	64000 bps
☒	300 bps	☒	9600 bps		
☒	600 bps	☒	19200 bps		
☒	1200 bps	☐	38400 bps		

Link layer

According to this application-related standard, only telegram format FT 1.2, single character 1 and the time monitoring interval are used.

Address field of the link layer:

☒	Not available (only balanced transfer)
☒	One octet
☒	Two octets
☒	Structured
☒	Non structured

Link layer transfer procedure:

☒	Balanced transfer
☒	Unbalanced transfer

Telegram length:

Maximum length L	255

In unbalanced mode some ASDU types are send with low (class 2) priority to the control station.

☐ Normed asssignment of ASDU type to class 2 data

Type identifier	Cause of transmission

☐ Special assignment of ASDU type to class 2 data

Type identifier	Cause of transmission

An outstation may return class 1 data when polled for class 2 data if there is no class 2 data available (configuration parameter).

Application layer

Transfer mode for application data

According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

Common ASDU address

☒	One octet	☒	Two octets

Address of the information object

☒	One octet	☒	Structured
☒	Two octets	☒	Non structured
☒	Three octets		

Cause of transmission

☒	One octet	☒	Two octets (with origin address). Preset to 0, if origin address is not available.

Selection of standardised ASDU's

Table 1: Process information in monitoring direction

☒	<1> :=Single-point information	M_SP_NA_1
☒	<2> := Single-point information with time tag	M_SP_TA_1
☒	<3> := Double point information	M_DP_NA_1
☒	<4> := Double point information with time tag	M_DP_TA_1
☒	<5> := Step position information	M_ST_NA_1
☒	<6> := Step position information with time tag	M_ST_TA_1
☒	<7> := Bitstring of 32 bits	M_BO_NA_1
☒	<7> := Bitstring of 32 bits with time tag	M_BO_TA_1
☒	<9> := Measured value, normalized	M_ME_NA_1
☒	<10> := Measured value, normalized value with time tag	M_ME_TA_1
☒	<11> := Measured value, scaled value	M_ME_NB_1
☒	<12> := Measured value, scaled value with time tag	M_ME_TB_1
☒	<13> := Measured value, short floating point value	M_ME_NC_1
☒	<14> := Measured value, short floating point value with time	M_ME_TC_1
☒	<15> := Integrated totals	M_IT_NA_1
☒	<16> := Integrated totals with time tag	M_IT_TA_1
☒	<17> := Event of protection equipment with time tag	M_EP_TA_1
☒	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
☒	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☒	<20> := Packed single-point information with status change detection	M_PS_NA_1
☒	<21> := Measured value, normalized value without quantity descriptor	M_ME_ND_1
☒	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
☒	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
☒	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
☒	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
☒	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
☒	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☒	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1

☒	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Table 2: Process information in control direction

☒	<45> := Single command	C_SC_NA_1
☒	<46> := Double command	C_DC_NA_1
☐	<47> := Regulating step command	C_RC_NA_1
☒	<48> := Set point command, normalized value	C_SE_NA_1
☒	<49> := Set point command, scaled value	C_SE_NB_1
☒	<50> := Set point command, short floating point value	C_SE_NC_1
☒	<51> := Bitstring of 32 bits	C_BO_NA_1
☒	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
☒	<59> := Double commandwith time tag CP56Time2a	C_DC_TA_1
☐	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☒	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
☒	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

Table 3: System information in monitoring direction

☒	<70> := End of initialization	M_EI_NA_1

Table 4: System information in control direction

☒	<100> := Interrogation command	C_IC_NA_1
☒	<101> := Counter interrogation command	C_CI_NA_1
☐	<102> := Read command	C_RD_NA_1
☒	<103> := Clock synchronization command	C_CS_NA_1
☒	<104> := Test command	C_TS_NA_1
☐	<105> := Reset process command	C_RP_NA_1
☐	<106> := Delay acquisition command	C_CD_NA_1
☐	<107> := Test command with time tagCP56Time2a	C_TS_TA_1

Table 5: Parameters in control direction

☐	<110> := Parameter of measured values, normalized value	P_ME_NA_1
☐	<111> := Parameter of measured values, scaled value	P_ME_NB_1
☐	<112> := Parameter of measured values, short floating point value	P_ME_NC_1
☐	<113>:= Parameter activation	P_AC_NA_1

Table 6: File tranfer

☐	<120>:= File ready	F_FR_NA_1
☐	<121>:= Section ready	F_SR_NA_1
☐	<122> := Call directory, select file, call file, call section	F_SC_NA_1
☐	<123> := Last section, last segment	F_LS_NA_1
☐	<124> := Ack file, ack section	F_AF_NA_1
☐	<125>:= Segment	F_SG_NA_1
☐	<126> := Directory (blank or X, available only in monitor [standard] direction)	F_DR_TA_1

Type identifier and cause of transmission asignments (station-specific parameters)

The combination of type identifier / cause of transmission is marked as:

Shaded	Function or ASDU are not required.
Empty	Function or ASDU is not used.
X	Only used in standard direction
R	Only used in reverse direction
B	Used in both directions

Table 7: Cause of transmission

Number	Cause of transmission
<0>	Not used
<1>	Periodic, cyclic
<2>	Background scan
<3>	Spontaneous
<4>	Initialized
<5>	Request or requested
<6>	Activation (ACT)
<7>	Activation confirmation (ACTCON)
<8>	Deactivation (DEACT)
<9>	Deactivation confirmation (DEACTCON)
<10>	Activation termination (ACTTERM)
<11>	Return information caused by a remote command
<12>	Return information caused by a local command
<13>	File transfer
<20..36>	General interrogation
<37..41>	Counter interrogation
<44>	Unknown type identification
<45>	Unknown cause of transmission
<46>	Unknown address of ASDU
<47>	Unknown information object address

Type identification		Cause of transmission																	
		<1>	<2>	<3>	<4>	<5>	<6>	<7>	<8>	<9>	<10>	<11>	<12>	<13>	<20..36>	<37..41>	<44>	<45>	<46>	<47>
		per/cyc	back	spont	init	req	act	actcon	deact	deactcon	actterm	retrem	retloc	file	inro	reqco	unknowntype	unknownCOT	unknownCAA	unknownIOA
<1>	M_SP_NA_1		X	X								X			X					
<2>	M_SP_TA_1			X								X								
<3>	M_DP_NA_1		X	X								X			X					
<4>	M_DP_TA_1			X								X								
<5>	M_ST_NA_1		X	X								X			X					
<6>	M_ST_TA_1			X																
<7>	M_BO_NA_1		X	X											X					
<8>	M_BO_TA_1			X																
<9>	M_ME_NA_1	X	X	X											X					
<10>	M_ME_TA_1			X																
<11>	M_ME_NB_1	X	X	X											X					
<12>	M_ME_TB_1			X																
<13>	M_ME_NC_1	X	X	X											X					
<14>	M_ME_TC_1			X																
<15>	M_IT_NA_1			X												X				
<16>	M_IT_TA_1			X												X				
<17>	M_EP_TA_1			X																
<18>	M_EP_TB_1			X																

Type identification		Cause of transmission																	
<19>	M_EP_TC_1			X															
<20>	M_PS_NA_1		X	X								X			X				
<21>	M_ME_ND_1	X	X	X											X				
<30>	M_SP_TB_1			X								X							
<31>	M_DP_TB_1			X								X							
<32>	M_ST_TB_1			X								X							
<33>	M_BO_TB_1			X															
<34>	M_ME_TD_1			X															
<35>	M_ME_TE_1			X															
<36>	M_ME_TF_1			X															
<37>	M_IT_TB_1			X												X			
<38>	M_EP_TD_1			X															
<39>	M_EP_TE_1			X															
<40>	M_EP_TF_1			X															
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1																			
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X	X	X	X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X

Type identification		Cause of transmission																	
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1																			
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X	X	X	X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1																			
<103>	C_CS_NA_1			X			X	X									X	X	X	X
<104>	C_TS_NA_1						X	X									X	X	X	X
<105>	C_RP_NA_1																			
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1																			
<110>	P_ME_NA_1																X	X	X	X
<111>	P_ME_NB_1																X	X	X	X
<112>	P_ME_NC_1																X	X	X	X
<113>	P_AC_NA_1																X	X	X	X
<120>	F_FR_NA_1																X	X	X	X
<121>	F_SR_NA_1																X	X	X	X
<122>	F_SC_NA_1																X	X	X	X

Type identification		Cause of transmission																	
<123>	F_LS_NA_1															X	X	X	X
<124>	F_AF_NA_1															X	X	X	X
<125>	F_SG_NA_1															X	X	X	X
<126>	F_DR_TA_1																		

Basic application functions

Station initilization

☒ Remote initialization

Table 8: Cyclic data transmission

☒ Cyclic data transmission

Table 9: Read procedure

☐ Read procedure

Table 10: Spontaneous transmission

☒ Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project specific list.

☐ Single point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 und M_SP_NA_1

☐ Double point information M_DP_NA_1, M_DP_TA_1 und M_DP_TB_1

☐ Step position information M_ST_NA_1, M_ST_TA_1 und M_ST_TB_1

☐ Bitstring of 32 bits M_BO_NA_1, M_BO_TA_1 und M_BO_TB_1

☐ Measured value, normalized valueM_ME_NA_1, M_ME_TA_1, M_ME_ND_1 und M_ME_TD_1

☐ Measured value, scaled valueM_ME_NB_1, M_ME_TB_1 und M_ME_TE_1

☐ Measured value, short floating point numberM_ME_NC_1, M_ME_TC_1 und M_ME_TF_1

Table 11: Station interrogation

☒	Global
☒	Group 1
☒	Group 2
☒	Group 3
☒	Group 4
☒	Group 5
☒	Group 6
☒	Group 7
☒	Group 8
☒	Group 9
☒	Group 10
☒	Group 11
☒	Group 12
☒	Group 13
☒	Group 14
☒	Group 15
☒	Group 16

The allocation of information object addresses to each individual group must be specified in a separate table.

Table 12: Clock synchronization

☒	Clock synchronization

Table 13: Command transmission

☒	Direct command transmission
☒	Direct setpoint command transmission
☐	Command .selection and execution
☐	Setpoint command .selection and execution
☒	C_SE ACTTERM used
☒	No additional definition
☐	Short pulse duration(duration determined by a system parameter in the outstation)
☐	Long pulse duration(duration determined by a system parameter in the outstation)
☐	Persistent output
☐	Supervision of maximum delay in command direction of commands and setpoint commands
	Maximum allowable delay of commands and setpoint commands

Table 14: Counter interrogation

☒	Counter read
☒	Counter freeze without reset
☒	Counter freeze with reset
☒	Counter reset
☒	General counter request
☒	Request counter group 1
☒	Request counter group 2
☒	Request counter group 3
☒	Request counter group 4
☒	Mode A:Local freeze with spontaneous transmission
☒	Mode B:Local freeze with counter interrogation
☐	Mode C:Freeze and transmit by counter-interrogation commands
☒	Modus D:Freeze by counter-interrogation command, frozen values reported simultaneously

Table 15: Parameter loading

☐	Threshold value
☐	Smoothing factor
☐	Lower limit value for transmission of measured values
☐	High limit value for transmission of measured values

Table 16: Parameter activation

☐	Activation/deactivation of cyclic or periodic transmission of the addressed object

Table 17: Test procedure

☒	Test procedure

File transfer

File transfer in monitoring direction:

☐	Transparent file
☐	Transmission of event sequences
☐	Transmission of disturbance data of protection equipment
☐	Transmission of sequences of recorded analog values

File transfer in control direction:

☐	Transparent file

Table 18: Background scan

☒	Background scan

Table 19: Telegram run time acquisition

☐	Telegram run time acquisition

Appendix

2.2.1.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

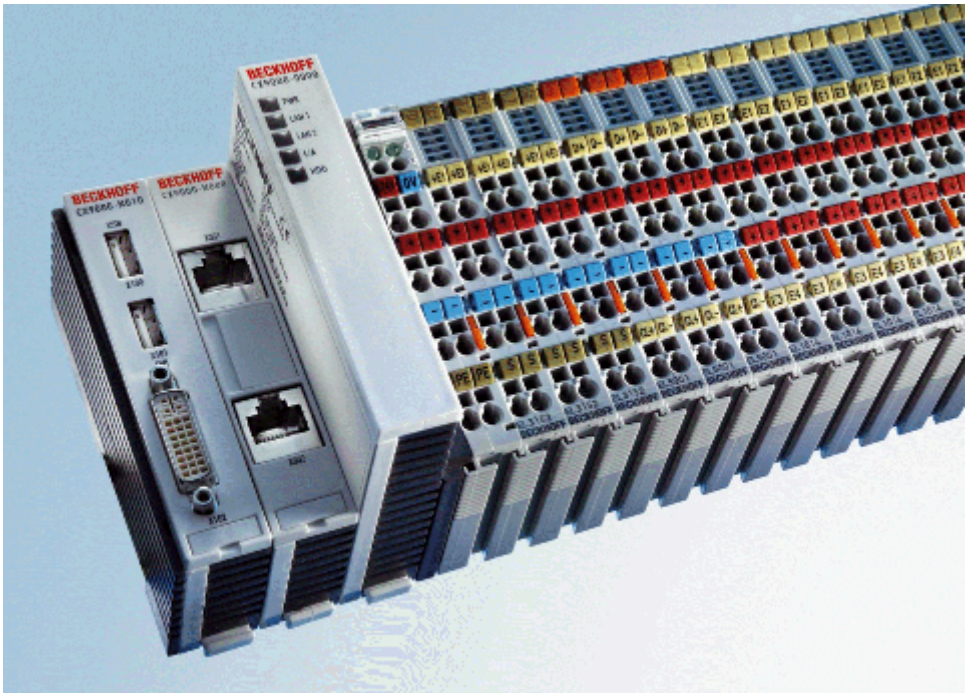
Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

2.2.2 Interoperability list according to IEC 60870-5-101 (slave)

Für die TwinCAT SPS Bibliothek: IEC 60870-5-101 Unterstation (slave).



Version: 1.3
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984586379.zip

Table of contents:

- General information [▶ 31]
  - Project information [▶ 31]
- Compatibility [▶ 31]
  - System or device [▶ 32]
  - Network configuration [▶ 32]
  - Physical layer [▶ 32]
  - Link layer [▶ 32]
  - Application layer [▶ 33]
  - Selection of standardised ASDUs [▶ 34]

Process information in monitoring direction

Process information in control direction

System information in monitoring direction

System information in control direction

Parameter in control direction

File transfer

- Basic application functions [▶ 42]

Station initialization

Cyclic data transfer

Read procedure

Spontaneous transmission

Double transmission

Station interrogation

Clock synchronisation

Command transmission

Counter interrogation

Parameter loading

Parameter activation

Test procedure

File transfer

Background scan

Telegram run time acquisition

- Appendix [▶ 45]
  - support and service [▶ 45]

General Information

This compatibility list refers to functions supported within IEC 870-5-101.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Projekt No.:
_____

Project: _____

Responsible:
_____

Date: _____

Compatibility

The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
X	Function or ASDU is used as specified (preferred option)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in regular and reverse mode

System or device

☐	System definition
☐	Specifications for the control station
☒	Specifications for the controlled station (slave)

Network configuration

☒	Point-to-point	☒	Multipoint-partyline
☒	Multiple point-to-point	☐	Multipoint-star

Physical layer

Transmission speed:

☐	100 bps	☒	2400 bps	☐	56000 bps
☐	200 bps	☒	4800 bps	☐	64000 bps
☒	300 bps	☒	9600 bps		
☒	600 bps	☒	19200 bps		
☒	1200 bps	☐	38400 bps		

Link layer

According to this application-related standard, only telegram format FT 1.2, single character 1 and the time monitoring interval are used.

Address field of the link layer:

☒	Not available (only balanced transfer)
☒	One octet
☒	Two octets
☒	Structured
☒	Non structured

Link layer transfer procedure:

☒	Balanced transfer
☒	Unbalanced transfer

Telegram length:

Maximum length L	255

In unbalanced mode some ASDU types are send with low (class 2) priority to the control station.

☒	Normed asssignment of ASDU type to class 2 data

Type identifier	Cause of transmission
9, 11, 13, 21	<1>
1, 3, 5, 7, 9, 11, 13, 20, 21	<2>

☐ Special assignment of ASDU type to class 2 data

Type identifier	Cause of transmission

An outstation may return class 1 data when polled for class 2 data if there is no class 2 data available (configuration parameter).

Application layer

Tranfer mode for application data

According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

Common ASDU address

☒	One octet	☒	Two octets

Address of the information object

☒	One octet	☒	Structured
☒	Two octets	☒	Non structured
☒	Three octets		

Cause of transmission

☒	One octet	☒	Two octets (with origin address). Preset to 0, if origin address is not available.

Selection of standardised ASDU's

Table 20: Process information in monitoring direction

☒	<1> :=Single-point information	M_SP_NA_1
☒	<2> := Single-point information with time tag	M_SP_TA_1
☒	<3> := Double point information	M_DP_NA_1
☒	<4> := Double point information with time tag	M_DP_TA_1
☒	<5> := Step position information	M_ST_NA_1
☒	<6> := Step position information with time tag	M_ST_TA_1
☒	<7> := Bitstring of 32 bits	M_BO_NA_1
☒	<7> := Bitstring of 32 bits with time tag	M_BO_TA_1
☒	<9> := Measured value, normalized	M_ME_NA_1
☒	<10> := Measured value, normalized value with time tag	M_ME_TA_1
☒	<11> := Measured value, scaled value	M_ME_NB_1
☒	<12> := Measured value, scaled value with time tag	M_ME_TB_1
☒	<13> := Measured value, short floating point value	M_ME_NC_1
☒	<14> := Measured value, short floating point value with time	M_ME_TC_1
☒	<15> := Integrated totals	M_IT_NA_1
☒	<16> := Integrated totals with time tag	M_IT_TA_1
☒	<17> := Event of protection equipment with time tag	M_EP_TA_1
☒	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
☒	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☒	<20> := Packed single-point information with status change detection	M_PS_NA_1
☒	<21> := Measured value, normalized value without quantity descriptor	M_ME_ND_1
☒	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
☒	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
☒	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
☒	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
☒	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
☒	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☒	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1

☒	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Table 21: Process information in control direction

☒	<45> := Single command	C_SC_NA_1
☒	<46> := Double command	C_DC_NA_1
☒	<47> := Regulating step command	C_RC_NA_1
☒	<48> := Set point command, normalized value	C_SE_NA_1
☒	<49> := Set point command, scaled value	C_SE_NB_1
☒	<50> := Set point command, short floating point value	C_SE_NC_1
☒	<51> := Bitstring of 32 bits	C_BO_NA_1
☒	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
☒	<59> := Double commandwith time tag CP56Time2a	C_DC_TA_1
☒	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☒	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
☒	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

Table 22: System information in monitoring direction

☒	<70> := End of initialization	M_EI_NA_1

Table 23: System information in control direction

☒	<100> := Interrogation command	C_IC_NA_1
☒	<101> := Counter interrogation command	C_CI_NA_1
☒	<102> := Read command	C_RD_NA_1
☒	<103> := Clock synchronization command	C_CS_NA_1
☒	<104> := Test command	C_TS_NA_1
☒	<105> := Reset process command	C_RP_NA_1
☐	<106> := Delay acquisition command	C_CD_NA_1
☒	<107> := Test command with time tagCP56Time2a	C_TS_TA_1

Table 24: Parameters in control direction

☐	<110> := Parameter of measured values, normalized value	P_ME_NA_1
☐	<111> := Parameter of measured values, scaled value	P_ME_NB_1
☐	<112> := Parameter of measured values, short floating point value	P_ME_NC_1
☐	<113>:= Parameter activation	P_AC_NA_1

Table 25: File tranfer

☐	<120>:= File ready	F_FR_NA_1
☐	<121>:= Section ready	F_SR_NA_1
☐	<122> := Call directory, select file, call file, call section	F_SC_NA_1
☐	<123> := Last section, last segment	F_LS_NA_1
☐	<124> := Ack file, ack section	F_AF_NA_1
☐	<125>:= Segment	F_SG_NA_1
☐	<126> := Directory (blank or X, available only in monitor [standard] direction)	F_DR_TA_1

Type identifier and cause of transmission asignments (station-specific parameters)

The combination of type identifier / cause of transmission is marked as:

Shaded	Function or ASDU are not required.
Empty	Function or ASDU is not used.
X	Only used in standard direction
R	Only used in reverse direction
B	Used in both directions

Table 26: Cause of transmission

Number	Cause of transmission
<0>	Not used
<1>	Periodic, cyclic
<2>	Background scan
<3>	Spontaneous
<4>	Initialized
<5>	Request or requested
<6>	Activation (ACT)
<7>	Activation confirmation (ACTCON)
<8>	Deactivation (DEACT)
<9>	Deactivation confirmation (DEACTCON)
<10>	Activation termination (ACTTERM)
<11>	Return information caused by a remote command
<12>	Return information caused by a local command
<13>	File transfer
<20..36>	General interrogation
<37..41>	Counter interrogation
<44>	Unknown type identification
<45>	Unknown cause of transmission
<46>	Unknown address of ASDU
<47>	Unknown information object address

Type identification		Cause of transmission																	
		<1>	<2>	<3>	<4>	<5>	<6>	<7>	<8>	<9>	<10>	<11>	<12>	<13>	<20..36>	<37..41>	<44>	<45>	<46>	<47>
		per/cyc	back	spont	init	req	act	actcon	deact	deactcon	actterm	retrem	retloc	file	inro	reqco	unknowntype	unknownCOT	unknownCAA	unknownIOA
<1>	M_SP_NA_1		X	X		X						X			X					
<2>	M_SP_TA_1			X		X						X								
<3>	M_DP_NA_1		X	X		X						X			X					
<4>	M_DP_TA_1			X		X						X								
<5>	M_ST_NA_1		X	X		X						X			X					
<6>	M_ST_TA_1			X		X														
<7>	M_BO_NA_1		X	X		X									X					
<8>	M_BO_TA_1			X		X														
<9>	M_ME_NA_1	X	X	X		X									X					
<10>	M_ME_TA_1			X		X														
<11>	M_ME_NB_1	X	X	X		X									X					
<12>	M_ME_TB_1			X		X														
<13>	M_ME_NC_1	X	X	X		X									X					
<14>	M_ME_TC_1			X		X														
<15>	M_IT_NA_1			X												X				
<16>	M_IT_TA_1			X												X				
<17>	M_EP_TA_1			X																
<18>	M_EP_TB_1			X																

Type identification		Cause of transmission																	
<19>	M_EP_TC_1			X															
<20>	M_PS_NA_1		X	X		X						X			X				
<21>	M_ME_ND_1	X	X	X		X									X				
<30>	M_SP_TB_1			X		X						X							
<31>	M_DP_TB_1			X		X						X							
<32>	M_ST_TB_1			X		X						X							
<33>	M_BO_TB_1			X		X													
<34>	M_ME_TD_1			X		X													
<35>	M_ME_TE_1			X		X													
<36>	M_ME_TF_1			X															
<37>	M_IT_TB_1			X												X			
<38>	M_EP_TD_1			X															
<39>	M_EP_TE_1			X															
<40>	M_EP_TF_1			X															
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1						X	X	X	X	X						X	X	X	X
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X	X	X	X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X

Type identification		Cause of transmission																	
<59>	C_DCT A_1						X	X	X	X	X						X	X	X	X
<60>	C_RCT A_1						X	X	X	X	X						X	X	X	X
<61>	C_SET A_1						X	X	X	X	X						X	X	X	X
<62>	C_SET B_1						X	X	X	X	X						X	X	X	X
<63>	C_SET C_1						X	X	X	X	X						X	X	X	X
<64>	C_BOT A_1						X	X	X	X	X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X											X	X	X	X
<103>	C_CS_NA_1			X			X	X									X	X	X	X
<104>	C_TS_NA_1						X	X									X	X	X	X
<105>	C_RP_NA_1						X	X									X	X	X	X
<106>	C_CD_NA_1																X	X	X	X
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1																X	X	X	X
<111>	P_ME_NB_1																X	X	X	X
<112>	P_ME_NC_1																X	X	X	X
<113>	P_AC_NA_1																X	X	X	X
<120>	F_FR_NA_1																X	X	X	X
<121>	F_SR_NA_1																X	X	X	X
<122>	F_SC_NA_1																X	X	X	X

Type identification		Cause of transmission																	
<123>	F_LS_NA_1															X	X	X	X
<124>	F_AF_NA_1															X	X	X	X
<125>	F_SG_NA_1															X	X	X	X
<126>	F_DR_TA_1																		

Basic application functions

Station initilization

☒ Remote initialzation

Table 27: Cyclic data transmission

☒ Cyclic data transmission

Table 28: Read procedure

☒ Read procedure

Table 29: Spontaneous transmission

☒ Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project specific list.

☐ Single point informaion M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 und M_SP_NA_1

☐ Double point information M_DP_NA_1, M_DP_TA_1 und M_DP_TB_1

☐ Step position information M_ST_NA_1, M_ST_TA_1 und M_ST_TB_1

☐ Bitstring of 32 bits M_BO_NA_1, M_BO_TA_1 und M_BO_TB_1

☐ Measured value, normalized valueM_ME_NA_1, M_ME_TA_1, M_ME_ND_1 und M_ME_TD_1

☐ Measured value, scaled valueM_ME_NB_1, M_ME_TB_1 und M_ME_TE_1

☐ Measured value, short floating point numberM_ME_NC_1, M_ME_TC_1 und M_ME_TF_1

Table 30: Station interrogation

☒	Global
☒	Group 1
☒	Group 2
☒	Group 3
☒	Group 4
☒	Group 5
☒	Group 6
☒	Group 7
☒	Group 8
☒	Group 9
☒	Group 10
☒	Group 11
☒	Group 12
☒	Group 13
☒	Group 14
☒	Group 15
☒	Group 16

The allocation of information object addresses to each individual group must be specified in a separate table.

Table 31: Clock synchronization

☒	Clock synchronization

Table 32: Command transmission

☒	Direct command transmission
☒	Direct setpoint command transmission
☒	Command .selection and execution
☒	Setpoint command .selection and execution
☒	C_SE ACTTERM used
☒	No additional definition
☐	Short pulse duration(duration determined by a system parameter in the outstation)
☐	Long pulse duration(duration determined by a system parameter in the outstation)
☐	Persistent output
☐	Supervision of maximum delay in command direction of commands and setpoint commands
	Maximum allowable delay of commands and setpoint commands

Table 33: Counter interrogation

☒	Counter read
☒	Counter freeze without reset
☒	Counter freeze with reset
☒	Counter reset
☒	General counter request
☒	Request counter group 1
☒	Request counter group 2
☒	Request counter group 3
☒	Request counter group 4
☒	Mode A:Local freeze with spontaneous transmission
☒	Mode B:Local freeze with counter interrogation
☒	Mode C:Freeze and transmit by counter-interrogation commands
☒	Modus D:Freeze by counter-interrogation command, frozen values reported simultaneously

Table 34: Parameter loading

☐	Threshold value
☐	Smoothing factor
☐	Lower limit value for transmission of measured values
☐	High limit value for transmission of measured values

Table 35: Parameter activation

☐	Activation/deactivation of cyclic or periodic transmission of the addressed object

Table 36: Test procedure

☒	Test procedure

File transfer

File transfer in monitoring direction:

☐	Transparent file
☐	Transmission of event sequences
☐	Transmission ofdisturbance data of protection equipment
☐	Transmissionof sequences of recorded analog values

File transfer in control direction:

☐	Transparent file

Table 37: Background scan

☒	Background scan

Table 38: Telegram run time acquisition

☐	Telegram run time acquisition

Appendix

2.2.2.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

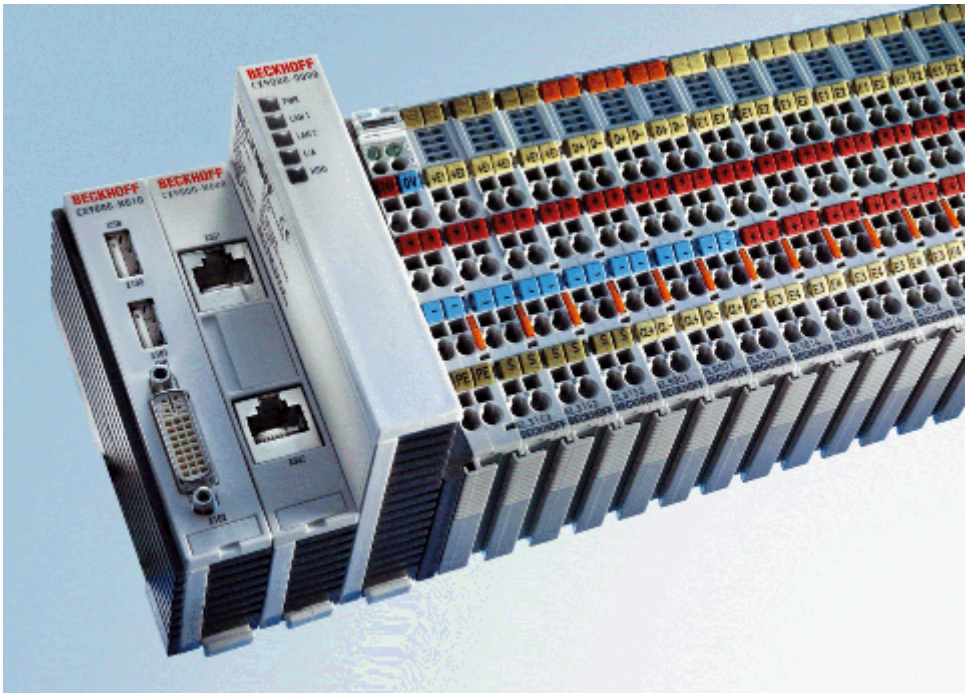
Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

2.2.3 Interoperability list according to IEC 60870-5-102 (master)

für TwinCAT SPS Bibliothek: IEC 60870-5-102 Zentralstation (master).



Version: 1.2
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984589323.zip

Table of contents:

- General information [▶ 48]
  - Project information [▶ 48]
- Compatibility [▶ 48]
  - System or device [▶ 48]
  - Network configuration [▶ 48]
  - Physical layer [▶ 48]
  - Link layer [▶ 49]
  - Application layer [▶ 49]
  - Selection of standardized ASDUs [▶ 51]

Process information in monitoring direction (slave->master)

System information in monitoring direction (slave->master)

System information in control direction (master->slave)

- Basic application functions [▶ 54]
  - Transmission of integrated totals [▶ 54]
  - Read of list [▶ 54]
  - Read of single values [▶ 54]
- Appendix [▶ 54]

- support and service [▶ 54]

General Information

This compatibility list refers to functions supported within IEC 870-5-102.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Project No.: _____

Project: _____

Responsible: _____

Date: _____

Compatibility

Octets
Subsets have to be chosen from the preceding parameter sets and alternatives, to create defined systems of transmission of counter readings.
Certain parameter like the number of octets with in the COMMON ADDRESS OF THE ASDU, exclude themselves reciprocally.
This means only one value of the specified parameter is permitted per system. Other parameters like a listed sets of different information in controller and monitoring devices enable the definition of the whole set or of subset that are suitable for the given application.

Octets
The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as specified (preferred option)

System or device

☐	System definition
☒	Specifications for the control station
☐	Specifications for the controlled station

Network configuration

☒	Point-to-point	☒	Multipoint-partyline
☒	Multiple point-to-point	☐	Multipoint-star

Physical layer

Transmission speed:

☐	100 bps	☒	2400 bps	☐	56000 bps
☐	200 bps	☒	4800 bps	☐	64000 bps
☒	300 bps	☒	9600 bps		
☒	600 bps	☒	19200 bps		
☒	1200 bps	☐	38400 bps		

Link layer

Octets
According to this application-related standard, only telegram format FT 1.2, single character 1 and the time monitoring interval are used.

Address field of the link layer:

☒	Not available (only symmetric transfer)
☒	One octet
☒	Two octets
☒	Structured
☒	Non structured

Octets
Link layer transfer procedure:

☐	Symmetric transfer
☒	Asymmetric transfer

Octets
Telegram length:

Maximum length L:	255

Octets
Link layer services:

☐	Reset of user process is not available

Application layer

Transfer mode for application data

Octets
According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

Manufacturing code

(Vendor specific parameter)

Number	Vendor
1	
2	
...	
255	

Address of the meter DTE (common ASDU address)

(System parameter)

☒	One octet	☒	Two octets

Signature

(System parameter)

☐	Signature	☐	No signature

Record address

(System parameter)

☐	General	
☐	Period 1	Cycle time:
☐	Period 2	Cycle time:
☐	Period 3	Cycle time:

☐	Daily values	☐	Month values	☐	Annual values
☐	Period 1	☐	Period 1	☐	Period 1
☐	Period 2	☐	Period 2	☐	Period 2
☐	Period 3	☐	Period 3	☐	Period 3

☐	Counter readings from the beginning of the account period
☐	monthly account period
☐	yearly account period
☐	quarterly account period (every three months)
☐	Oldest single indication
☐	Complete list with single indications
☐	Partial list 1 with single indications
☐	Partial list 2 with single indications
☐	Partial list 3 with single indications
☐	Partial list 4 with single indications

Octets
rate information

Octets
(System parameter)

☐	rate information	☐	no rate information

Selection of standardized ASDUs

Table 39: Process information in monitoring direction

☒	<1> := Single point information with time tag	M_SP_TA_2
☒	<2> := Accounting integrated totals, four octets each	M_IT_TA_2
☒	<3> := Accounting integrated totals, three octets each	M_IT_TB_2
☒	<4> := Accounting integrated totals, two octets each	M_IT_TC_2
☒	<5> := Periodically reset accounting integrated totals, four octets each	M_IT_TD_2
☒	<6> := Periodically reset accounting integrated totals, three octets each	M_IT_TE_2
☒	<7> := Periodically reset accounting integrated totals, two octets each	M_IT_TF_2
☒	<8> := Operational integrated totals, four octets each	M_IT_TG_2
☒	<9> := Operational integrated totals, three octets each	M_IT_TH_2
☒	<10> := Operational integrated totals, two octets each	M_IT_TI_2
☒	<11> := Periodically reset operational integrated totals, four octets each	M_IT_TK_2
☒	<12> := Periodically reset operational integrated totals, three octets each	M_IT_TL_2
☒	<13> := Periodically reset operational integrated totals, two octets each	M_IT_TM_2

Table 40: System information in control direction

☒	<70> := End of initialization	M_EI_NA_2
☒	<71> := Manufacturer and product specification of integrated total DTE	P_MP_NA_2
☒	<72> := Current system time of integrated total DTE	M_TI_TA_2

Table 41: System information in control direction

☒	<100>:= Read manufacturer and product specification	C_RD_NA_2
☒	<101>:= Read record of single -point information with time tag	C_SP_NA_2
☒	<102>:= Read record of single-point information with time tag of selected time range	C_SP_NB_2
☒	<103>:= Read current system time of integrated total DTE	C_TI_NA_2
☒	<104>:= Read accounting integrated totals of the oldest integration period	C_CI_NA_2
☒	<105>:= Read accounting integrated totals of the oldest integration period and of a selected range of addresses	C_CI_NB_2
☒	<106>:= Read accounting integrated totals of a specific past integration period	C_CI_NC_2
☒	<107>:= Read accounting integrated totals of a specific past integration period and of a selected range of addresses	C_CI_ND_2
☒	<108>:= Read periodically reset accounting integrated totals of the oldest integration period	C_CI_NE_2
☒	<109>:= Read periodically reset accounting integrated totals of the oldest integration period and of a selected range of addresses	C_CI_NF_2
☒	<110>:= Read periodically reset accounting integrated totals of a specific past integration period	C_CI_NG_2
☒	<111>:= Read periodically reset accounting integrated totals of a specific past integration period and of a selected range of addresses	C_CI_NH_2
☒	<112>:= Read operational integrated totals of the oldest integration period	C_CI_NI_2
☒	<113>:= Read operational integrated totals of the oldest integration period and of a selected range of addresses	C_CI_NK_2
☒	<114>:= Read operational integrated totals of a specific past integration period	C_CI_NL_2
☒	<115>:= Read operational integrated totals of a specific past integration period and of a selected range of addresses	C_CI_NM_2
☒	<116>:= Read periodically reset operational integrated totals of the oldest integration period	C_CI_NN_2
☒	<117>:= Read periodically reset operational integrated totals of the oldest integration period and of a selected range of addresses	C_CI_NO_2
☒	<118>:= Read periodically reset operational integrated totals of a specific past integration period	C_CI_NP_2
☒	<119>:= Read periodically reset operational integrated totals of a specific past integration period and of a selected range of addresses	C_CI_NQ_2
☒	<120>:= Read accounting integrated totals of a selected time range and of a selected range of addresses	C_CI_NR_2
☒	<121>:= Read periodically reset accounting integrated totals of a selected time range and of a selected range of addresses	C_CI_NS_2
☒	<122>:= Read operational integrated totals of a selected time range and of a selected range of addresses	C_CI_NT_2

☒	<123>:= Read periodically reset operational integrated totals of a selected time range and of a selected range of addresses	C_CI_NU_2

Basic application function

Octets Transmission of counter readings

Octets
(Station dependent parameter)

Octets
Number of billing counter readings:
Octets
Number of periodically resetted billing counter readings:
Octets
Number of operation counter readings:
Octets
Number of periodically resetted operation counter readings:

☐ Octets
counter readings are transmitted only by call functions (including the latest period)

Octets Calling of lists

Octets
(Station dependent parameter)

Octets
List volume:

Octets Complete list	☐	Octets number of single indications:
Octets Partial list 1	☐	Octets number of single indications:
Octets Partial list 2	☐	Octets number of single indications:
Octets Partial list 3	☐	Octets number of single indications:
Octets Partial list 4	☐	Octets number of single indications:

☐ Octets
No list, single indications are transmitted spontaneously

Octets Transmission of single indications

Octets
(Station dependent parameter)

☐ Octets
Transmission of locally acknowledged single indications

☐ Octets
No transmission of locally acknowledged single indications

Appendix

Also see about this

- Interoperability list according to IEC 60870-5-102 (master) [▶ 54]

2.2.3.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

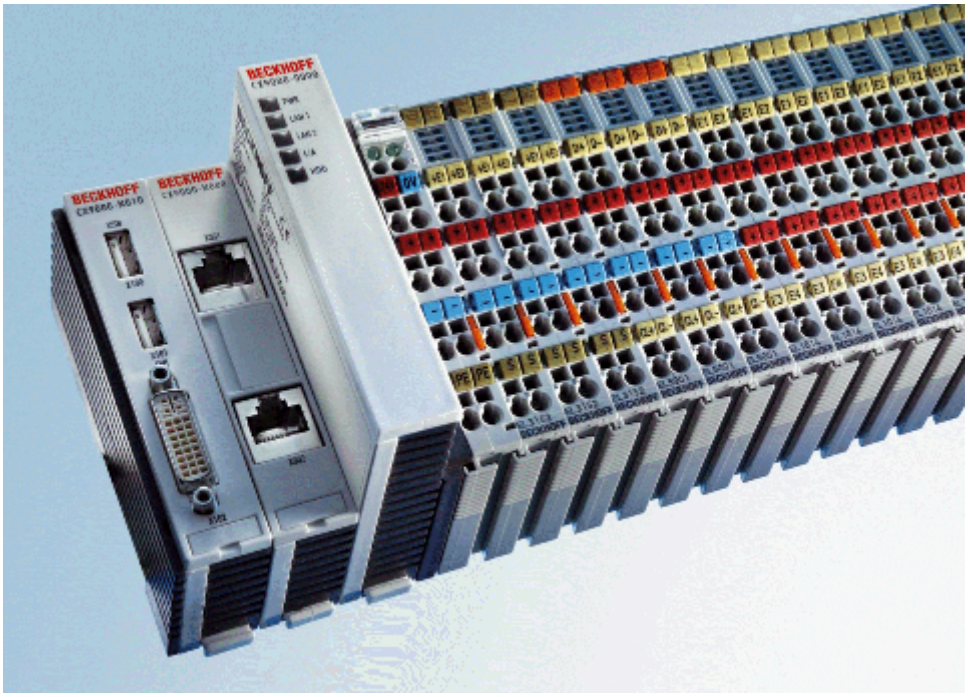
Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

2.2.4 Interoperability list according to IEC 60870-5-103 (master)

für TwinCAT SPS Bibliothek: IEC 60870-5-103 Zentralstation (master).



Version: 1.2
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984592267.zip

Table of contents:

- General information [▶ 57]
  - Project information [▶ 57]
- Compatibility [▶ 57]
  - System or device [▶ 57]
  - Physical layer [▶ 57]

Electrical interface

Optical interface

Transfer velocity

- Link layer
- Application layer

Transfer modus for application data

Common address of ASDU

Selection of standardised ASDUs in monitoring direction

Selection of standardised ASDUs in control direction

Selection of standardised information numbers in monitoring direction

Selection of standardised information numbers in control direction

Basic application functions [▶ 64]

Miscellaneous

- Appendix [▶ 65]
  - support and service [▶ 65]

General Information

This compatibility list refers to functions supported within IEC 870-5-103.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Projekt No.:

_____

Project: _____

Responsible:

_____

Date: _____

Compatibility

The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as specified (preferred option)

System or device

☐	System definition
☒	Specifications for the control station
☐	Specifications for the controlled station

Physical layer

☐		EIA RS-485	
☐	Count of load	.....	for one protection direction

EIA RS-485 defines the unit loads in that way, that 32 of them can be operated at one line.

Single information, see EIA RS 485, article 3

☐	Fibre optics
☐	plactic fibre
☐	F-SMA plug
☐	BFOC/2,5 plug
☒	9600 bit/s
☒	19200 bit/s

Link layer

No options for the link layer.

Application layer

Transfer mode for application data

According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

☒	One COMMON ADDRESS of ASDU	☒	Identical with station address	☒	Not identical with station address		
☐	More than one COMMON ADDRESS of ASDU						

	<0> := not used	-
☒	<1> :=Time-tagged message	M_TTM_TA_3
☒	<2> := Time-tagged message with relative time	M_TMR_TA_3
☒	<3> := Measurands I	M_MEI_NA_3
☒	<4> := Time-tagged measurands with relative time	M_TME_TA_3
☒	<5> := Identification	M_IRC_NA_3
☒	<6> := Time synchronisation	M_SYN_TA_3
☒	<8> := Termination of general interrogation	M_TGI_NA_3
☒	<9> := Measurands II	M_MEII_NA_3
☒	<10> := Generic data	M_GD_XA_3
☒	<11> := Generic identification	M_GI_XA_3
☒	<23> := List of recorded disturbances	M_LRD_TA_3
☒	<26> := Ready for transmission of disturbance data	M_RTD_TA_3
☒	<27> := Ready for transmission of channel	M_RTC_NA_3
☒	<28> := Ready for transmission of tags	M_RTT_NA_3
☒	<29> := Transmission of tags	M_TOT_NA_3
☒	<30> := Transmission of disturbance values	M_TOV_NA_3
☒	<31> := End of transmission	M_EOT_NA_3

	<0> := not used	-
☒	<6> := Time synchronisation	C_SYN_TA_3
☒	<7> := General interrogation	C_IGI_NA_3
☒	<10> := Generic data	C_GD_NA_3
☒	<20> := General command	C_GRC_NA_3
☒	<21> := Generic command	C_GC_NA_3
☒	<24> := Order for disturbance data transmission	C_ODT_NA_3
☒	<25> := Acknowledgement for disturbance data transmission	C_ADT_NA_3

Selection of standardised ASDUs in monitoring direction

System functions in monitoring direction

	INF	Description
☐	<0>	End of general interrogation
☐	<1>	Time synchronisation
☐	<2>	Reset FCB
☐	<3>	Reset KE
☐	<4>	Start / Restart
☐	<5>	First start

Items describing the status in monitoring direction

	INF	Description
☐	<16>	Automatic restart active
☐	<17>	Protection signal transfer active
☐	<18>	Protection active
☐	<19>	Reset display
☐	<20>	Latch of monitoring direction
☐	<21>	Test operation
☐	<22>	Parametrisation on site
☐	<23>	Characteristic curve 1
☐	<24>	Characteristic curve 2
☐	<25>	Characteristic curve 3
☐	<26>	Characteristic curve 4
☐	<27>	Input 1
☐	<28>	Input 2
☐	<29>	Input 3
☐	<30>	Input 4

Monitoring messages in monitoring direction

	INF	Description
☐	<32>	Measurement monitoring I
☐	<33>	Measurement monitoring V
☐	<35>	Rotating field monitoring
☐	<36>	Trigger cycle monitoring
☐	<37>	UMZ emergency operation
☐	<38>	VT automatic case
☐	<39>	Protection signal transfer defective
☐	<46>	collective message caution
☐	<47>	collective message error

Earth leakage messages in monitoring direction

	INF	Description
☐	<48>	Earth leakage L_1
☐	<49>	Earth leakage L_2
☐	<50>	Earth leakage L_3
☐	<51>	Earth leakage forwards, i.e. line
☐	<52>	Earth leakage backwards, i.e. bus bar

Störfall messages in monitoring direction

	INF	Description
☐	<64>	Trigger L_1
☐	<65>	Trigger L_2
☐	<66>	Trigger L_3
☐	<67>	Trigger N
☐	<68>	General Trigger
☐	<69>	Trigger L_1
☐	<70>	Trigger L_2
☐	<71>	Trigger L_3
☐	<72>	Auslösung UMZ- emergency operation
☐	<73>	Error place X in Ohm
☐	<74>	Error forwards / Line
☐	<75>	Error backwards / bus bar
☐	<76>	Protection signal transfer signal sent
☐	<77>	Protection signal transfer signal received
☐	<78>	Step 1
☐	<79>	Step 2
☐	<80>	Step 3
☐	<81>	Step 4
☐	<82>	Step 5
☐	<83>	Step 6
☐	<84>	General excitation
☐	<85>	Switch failure
☐	<86>	Trigger measurement system L_1
☐	<87>	Trigger measurement system L_2
☐	<88>	Trigger measurement system L_3
☐	<89>	Trigger measurement system E
☐	<90>	Trigger I>
☐	<91>	Trigger I>>
☐	<92>	Trigger IN>
☐	<93>	Trigger IN>>

AWE messages in monitoring direction

	INF	Description
☐	<128>	LS EIN via AWE
☐	<129>	LS EIN via long term AWE
☐	<130>	AWE blocked

Operating measurement in monitoring direction

	INF	Description
☐	<144>	Operating measurement I
☐	<145>	Operating measurements I, U
☐	<146>	Operating measurements I, U, P, Q
☐	<147>	Operating measurements I_N, U_{EN}
☐	<148>	Operating measurements $I_{L1,2,3}$, $U_{L1,2,3}$, P, Q, f

Generic function in monitoring direction

	INF	Description
☐	<240>	Read description of all defined groups
☐	<241>	Read values or attributes of all entries of a group
☐	<242>	Read directory of a single entry
☐	<243>	Read value or attribute of a single entry
☐	<244>	End of general interrogation generic data
☐	<249>	Write entry (with acknowledge)
☐	<250>	Write entry (with execution)
☐	<251>	Write entry (with cancel)

Selection of standardised information numbers in control direction

System functions in control direction

	INF	Description
☐	<0>	General interrogation disgust
☐	<1>	Time synchronisation

General commands in control direction

	INF	Description
☐	<16>	Restart ON/OFF
☐	<17>	Protection signal tansfer ON/OFF
☐	<18>	Protection ON/OFF
☐	<19>	Reset displays
☐	<23>	Activate characteristic curve 1
☐	<24>	Activate characteristic curve 2
☐	<25>	Activate characteristic curve 3
☐	<26>	Activate characteristic curve 4

Generic functions in control direction

	INF	Description
☐	<240>	Read description of all defined groups
☐	<241>	Read value or attribute of all entries of one group
☐	<243>	Read directory of single entry
☐	<244>	Read value or attribute of a single entry
☐	<245>	End od general interrogation generic data
☐	<248>	Write entry
☐	<249>	Write entry (with acknowledge)
☐	<250>	Write entry (with execution)
☐	<251>	Write entry (with cancel)

Basic application functons

☐	Test run
☐	Latch of monitoring direction
☐	Disturbance data
☐	Generic services
☐	Private data

Miscellaneous

Measured values are transferred with ASDU <3> and ASDU <9>. According to 7.2.6.8 the maximum of MVAL can be the nominal value multiplied by 1,2 or 2,4 .

In ASDU <3> and ASDU <9> no different (referring) factors can be used, i.e. for each measurement value is one selection allowed.

Measurement reading	rated value x (1 or 2)	Max. MVAL rated value x (2 or 4)
Current L_1	☐	☐
Current L_2	☐	☐
Current L_3	☐	☐
Voltage L_{1-E}	☐	☐
Voltage L_{2-E}	☐	☐
Voltage L_{3-E}	☐	☐
Real power Q	☐	☐
Frequence f	☐	☐
Voltage between L_1 and L_2	☐	☐

Appendix

2.2.4.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

- on-site service

- repair service
- spare parts service
- hotline service

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

Beckhoff Headquarters

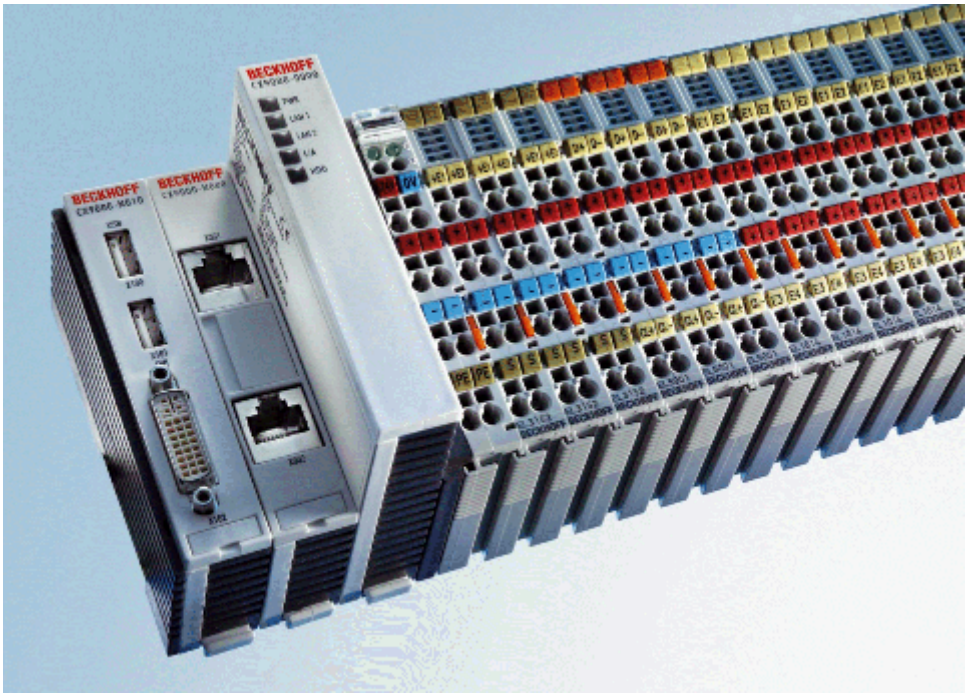
Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

2.2.5 Interoperability list according to IEC 60870-5-104 (master)

für TwinCAT SPS Bibliothek: IEC 60870-5-104 Zentralstation (master)



Version: 1.4
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984595211.zip

Table of contents:

- General information [▶ 68]
  - Project information [▶ 68]
- Compatibility [▶ 68]
  - System or device [▶ 69]
  - Network configuration [▶ 69]
  - Physical layer [▶ 69]
  - Link layer [▶ 69]
  - Application layer [▶ 69]
  - Selection of standardised ASDUs [▶ 71]

Process information in monitoring direction

Process information in control direction

System information in monitoring direction

System information in control direction

Parameter in control direction

File transfer

- Basic application functions [▶ 79]
  - Station initialization

- Cyclic data transfer
- Read procedure
- Spontaneous transmission
- Double transmission
- Station interrogation
- Clock synchronisation
- Command transmission
- Counter interrogation
- Parameter loading
- Parameter activation
- Test procedure
- File transfer
- Background scan
- Telegram run time acquisition

- Definition of time outs [▶ 82]
- k- and w-parameters
  - Port number [▶ 83]
- RFC-2200 suite [▶ 83]
- Appendix
  - support and service [▶ 83]

General Information

This compatibility list refers to functions supported within IEC 870-5-104.

⊟	Fields marked with are not used with TCP/IP coupling via IEC 870-5-104.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Projekt No.:

_____

Project: _____

Responsible:

_____

Date: _____

Compatibility

The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as specified (preferred option)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in regular and reverse mode

System or device

☐	System definition
☒	Specifications for the control station (master)
☐	Specifications for the controlled station

Network configuration

☐	Point-to-point	☐	Multipoint-partyline
☐	Multiple point-to-point	☐	Multipoint-star

Physical layer

Transmission speed:

☐	100 bps	☐	2400 bps	☐	56000 bps
☐	200 bps	☐	4800 bps	☐	64000 bps
☐	300 bps	☐	9600 bps		
☐	600 bps	☐	19200 bps		
☐	1200 bps	☐	38400 bps		

Link layer

According to this application-related standard, only telegram format FT 1.2, single character 1 and the time monitoring interval are used.

Address field of the link layer:

☐	Not available (only symmetric transfer)
☐	One octet
☐	Two octets
☐	Structured
☐	Non structured

Link layer transfer procedure:

☐	Symmetric transfer
☐	Asymmetric transfer

Telegram length:

☐	Maximum length L

Application layer

Tranfer mode for application data

According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

Common ASDU address

☐	One octet	☒	Two octets

Address of the information object

☐	One octet	☒	Structured
☐	Two octets	☒	Non structured
☒	Three octets		

Cause of transmission

☐	One octet	☒	Two octets (with origin address). Preset to 0, if origin address is not available.

Length of the *APDU*

(system-specific parameter, the maximum length of the *APDU* must be specified for each system)

The maximum length of the *APDU* is 253 (unless specified otherwise). The maximum length may be reduced, depending on the system.

253	Maximum length of the *APDU* for each system

Selection of standardised ASDU's

Table 42: Process information in monitoring direction

☒	<1> :=Single-point information	M_SP_NA_1
☐	<2> := Single-point information with time tag	M_SP_TA_1
☒	<3> := Double point information	M_DP_NA_1
☐	<4> := Double point information with time tag	M_DP_TA_1
☒	<5> := Step position information	M_ST_NA_1
☐	<6> := Step position information with time tag	M_ST_TA_1
☒	<7> := Bitstring of 32 bits	M_BO_NA_1
☐	<7> := Bitstring of 32 bits with time tag	M_BO_TA_1
☒	<9> := Measured value, normalized	M_ME_NA_1
☐	<10> := Measured value, normalized value with time tag	M_ME_TA_1
☒	<11> := Measured value, scaled value	M_ME_NB_1
☐	<12> := Measured value, scaled value with time tag	M_ME_TB_1
☒	<13> := Measured value, short floating point value	M_ME_NC_1
☐	<14> := Measured value, short floating point value with time	M_ME_TC_1
☒	<15> := Integrated totals	M_IT_NA_1
☐	<16> := Integrated totals with time tag	M_IT_TA_1
☐	<17> := Event of protection equipment with time tag	M_EP_TA_1
☐	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
☐	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☒	<20> := Packed single-point information with status change detection	M_PS_NA_1
☒	<21> := Measured value, normalized value without quantity descriptor	M_ME_ND_1
☒	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
☒	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
☒	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
☒	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
☒	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
☒	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☒	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1

☒	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Table 43: Process information in control direction

☒	<45> := Single command	C_SC_NA_1
☒	<46> := Double command	C_DC_NA_1
☐	<47> := Regulating step command	C_RC_NA_1
☒	<48> := Set point command, normalized value	C_SE_NA_1
☒	<49> := Set point command, scaled value	C_SE_NB_1
☒	<50> := Set point command, short floating point value	C_SE_NC_1
☒	<51> := Bitstring of 32 bits	C_BO_NA_1
☒	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
☒	<59> := Double commandwith time tag CP56Time2a	C_DC_TA_1
☐	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☒	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
☒	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

Table 44: System information in monitoring direction

☒	<70> := End of initialization	M_EI_NA_1

Table 45: System information in control direction

☒	<100> := Interrogation command	C_IC_NA_1
☒	<101> := Counter interrogation command	C_CI_NA_1
☐	<102> := Read command	C_RD_NA_1
☒	<103> := Clock synchronization command	C_CS_NA_1
▬	<104> := Test command	C_TS_NA_1
☐	<105> := Reset process command	C_RP_NA_1
▬	<106> := Delay acquisition command	C_CD_NA_1
☒	<107> := Test command with time tagCP56Time2a	C_TS_TA_1

Table 46: Parameters in control direction

☐	<110> := Parameter of measured values, normalized value	P_ME_NA_1
☐	<111> := Parameter of measured values, scaled value	P_ME_NB_1
☐	<112> := Parameter of measured values, short floating point value	P_ME_NC_1
☐	<113>:= Parameter activation	P_AC_NA_1

Table 47: File tranfer

☐	<120>:= File ready	F_FR_NA_1
☐	<121>:= Section ready	F_SR_NA_1
☐	<122> := Call directory, select file, call file, call section	F_SC_NA_1
☐	<123> := Last section, last segment	F_LS_NA_1
☐	<124> := Ack file, ack section	F_AF_NA_1
☐	<125>:= Segment	F_SG_NA_1
☐	<126> := Directory (blank or X, available only in monitor [standard] direction)	F_DR_TA_1

Type identifier and cause of transmission asignments (station-specific parameters)

The combination of type identifier / cause of transmission is marked as:

Shaded	Function or ASDU are not required.
Empty	Function or ASDU is not used.
X	Only used in standard direction
R	Only used in reverse direction
B	Used in both directions

Table 48: Cause of transfer

Number	Cause of tranfser
<0>	Not used
<1>	Periodic, cyclic
<2>	Background scan
<3>	Spontaneous
<4>	Initialized
<5>	Request or requested
<6>	Activation (ACT)
<7>	Activation confirmation (ACTCON)
<8>	Deactivation (DEACT)
<9>	Deactivation confirmation (DEACTCON)
<10>	Activation termination (ACTTERM)
<11>	Return information caused by a remote command
<12>	Return information caused by a local command
<13>	File transfer
<20..36>	General interrogation
<37..41>	Counter interrogation
<44>	Unknown type identification
<45>	Unknown cause of transmission
<46>	Unknown address of ASDU
<47>	Unknown information object address

Type identification		Cause of transmission																	
		<1>	<2>	<3>	<4>	<5>	<6>	<7>	<8>	<9>	<10>	<11>	<12>	<13>	<20..36>	<37..41>	<44>	<45>	<46>	<47>
		per/cyc	back	spont	init	req	act	actcon	deact	deactcon	actterm	retrem	retloc	file	inro	reqco	unknowntype	unknownCOT	unknownCAA	unknownIOA
<1>	M_SP_NA_1		X	X								X	X		X					
<2>	M_SP_TA_1																			
<3>	M_DP_NA_1		X	X								X	X		X					
<4>	M_DP_TA_1																			
<5>	M_ST_NA_1		X	X								X	X		X					
<6>	M_ST_TA_1																			
<7>	M_BO_NA_1		X	X											X					
<8>	M_BO_TA_1																			
<9>	M_ME_NA_1	X	X	X											X					
<10>	M_ME_TA_1																			
<11>	M_ME_NB_1	X	X	X											X					
<12>	M_ME_TB_1																			
<13>	M_ME_NC_1	X	X	X											X					
<14>	M_ME_TC_1																			
<15>	M_IT_NA_1			X												X				
<16>	M_IT_TA_1																			
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			

Type identification		Cause of transmission																	
<19>	M_EP_TC_1																		
<20>	M_PS_NA_1		X	X								X	X		X				
<21>	M_ME_ND_1	X	X	X											X				
<30>	M_SP_TB_1			X								X	X						
<31>	M_DP_TB_1			X								X	X						
<32>	M_ST_TB_1			X								X	X						
<33>	M_BO_TB_1			X															
<34>	M_ME_TD_1			X															
<35>	M_ME_TE_1			X															
<36>	M_ME_TF_1			X															
<37>	M_IT_TB_1			X												X			
<38>	M_EP_TD_1			X															
<39>	M_EP_TE_1			X															
<40>	M_EP_TF_1			X															
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1																			
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X	X	X	X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X

Type identification		Cause of transmission																	
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1																			
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X	X	X	X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1																			
<103>	C_CS_NA_1			X			X	X									X	X	X	X
<104>	C_TS_NA_1																			
<105>	C_RP_NA_1																			
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1																X	X	X	X
<111>	P_ME_NB_1																X	X	X	X
<112>	P_ME_NC_1																X	X	X	X
<113>	P_AC_NA_1																X	X	X	X
<120>	F_FR_NA_1																X	X	X	X
<121>	F_SR_NA_1																X	X	X	X
<122>	F_SC_NA_1																X	X	X	X

Type identification		Cause of transmission																	
<123>	F_LS_NA_1															X	X	X	X
<124>	F_AF_NA_1															X	X	X	X
<125>	F_SG_NA_1															X	X	X	X
<126>	F_DR_TA_1																		

Basic application functions

Station initilization

☒	Remote initialzation

Table 49: Cyclic data transmission

☒	Cyclic data transmission

Table 50: Read procedure

☐	Read procedure

Table 51: Spontaneous transmission

☒	Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project specific list.

☐	Single point informaion M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 und M_SP_NA_1
☐	Double point information M_DP_NA_1, M_DP_TA_1 und M_DP_TB_1
☐	Step position information M_ST_NA_1, M_ST_TA_1 und M_ST_TB_1
☐	Bitstring of 32 bits M_BO_NA_1, M_BO_TA_1 und M_BO_TB_1
☐	Measured value, normalized valueM_ME_NA_1, M_ME_TA_1, M_ME_ND_1 und M_ME_TD_1
☐	Measured value, scaled valueM_ME_NB_1, M_ME_TB_1 und M_ME_TE_1
☐	Measured value, short floating point numberM_ME_NC_1, M_ME_TC_1 und M_ME_TF_1

Table 52: Station interrogation

☒	Global
☒	Group 1
☒	Group 2
☒	Group 3
☒	Group 4
☒	Group 5
☒	Group 6
☒	Group 7
☒	Group 8
☒	Group 9
☒	Group 10
☒	Group 11
☒	Group 12
☒	Group 13
☒	Group 14
☒	Group 15
☒	Group 16

The allocation of information object addresses to each individual group must be specified in a separate table.

Table 53: Clock synchronization

☒	Clock synchronization

Table 54: Command transmission

☒	Direct command transmission
☒	Direct setpoint command transmission
☐	Command .selection and execution
☐	Setpoint command .selection and execution
☒	C_SE ACTTERM used
☒	No additional definition
☐	Short pulse duration(duration determined by a system parameter in the outstation)
☐	Long pulse duration(duration determined by a system parameter in the outstation)
☐	Persistent output
☐	Supervision of maximum delay in command direction of commands and setpoint commands
	Maximum allowable delay of commands and setpoint commands

Table 55: Counter interrogation

☒	Counter read
☒	Counter freeze without reset
☒	Counter freeze with reset
☒	Counter reset
☒	General counter request
☒	Request counter group 1
☒	Request counter group 2
☒	Request counter group 3
☒	Request counter group 4
☒	Mode A:Local freeze with spontaneous transmission
☒	Mode B:Local freeze with counter interrogation
☐	Mode C:Freeze and transmit by counter-interrogation commands
☒	Modus D:Freeze by counter-interrogation command, frozen values reported simultaneously

Table 56: Parameter loading

☐	Threshold value
☐	Smoothing factor
☐	Lower limit value for transmission of measured values
☐	High limit value for transmission of measured values

Table 57: Parameter activation

☐	Activation/deactivation of cyclic or periodic transmission of the addressed object

Table 58: Test procedure

☒	Test procedure

File transfer

File transfer in monitoring direction:

☐	Transparent fiel
☐	Transmission of event sequences
☐	Transmission ofdisturbance data of protection equipment
☐	Transmissionof sequences of recorded analog values

File transfer in control direction:

☐	Transparent file

Table 59: Background scan

☒	Background scan

Table 60: Telegram run time acquisition

☐	Telegram run time acquisition

Definition of time outs

Parameter	Default value	Comments	Selected value
t0	30s	Timeout of connection establishment	
t1	15s	Timeout of connection establishment	
t2	10s	Timeout for acknowledgements, if no data telegrams are transferred $t_2<t_1$	
t3	20s	Timeout for sent test telegrams in the event of long idle states	

Maximum range of all time outs: 1 to 255 s, accuracy 1 s.

Maximum number k of unacknowledgedAPDUin I-format and latest APDU acknowledgement (w)

Parameter	Default value	Comments	Selected value
k	12 APDU	Maximum difference between the number of reception sequences and the number of send sequences	
w	8 APDU	Latest acknowledgement after receipt of*w*APDU in I format	

Maximum range of values*k*: 1 to 32767 (2^{15} -1)*APDU*, accuracy 1*APDU*.

Maximum range of values*w*: 1 to 32767 *APDU*, accuracy 1*APDU*(*w*should not exceed two thirds of*k*).

Port number

Parameter	Value	Comments
Port number	2404	in all cases

RFC-2200 suite

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

☒ Ethernet 802.3

☐ Serial interface X.21

☐ Other selection from RFC 2200

List of RFC-2200 documents to be used

1.
2.
3.
4.
5.
6.
7. etc.

Appendix

2.2.5.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

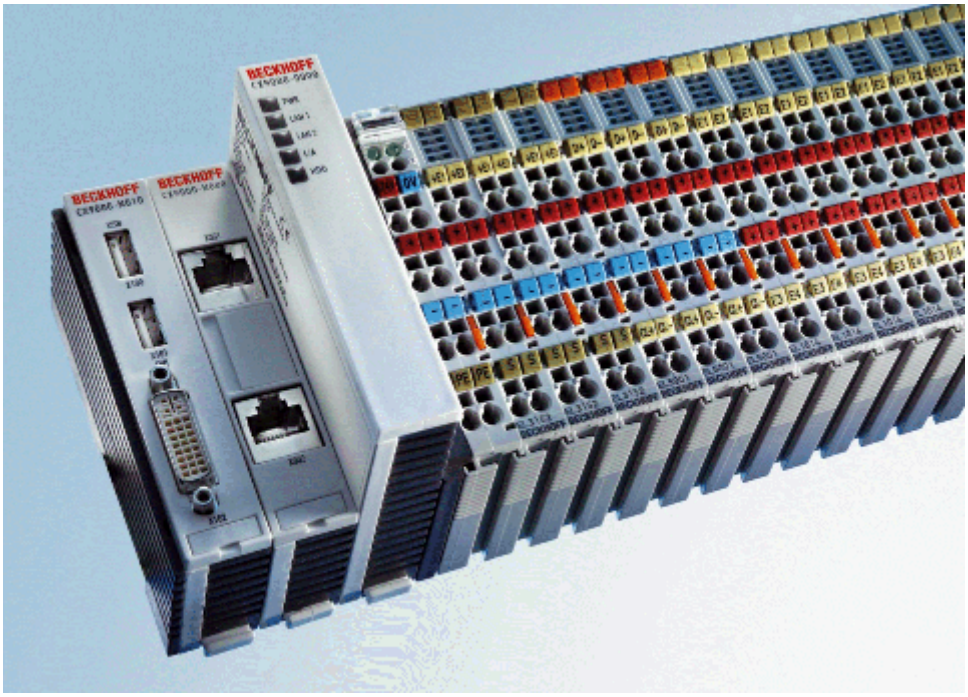
Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

2.2.6 Interoperability list according to IEC 60870-5-104 (slave)

für TwinCAT SPS Bibliothek: IEC 60870-5-104 Unterstation (slave).



Version: 1.5
Datum: 07.03.2014

Here you can open/save the compatibility list (zipped PDF): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984598155.zip

Table of contents:

- General information [▶ 86]
  - Project information [▶ 86]
- Compatibility [▶ 86]
  - System or device [▶ 87]
  - Network configuration [▶ 87]
  - Physical layer [▶ 87]
  - Link layer [▶ 87]
  - Application layer [▶ 87]
  - Selection of standardised ASDUs [▶ 89]

Process information in monitoring direction

Process information in control direction

System information in monitoring direction

System information in control direction

Parameter in control direction

- Basic application functions [▶ 97]
  - Station initialization [▶ 97]
  - Cyclic data transfer

- Read procedure
- Spontaneous transmission
- Double transmission
- Station interrogation
- Clock synchronisation
- Command transmission
- Counter interrogation
- Parameter loading
- Parameter activation
- Test procedure
- File transfer [▶ 100]
- Background scan
- Telegram run time acquisition

- Definition of time outs [▶ 100]
- k- and w-parameters
- Port number
- RFC-2200 suite [▶ 101]
- Appendix [▶ 101]
  - support and service [▶ 101]

General Information

This compatibility list refers to functions supported within IEC 870-5-104.

⊟	Fields marked with are not used with TCP/IP coupling via IEC 870-5-104.

Compatibility of the individual components has to be agreed for individual projects by all parties involved.

Project information

Projekt No.: _____

Project: _____

Responsible: _____

Date: _____

Compatibility

The selected parameters should be marked in the white control fields as follows:

☐	Function or ASDU is not used
☒	Function or ASDU is used as specified (preferred option)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in regular and reverse mode

System or device

☐	System definition
☐	Specifications for the control station
☒	Specifications for the controlled station (slave)

Network configuration

☐	Point-to-point	☐	Multipoint-partyline
☐	Multiple point-to-point	☐	Multipoint-star

Physical layer

Transmission speed:

☐	100 bps	☐	2400 bps	☐	56000 bps
☐	200 bps	☐	4800 bps	☐	64000 bps
☐	300 bps	☐	9600 bps		
☐	600 bps	☐	19200 bps		
☐	1200 bps	☐	38400 bps		

Link layer

According to this application-related standard, only telegram format FT 1.2, single character 1 and the time monitoring interval are used.

Address field of the link layer:

☐	Not available (only symmetric transfer)
☐	One octet
☐	Two octets
☐	Structured
☐	Non structured

Link layer transfer procedure:

☐	Symmetric transfer
☐	Asymmetric transfer

Telegram length:

☐	Maximum length L

Application layer

Tranfer mode for application data

According to this application-related standard, only mode 1 (octet with the lowest value first) according to 4.10 IEC 60870-5-4 is used.

Common ASDU address

☐	One octet	☒	Two octets

Address of the information object

☐	One octet	☒	Structured
☐	Two octets	☒	Non structured
☒	Three octets		

Cause of transmission

☐	One octet	☒	Two octets (with origin address). Preset to 0, if origin address is not available.

Length of the *APDU*

(system-specific parameter, the maximum length of the *APDU* must be specified for each system)

The maximum length of the *APDU* is 253 (unless specified otherwise). The maximum length may be reduced, depending on the system.

253	Maximum length of the *APDU* for each system

Selection of standardised ASDU's

Table 61: Process information in monitoring direction

☒	<1> :=Single-point information	M_SP_NA_1
☐	<2> := Single-point information with time tag	M_SP_TA_1
☒	<3> := Double point information	M_DP_NA_1
☐	<4> := Double point information with time tag	M_DP_TA_1
☒	<5> := Step position information	M_ST_NA_1
☐	<6> := Step position information with time tag	M_ST_TA_1
☒	<7> := Bitstring of 32 bits	M_BO_NA_1
☐	<7> := Bitstring of 32 bits with time tag	M_BO_TA_1
☒	<9> := Measured value, normalized	M_ME_NA_1
☐	<10> := Measured value, normalized value with time tag	M_ME_TA_1
☒	<11> := Measured value, scaled value	M_ME_NB_1
☐	<12> := Measured value, scaled value with time tag	M_ME_TB_1
☒	<13> := Measured value, short floating point value	M_ME_NC_1
☐	<14> := Measured value, short floating point value with time	M_ME_TC_1
☒	<15> := Integrated totals	M_IT_NA_1
☐	<16> := Integrated totals with time tag	M_IT_TA_1
☐	<17> := Event of protection equipment with time tag	M_EP_TA_1
☐	<18> := Packed start events of protection equipment with time tag	M_EP_TB_1
☐	<19> := Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☒	<20> := Packed single-point information with status change detection	M_PS_NA_1
☒	<21> := Measured value, normalized value without quantity descriptor	M_ME_ND_1
☒	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
☒	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
☒	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
☒	<33> := Bitstring of 32 bits with time tag CP56Time2a	M_BO_TB_1
☒	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
☒	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
☒	<37> := Integrated totals with time tag CP56Time2a	M_IT_TB_1
☒	<38> := Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☒	<39> := Packed start events of protection equipment with time tag CP56Time2a	M_EP_TE_1

☒	<40> := Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

Table 62: Process information in control direction

☒	<45> := Single command	C_SC_NA_1
☒	<46> := Double command	C_DC_NA_1
☒	<47> := Regulating step command	C_RC_NA_1
☒	<48> := Set point command, normalized value	C_SE_NA_1
☒	<49> := Set point command, scaled value	C_SE_NB_1
☒	<50> := Set point command, short floating point value	C_SE_NC_1
☒	<51> := Bitstring of 32 bits	C_BO_NA_1
☒	<58> := Single command with time tag CP56Time2a	C_SC_TA_1
☒	<59> := Double commandwith time tag CP56Time2a	C_DC_TA_1
☒	<60> := Regulating step command with time tag CP56Time2a	C_RC_TA_1
☒	<61> := Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☒	<62> := Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☒	<63> := Set point command, short floating point value with time tag CP56Time2a	C_SE_TC_1
☒	<64> := Bitstring of 32 bits with time tag CP56Time2a	C_BO_TA_1

Table 63: System information in monitoring direction

☒	<70> := End of initialization	M_EI_NA_1

Table 64: System information in control direction

☒	<100> := Interrogation command	C_IC_NA_1
☒	<101> := Counter interrogation command	C_CI_NA_1
☒	<102> := Read command	C_RD_NA_1
☒	<103> := Clock synchronization command	C_CS_NA_1
☐	<104> := Test command	C_TS_NA_1
☒	<105> := Reset process command	C_RP_NA_1
☐	<106> := Delay acquisition command	C_CD_NA_1
☒	<107> := Test command with time tagCP56Time2a	C_TS_TA_1

Table 65: Parameters in control direction

☐	<110> := Parameter of measured values, normalized value	P_ME_NA_1
☐	<111> := Parameter of measured values, scaled value	P_ME_NB_1
☐	<112> := Parameter of measured values, short floating point value	P_ME_NC_1
☐	<113>:= Parameter activation	P_AC_NA_1

Table 66: File tranfer

☐	<120>:= File ready	F_FR_NA_1
☐	<121>:= Section ready	F_SR_NA_1
☐	<122> := Call directory, select file, call file, call section	F_SC_NA_1
☐	<123> := Last section, last segment	F_LS_NA_1
☐	<124> := Ack file, ack section	F_AF_NA_1
☐	<125>:= Segment	F_SG_NA_1
☐	<126> := Directory (blank or X, available only in monitor [standard] direction)	F_DR_TA_1

Type identifier and cause of transmission asignments (station-specific parameters)

The combination of type identifier / cause of transmission is marked as:

Shaded	Function or ASDU are not required.
Empty	Function or ASDU is not used.
X	Only used in standard direction
R	Only used in reverse direction
B	Used in both directions

Table 67: Cause of transmission

Number	Cause of transmission
<0>	Not used
<1>	Periodic, cyclic
<2>	Background scan
<3>	Spontaneous
<4>	Initialized
<5>	Request or requested
<6>	Activation (ACT)
<7>	Activation confirmation (ACTCON)
<8>	Deactivation (DEACT)
<9>	Deactivation confirmation (DEACTCON)
<10>	Activation termination (ACTTERM)
<11>	Return information caused by a remote command
<12>	Return information caused by a local command
<13>	File transfer
<20..36>	General interrogation
<37..41>	Counter interrogation
<44>	Unknown type identification
<45>	Unknown cause of transmission
<46>	Unknown address of ASDU
<47>	Unknown information object address

Type identification		Cause of transmission																	
		<1>	<2>	<3>	<4>	<5>	<6>	<7>	<8>	<9>	<10>	<11>	<12>	<13>	<20..36>	<37..41>	<44>	<45>	<46>	<47>
		per/cyc	back	spont	init	req	act	actcon	deact	deactcon	actterm	retrem	retloc	file	inro	reqco	unknowntype	unknownCOT	unknownCAA	unknownIOA
<1>	M_SP_NA_1		X	X		X						X			X					
<2>	M_SP_TA_1																			
<3>	M_DP_NA_1		X	X		X						X			X					
<4>	M_DP_TA_1																			
<5>	M_ST_NA_1		X	X		X						X			X					
<6>	M_ST_TA_1																			
<7>	M_BO_NA_1		X	X		X									X					
<8>	M_BO_TA_1																			
<9>	M_ME_NA_1	X	X	X		X									X					
<10>	M_ME_TA_1																			
<11>	M_ME_NB_1	X	X	X		X									X					
<12>	M_ME_TB_1																			
<13>	M_ME_NC_1	X	X	X		X									X					
<14>	M_ME_TC_1																			
<15>	M_IT_NA_1			X												X				
<16>	M_IT_TA_1																			
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			

Type identification		Cause of transmission																	
<19>	M_EP_TC_1																		
<20>	M_PS_NA_1		X	X		X						X			X				
<21>	M_ME_ND_1	X	X	X		X									X				
<30>	M_SP_TB_1			X		X						X							
<31>	M_DP_TB_1			X		X						X							
<32>	M_ST_TB_1			X		X						X							
<33>	M_BO_TB_1			X		X													
<34>	M_ME_TD_1			X		X													
<35>	M_ME_TE_1			X		X													
<36>	M_ME_TF_1			X															
<37>	M_IT_TB_1			X												X			
<38>	M_EP_TD_1			X															
<39>	M_EP_TE_1			X															
<40>	M_EP_TF_1			X															
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1						X	X	X	X	X						X	X	X	X
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X	X	X	X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X

Type identification		Cause of transmission																	
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1						X	X	X	X	X						X	X	X	X
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X	X	X	X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X											X	X	X	X
<103>	C_CS_NA_1			X			X	X									X	X	X	X
<104>	C_TS_NA_1																			
<105>	C_RP_NA_1						X	X									X	X	X	X
<106>	C_CD_NA_1																			
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1																X	X	X	X
<111>	P_ME_NB_1																X	X	X	X
<112>	P_ME_NC_1																X	X	X	X
<113>	P_AC_NA_1																X	X	X	X
<120>	F_FR_NA_1																X	X	X	X
<121>	F_SR_NA_1																X	X	X	X
<122>	F_SC_NA_1																X	X	X	X

Type identification		Cause of transmission																	
<123>	F_LS_NA_1															X	X	X	X
<124>	F_AF_NA_1															X	X	X	X
<125>	F_SG_NA_1															X	X	X	X
<126>	F_DR_TA_1																		

Basic application functions

Station initilization

☒ Remote initialzation

Table 68: Cyclic data transmission

☒ Cyclic data transmission

Table 69: Read procedure

☒ Read procedure

Table 70: Spontaneous transmission

☒ Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular information object addresses for which double transmission is enabled are defined in a project specific list.

☐ Single point informaion M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 und M_SP_NA_1

☐ Double point information M_DP_NA_1, M_DP_TA_1 und M_DP_TB_1

☐ Step position information M_ST_NA_1, M_ST_TA_1 und M_ST_TB_1

☐ Bitstring of 32 bits M_BO_NA_1, M_BO_TA_1 und M_BO_TB_1

☐ Measured value, normalized valueM_ME_NA_1, M_ME_TA_1, M_ME_ND_1 und M_ME_TD_1

☐ Measured value, scaled valueM_ME_NB_1, M_ME_TB_1 und M_ME_TE_1

☐ Measured value, short floating point numberM_ME_NC_1, M_ME_TC_1 und M_ME_TF_1

Table 71: Station interrogation

☒	Global
☒	Group 1
☒	Group 2
☒	Group 3
☒	Group 4
☒	Group 5
☒	Group 6
☒	Group 7
☒	Group 8
☒	Group 9
☒	Group 10
☒	Group 11
☒	Group 12
☒	Group 13
☒	Group 14
☒	Group 15
☒	Group 16

The allocation of information object addresses to each individual group must be specified in a separate table.

Table 72: Clock synchronization

☒	Clock synchronization

Table 73: Command transmission

☒	Direct command transmission
☒	Direct setpoint command transmission
☒	Command .selection and execution
☒	Setpoint command .selection and execution
☒	C_SE ACTTERM used
☒	No additional definition
☐	Short pulse duration(duration determined by a system parameter in the outstation)
☐	Long pulse duration(duration determined by a system parameter in the outstation)
☐	Persistent output
☐	Supervision of maximum delay in command direction of commands and setpoint commands
	Maximum allowable delay of commands and setpoint commands

Table 74: Counter interrogation

☒	Counter read
☒	Counter freeze without reset
☒	Counter freeze with reset
☒	Counter reset
☒	General counter request
☒	Request counter group 1
☒	Request counter group 2
☒	Request counter group 3
☒	Request counter group 4
☒	Mode A:Local freeze with spontaneous transmission
☒	Mode B:Local freeze with counter interrogation
☒	Mode C:Freeze and transmit by counter-interrogation commands
☒	Modus D:Freeze by counter-interrogation command, frozen values reported simultaneously

Table 75: Parameter loading

☐	Threshold value
☐	Smoothing factor
☐	Lower limit value for transmission of measured values
☐	High limit value for transmission of measured values

Table 76: Parameter activation

☐	Activation/deactivation of cyclic or periodic transmission of the addressed object

Table 77: Test procedure

☒	Test procedure

File transfer

File transfer in monitoring direction:

☐	Transparent fiel
☐	Transmission of event sequences
☐	Transmission ofdisturbance data of protection equipment
☐	Transmissionof sequences of recorded analog values

File transfer in control direction:

☐	Transparent file

Table 78: Background scan

☒	Background scan

Table 79: Telegram run time acquisition

☐	Telegram run time acquisition

Definition of time outs

Parameter	Default value	Comments	Selected value
t0	30s	Timeout of connection establishment	
t1	15s	Timeout of connection establishment	
t2	10s	Timeout for acknowledgements, if no data telegrams are transferred $t_2<t_1$	
t3	20s	Timeout for sent test telegrams in the event of long idle states	

Maximum range of all time outs: 1 to 255 s, accuracy 1 s.

Maximum number k of unacknowledgedAPDUin I-format and latest APDU acknowledgement (w)

Parameter	Default value	Comments	Selected value
k	12 APDU	Maximum difference between the number of reception sequences and the number of send sequences	
w	8 APDU	Latest acknowledgement after receipt of*w*APDU in I format	

Maximum range of values*k*: 1 to 32767 (2^{15} -1)*APDU*, accuracy 1*APDU*.

Maximum range of values*w*: 1 to 32767 *APDU*, accuracy 1*APDU*(*w*should not exceed two thirds of*k*).

Port number

Parameter	Value	Comments
Port number	2404	in all cases

RFC-2200 suite

RFC 2200 is an official Internet Standard which describes the state of standardization of protocols used in the Internet as determined by the Internet Architecture Board (IAB). It offers a broad spectrum of actual standards used in the Internet. The suitable selection of documents from RFC 2200 defined in this standard for given projects has to be chosen by the user of this standard.

☒ Ethernet 802.3

☐ Serial interface X.21

☐ Other selection from RFC 2200

List of RFC-2200 documents to be used

1.
2.
3.
4.
5.
6.
7. etc.

Appendix

2.2.6.1 Support and Service

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

Download finder

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

The downloads are available in various formats.

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

3 Installation

3.1 System requirements

The following article describes the minimum requirements needed for engineering and/or runtime systems.

Engineering environment

An engineering environment describes a computer which used to develop but NOT run PLC programs. On an engineering computer, the following requirements are needed:

- TwinCAT 3.1.4018.0 XAE (engineering installation) or higher
- TwinCAT 3 Function TF6500
- Please note: For engineering purposes, a 7-Day trial license may be (repeatedly) used, as described in our licensing article

Runtime environment

A runtime environment describes a computer which runs PLC programs. On a runtime computer, the following requirements are needed:

- TwinCAT 3.1.4018.0 XAR (runtime installation) or higher
- TwinCAT 3 Function TF6500
- Licenses for TC1200 PLC and TF6500
- Please note: For testing purposes, a 7-Day trial license may be used, as described in our licensing article

Engineering and runtime on the same computer

In case you would like to run both the engineering and runtime environments on the same computer (for example to test the PLC program before downloading it to the target runtime), the following requirements are needed:

- TwinCAT 3.1.4018.0 XAE (engineering installation) or higher
- TwinCAT 3 Function TF6500
- Licenses for TC1200 PLC and TF6500
- Please note: For testing purposes, a 7-Day trial license may be used, as described in our licensing article

3.2 Installation

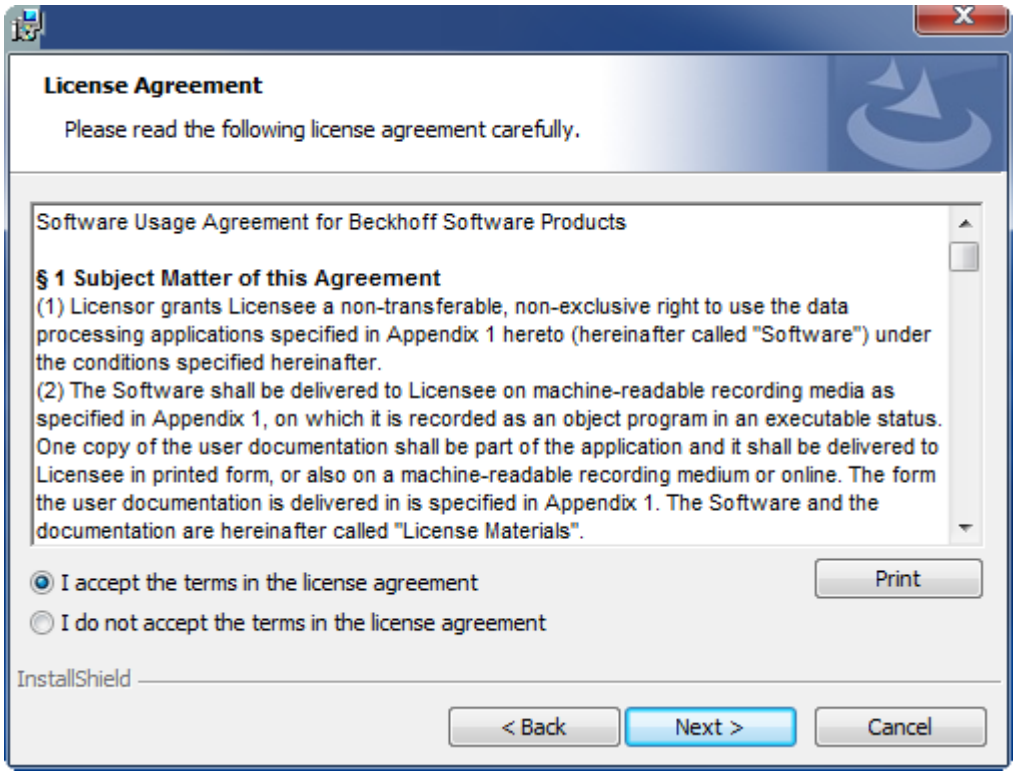
Setup installation (TwinCAT 3.1 Build 4024)

The following section describes how to install the TwinCAT 3 function for Windows-based operating systems.

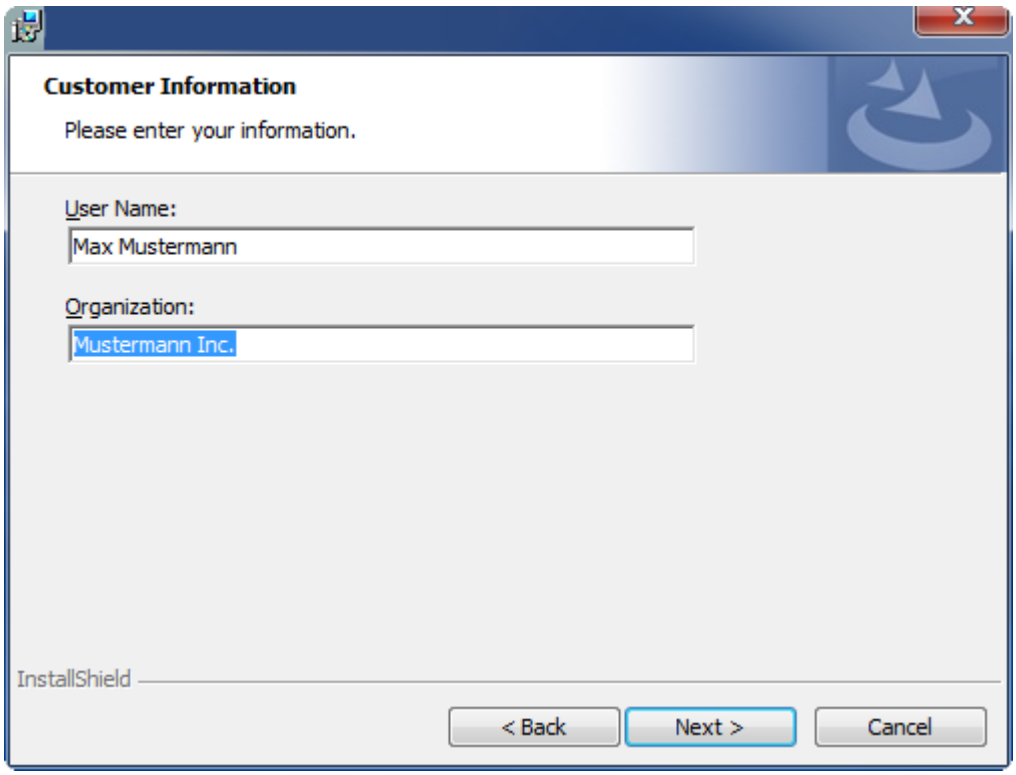
✓ The TwinCAT 3 function setup file was downloaded from the Beckhoff website.

1. Run the setup file as administrator. To do this, select the **Run As Admin** command in the context menu of the file.
   ⇨ The installation dialog opens.

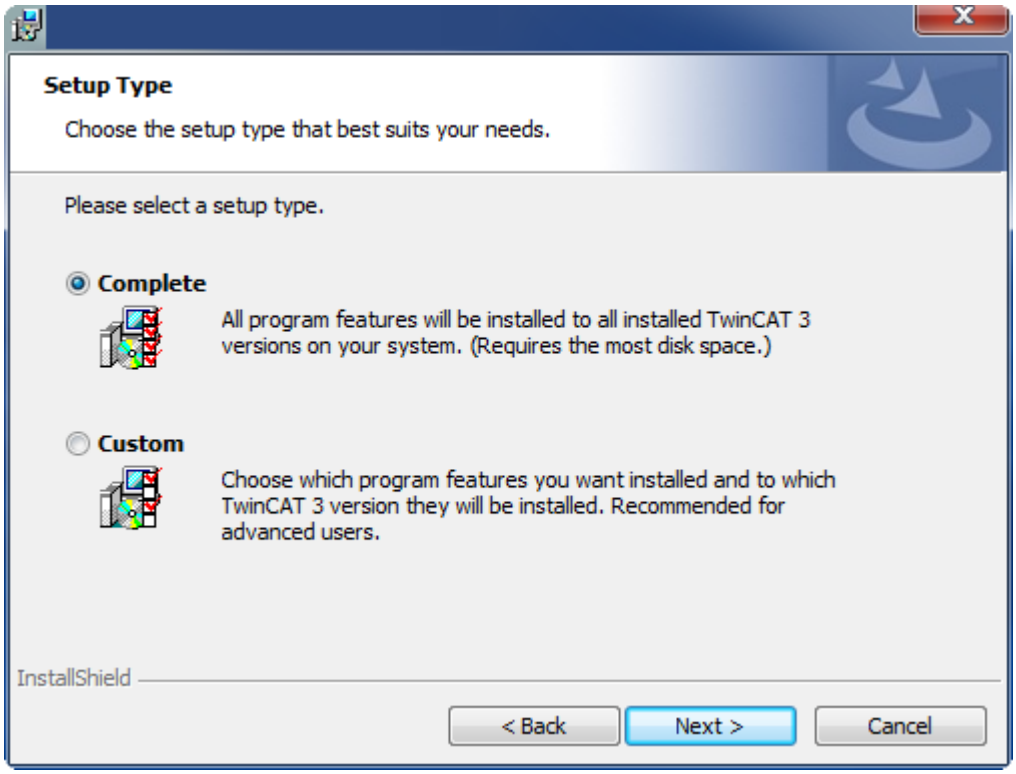
2. Accept the end user licensing agreement and click **Next**.



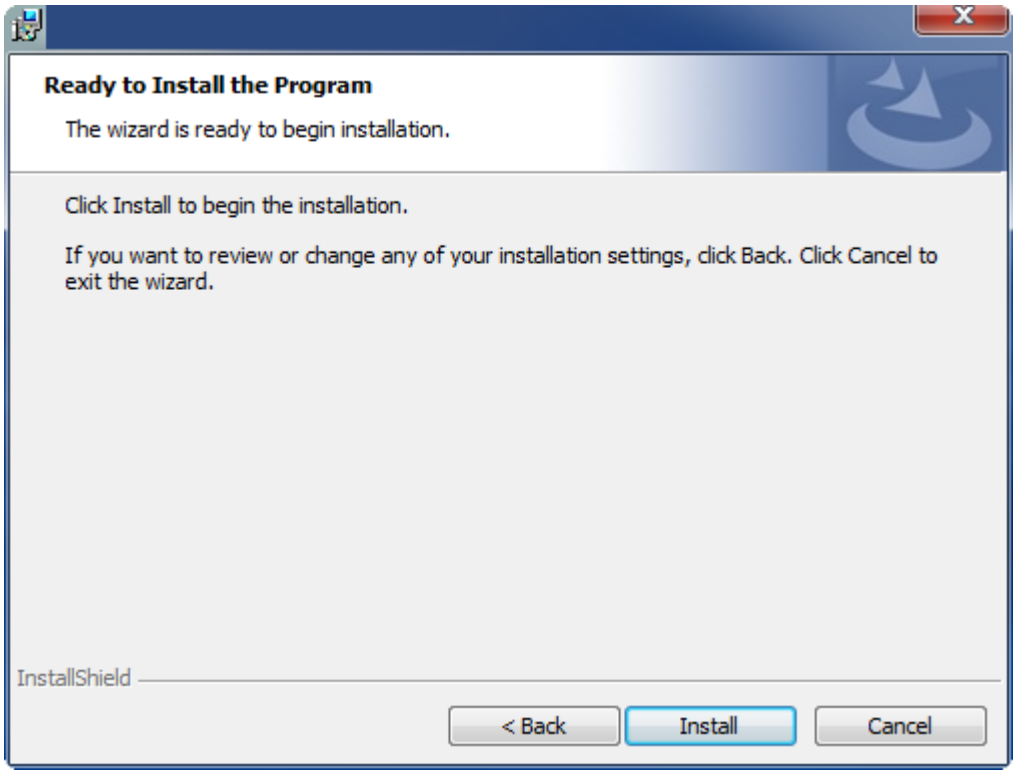
3. Enter your user data.



4. If you want to install the full version of the TwinCAT 3 function, select **Complete** as the installation type. If you want to install the TwinCAT 3 function components separately, select **Custom**.

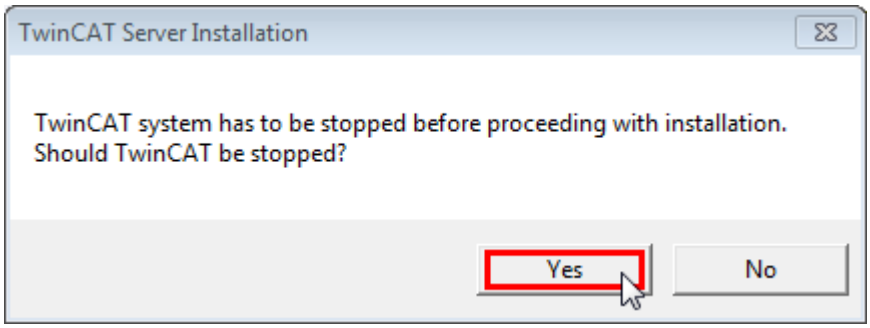


5. Click **Next**, then **Install** to start the installation.

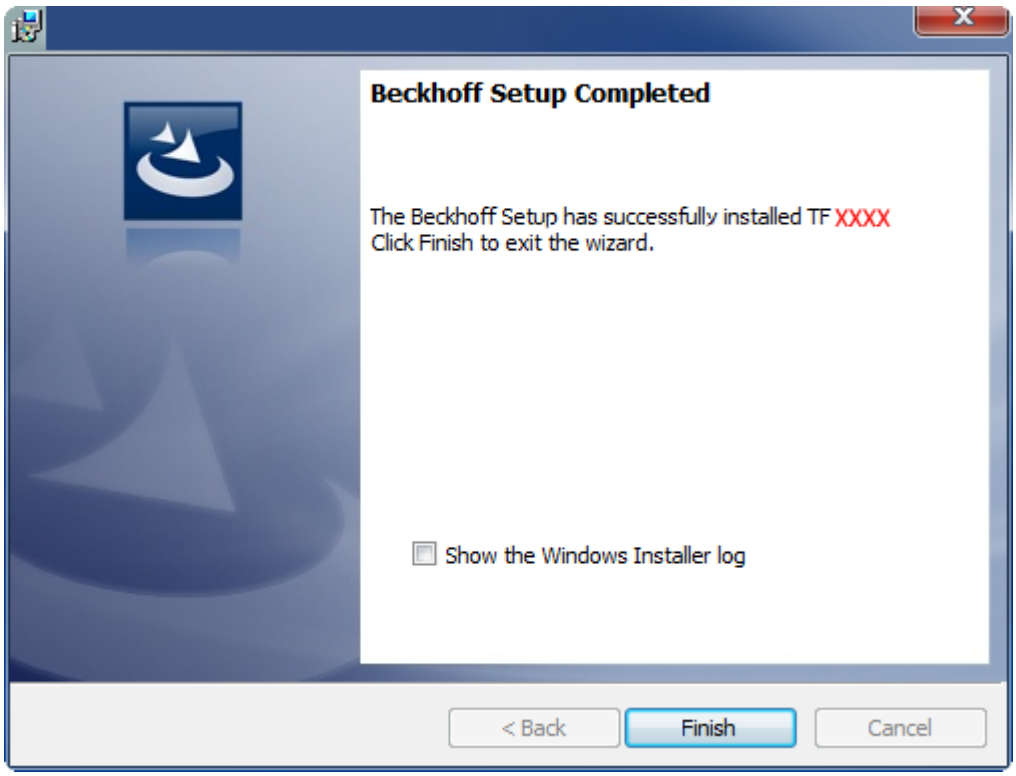


⇨ A dialog box informs you that the TwinCAT system must be stopped to proceed with the installation.

6. Confirm the dialog with **Yes.**



7. Click **Finish** to exit the setup.



⇨ The TwinCAT 3 function has been installed successfully.

3.3 Installing TwinCAT/BSD

The TF6500 TwinCAT 3 IEC 60870-5-10x function is not directly available as a package for TwinCAT/BSD. The TwinCAT 3 TCP/IP Server package is relevant for operation under TwinCAT/BSD. This is available in the TwinCAT/BSD package repository. It can be installed via the command, under the package name "TF6310-TCP-IP".

```
doas pkg install TF6310-TCP-IP
```

Further information on the package server can be found in the Embedded PC section of the TwinCAT/BSD manual.

After the system or TwinCAT has been restarted, the TwinCAT 3 TCP/IP Server is also started and can be used via the telecontrol protocol.

3.4 Installation Windows CE

The following section describes how to install a TwinCAT 3 function (TFxxx) on a Beckhoff Embedded PC with Windows CE.

1. Download and install the setup file [▶ 107]

2. Transfer the CAB file to the Windows CE device [▶ 107]
3. Run the CAB file on the Windows CE device [▶ 107]

If an older TFxxx version is already installed on the Windows CE device, it can be updated:

- Software upgrade [▶ 108]

Download and install the setup file

The CAB installation file for Windows CE is part of the TFxxx setup. This is made available on the Beckhoff website www.beckhoff.com and automatically contains all versions for Windows XP, Windows 7 and Windows CE (x86 and ARM).

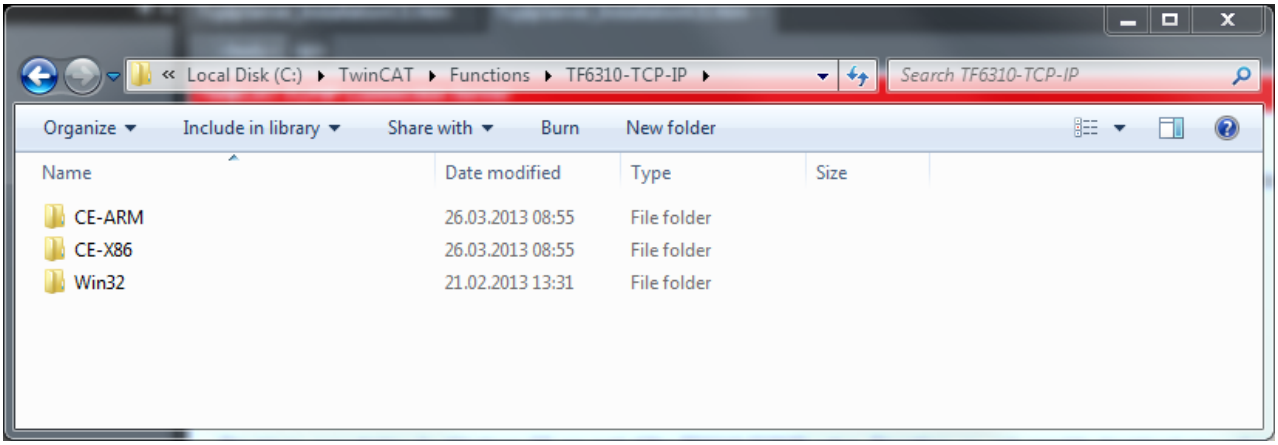
Download the TFxxx setup file and install the TwinCAT 3 function as described in the Installation [▶ 103] section.

After the installation, the installation folder contains three directories (one directory per hardware platform):

- **CE-ARM:** ARM-based embedded PCs running Windows CE, e.g. CX8090, CX9020
- **CE-X86:** X86-based embedded PCs running Windows CE, e.g. CX50xx, CX20x0
- **Win32:** embedded PCs running Windows XP, Windows 7 or Windows Embedded Standard

The CE-ARM and CE-X86 directories contain the CAB files of the TwinCAT 3 function for Windows CE in relation to the respective hardware platform of the Windows CE device.

Example: installation folder "TF6310"



Transfer the CAB file to the Windows CE device

Transfer the corresponding CAB file to the Windows CE device.

There are various options for transferring the executable file:

- via network shares
- via the integrated FTP server
- via ActiveSync
- via CF/SD cards

Further information can be found in the Beckhoff Information System in the "Operating Systems" documentation (Embedded PC > Operating Systems > CE).

Run the CAB file on the Windows CE device

After transferring the CAB file to the Windows CE device, double-click the file there. Confirm the installation dialog with **OK**. Then restart the Windows CE device.

After restarting the device, the files of the TwinCAT 3 function (TFxxxx) are automatically loaded in the background and are then available.

The software is installed in the following directory on the Windows CE device:
\Hard Disk\TwinCAT\Functions\TFxxxx

Software upgrade

If an older version of the TwinCAT 3 function is already installed on the Windows CE device, carry out the following steps on the Windows CE device to upgrade to a new version:

1. Open the CE Explorer by clicking **Start > Run** and entering "Explorer".
2. Navigate to *\Hard Disk\TwinCAT\Functions\TFxxx\xxxx*.
3. Rename the file *Tc*.exe* to *Tc*.old*.
4. Restart the Windows CE device.
5. Transfer the new CAB file to the Windows CE device.
6. Run the CAB file on the Windows CE device and install the new version.
7. Delete the file *Tc*.old*.
8. Restart the Windows CE device.

⇨ The new version is active after the restart.

3.5 Licensing

The TwinCAT 3 function can be activated as a full version or as a 7-day test version. Both license types can be activated via the TwinCAT 3 development environment (XAE).

Licensing the full version of a TwinCAT 3 Function

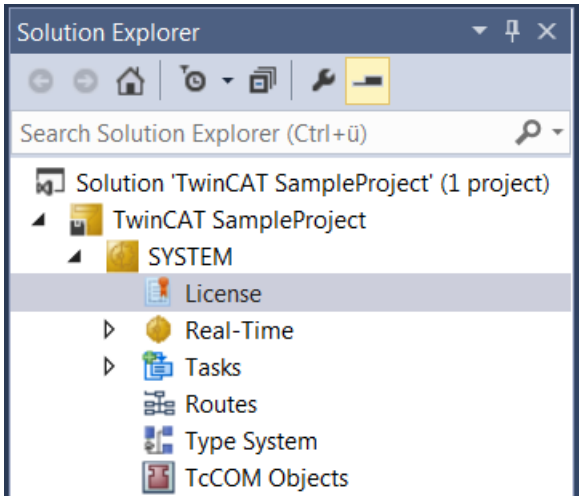
A description of the procedure to license a full version can be found in the Beckhoff Information System in the documentation "TwinCAT 3 Licensing".

Licensing the 7-day test version of a TwinCAT 3 Function



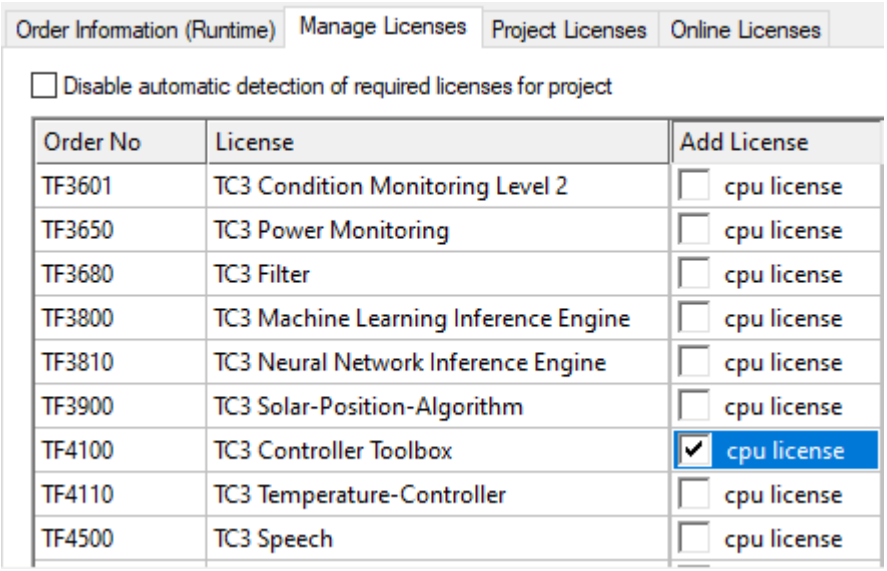
A 7-day test version cannot be enabled for a TwinCAT 3 license dongle.

1. Start the TwinCAT 3 development environment (XAE).
2. Open an existing TwinCAT 3 project or create a new project.
3. If you want to activate the license for a remote device, set the desired target system. To do this, select the target system from the **Choose Target System** drop-down list in the toolbar.
   ⇨ The licensing settings always refer to the selected target system. When the project is activated on the target system, the corresponding TwinCAT 3 licenses are automatically copied to this system.
4. In the **Solution Explorer**, double-click **License** in the **SYSTEM** subtree.



⇨ The TwinCAT 3 license manager opens.

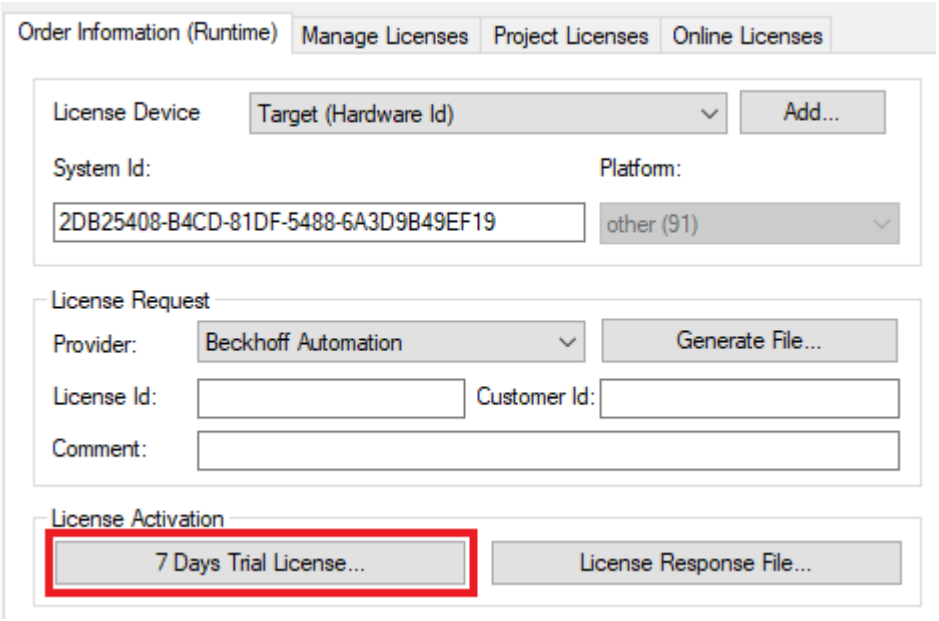
5. Open the **Manage Licenses** tab. In the **Add License** column, check the check box for the license you want to add to your project (e.g. "TF4100 TC3 Controller Toolbox").



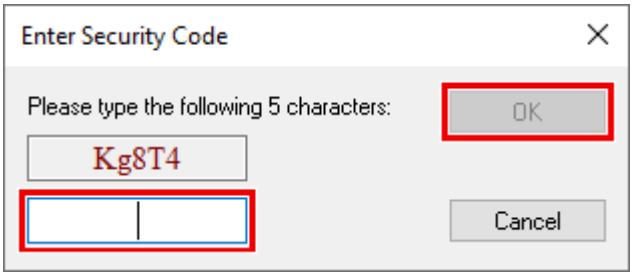
6. Open the **Order Information (Runtime)** tab.

⇨ In the tabular overview of licenses, the previously selected license is displayed with the status “missing”**.**

7. Click **7-Day Trial License...** to activate the 7-day trial license.



⇨ A dialog box opens, prompting you to enter the security code displayed in the dialog.



8. Enter the code exactly as it is displayed and confirm the entry.
9. Confirm the subsequent dialog, which indicates the successful activation.

⇨ In the tabular overview of licenses, the license status now indicates the expiry date of the license.

10. Restart the TwinCAT system.

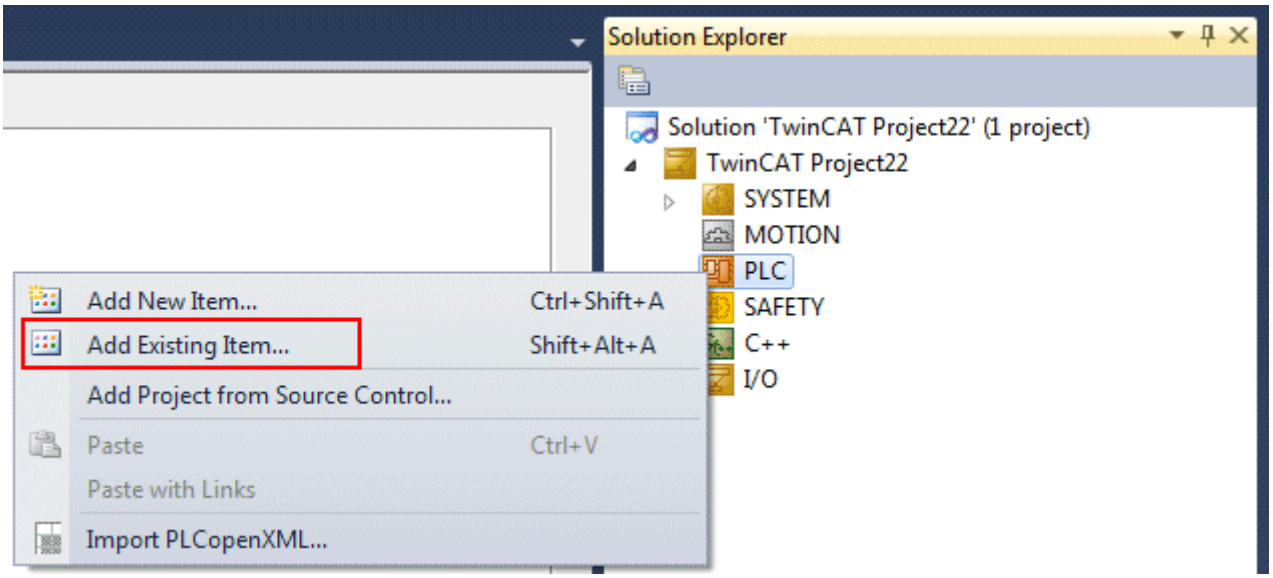
⇨ The 7-day trial version is enabled.

3.6 Migration from TwinCAT 2

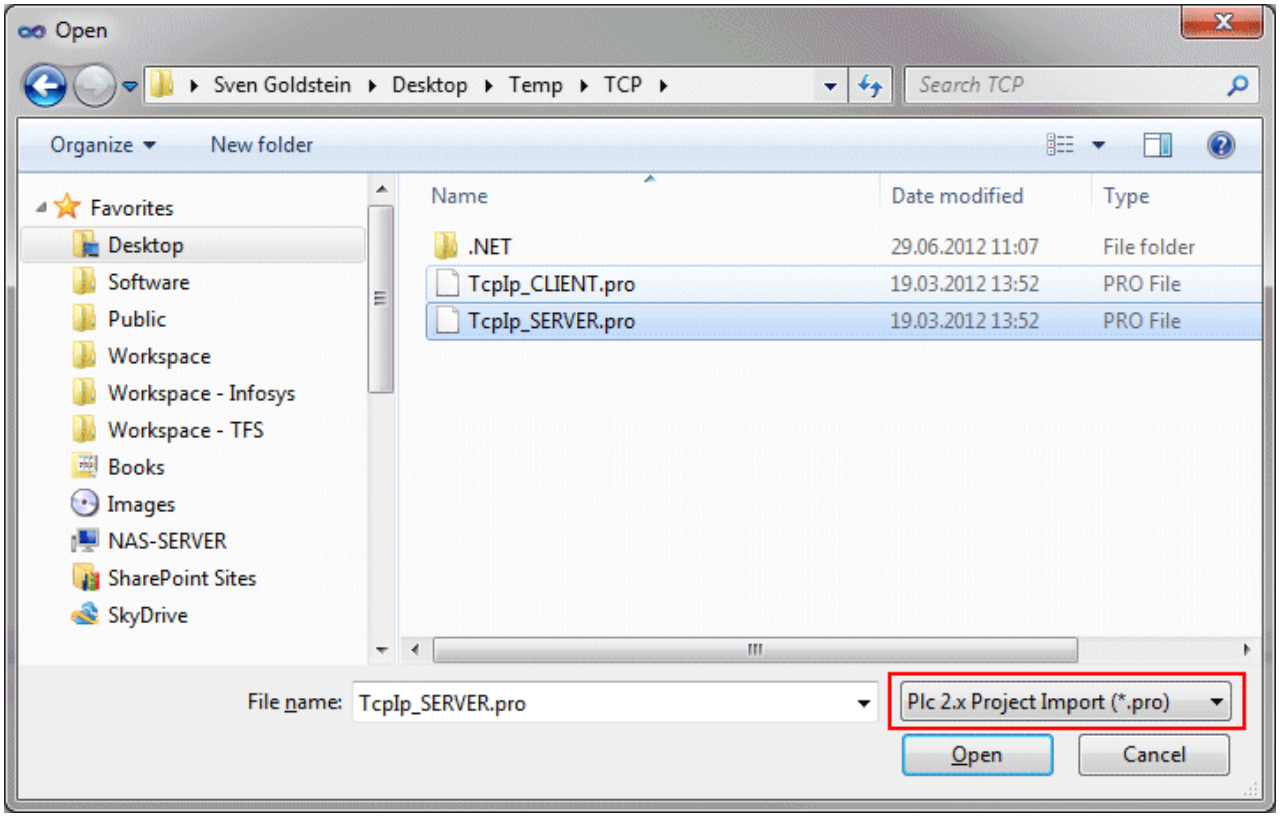
If you would like to migrate an existing TwinCAT 2 PLC project which uses the IEC 60870-5-10x PLC (supplement) libraries, you need to perform some manual steps to ensure that the TwinCAT 3 PLC converter can process the TwinCAT 2 project file (*.pro).

The TwinCAT 2 Libraries are delivered with the Supplement Installation.
By default, these library files are installed in "C:\TwinCAT\Plc\Lib\". Depending on the library used in your PLC project, you need to copy the corresponding library file to "C:\TwinCAT\3.1\Components\Plc\Converter\Lib\" and then perform the following steps:

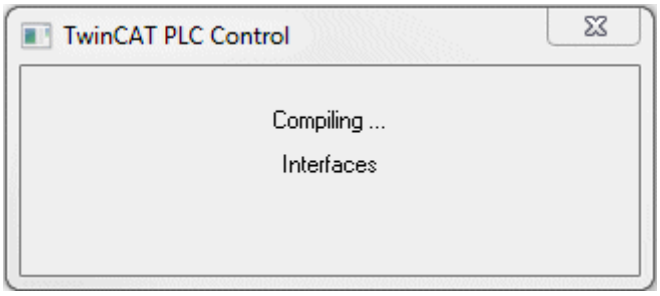
- Open TwinCAT XAE
- Create a new TwinCAT 3 solution
- Right-click the *PLC* node and select *Add existing item*

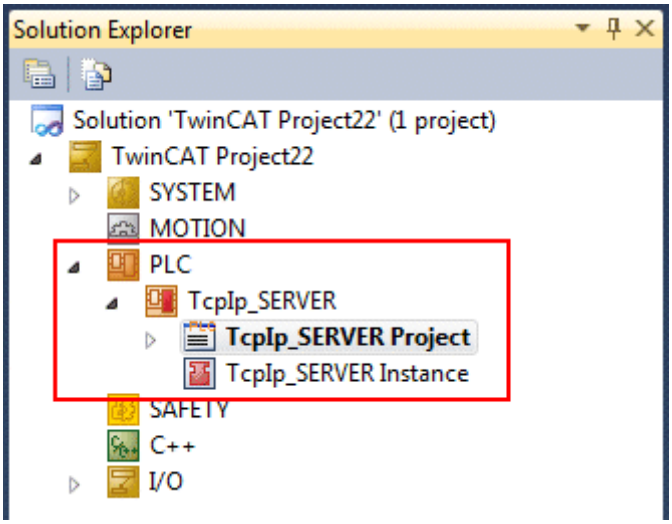


- In the Open File Dialog, select the file type *Plc 2.x Project Import (*.pro),* browse to the folder containing your TwinCAT2 PLC project and select the corresponding pro-file followed by clicking on *Open*



- After clicking on *Open*, TwinCAT 3 starts the converter process and will finally display the converted PLC project under the *PLC* node





4 Technical reference

4.1 IEC60870-5-101

TF6500 implements two interfaces for the transport of the IEC 60870-5-101 ASDUs (service data unitof the application layer) via the serial interfaces of the pc and the Beckhoff KL6xxx/EL6xxx bus terminals.

‘High level’ interface:

This interface is a so-called ‘single-block solution’. All functions are encapsulated in one PLC block. The block implements the most important services and functions. This implementation is sufficient for over 90% of applications.

Pro: Very little PLC programming work is required in order to create an application; all services, such as general query, counter query, time synchronisation, command execution, spontaneous data transmission etc. are already implemented in the block and are executed automatically; the mapping of the IEC<->PLC process data and that of the data points is configured via function calls; the PLC programmer does not need to be very well acquainted with the protocol standard;

Contra: The PLC application has only a small influence on the execution of the protocol; no influence on the execution of the services – these are automatically implemented internally; time stamps are automatically generated by the block and cannot be changed (handed over by externals); only the direct command execution, for example, is possible; poorer performance if there are many data points.

This interface is recommended if you:

- are not familiar with the protocol standard;
- are implementing a simple application with few data points (<1000);
- are not placing any great performance demands on the application;
- are not sending any special command execution such as Select/Execute or data + time stamp from external devices;
- do not require any functions that are not supported according to the compatibility list;

‘Low level’ interface:

This interface starts lower down on the protocol stack and enables the sending and/or receipt of individual frames (ASDUs). Within the protocol structure this interface is above the link layer and pre-implements the required procedures and transfer telegram formats. Application functions such as calling of the individual counter readings or the product identification are not implemented in the interface, although the user can implement these application functions via the interface.

Pro: Very flexible; all properties in the ASDU frame can be changed (e.g. an own time stamp, Select/ Execute or a special command execution etc.); high performance can be achieved because only the necessary services are implemented; high performance if there are many data points;

Contra: Larger amount of programming work; all services, such as general query, counter query, time synchronization, command execution, spontaneous data transmission etc. must be implemented by the PLC programmer himself (programmed out); the PLC programmer must be familiar with the protocol standard.

This interface is recommended if you:

- are familiar with the protocol standard;
- are implementing a protocol converter application;
- are implementing virtually all available standard functions in the application;
- are using special functions, such as the relaying of the time stamp from a Modbus device or the gaining of control over the command execution;
- require functions that are not supported according to the compatibility list;
- have many data points (>1000) and need high performance;

Within the protocol structure this interface is above the link layer and pre-implements the required procedures and transfer telegram formats. Application functions such as general query, time synchronization or counter queries are not implemented in the interface, although the user can implement these application functions via the interface.

Protocol structure of the end system:

Selection of application functions from IEC 60870-5-5	User process
Selection of ASDU (service data units of the application layer) from IEC 60870-5-3, IEC 60870-5-4 and IEC 60870-5-101	Application layer (7)
N/A	Presentation layer (6)
	Session layer (5)
	Transport layer (4)
	Network layer (3)
unbalanced / balanced IEC 60870-5-2 IEC 60870-5-1 (FT 1.2)	Link layer (2)
EIA RS485, RS232 (V.24), Fiber Optics	Physical layer (1)

Layers 3 to 6 are not used.

4.1.1 IEC 60870-5-101 telegram structure

- FT 1.2 frame format is used;
- Frames with fixed and variable block length and single control characters are used;
- Hamming distance = 4;

Frame with variable length

This frame type is used to transmit user data between controlling and controlled station

byte\bit	7	6	5	4	3	2	1	0					
0	Start byte 1 (0x68)								Header		LPCI		LPDU
1	Block length												
2	Block length (copy)												
3	Start byte 2 (0x68)												
4	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field				
5	Link address fields												
	(0, 1 or 2 octets)												
	Type identification [▶ 601]								DATA UNIT IDENTIFIER		ASDU		
	SQ	Number of objects											
	T	P/N	Cause of transmission (COT) [▶ 611]										
	Originator address (ORG, 0 or 1 octets)												
	ASDU address fields												
	(1 or 2 octets)												
	Information object address fields (IOA)								Info-object				
	(1,2 or 3 octets)												
	Object information												
n-1	Checksum								Tail		LPCI		
n	Stop byte (0x16)												

Frame with fixed length

This frame type is used for link layer services. Sometimes used as an confirmation.

byte\bit	7	6	5	4	3	2	1	0			
0	Start byte 1 (0x10)										LPCI
1	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field		
2	Link address										
	(0, 1 or 2 octets)										
n-1	Checksum										
n	Stop byte (0x16)										

Single character data

Single characters are used to confirm link layer and user data transmissions.

- 0xE5 (positive acknowledge);
- 0xA2 (negative acknowledge);

Simple samples and explanations

101substation configuration: Link address = 2 octets, COT = 1 octet (ORG address not used), ASDU address = 2 octets, IOA address = 2 octets

Sample 1

10 49 0C 00 55 16

LPDU bytes	Explanation
10	Start byte: frame with fixed length
49	Control filed: PRM-bit set (frame from primary station), function code = 9 (link status)
0C 00	Link address (2 octets) = 12 dec.
55	Checksum
16	Stop byte

Sample 2

10 0B 0C 00 17 16

LPDU bytes	Explanation
10	Start byte: frame with fixed length
0B	Control field: PRM-bit not set (frame from secondary station), function code = 11 (status of link or access demand)
0C 00	Link address (2 octets) = 12 dec.
17	Checksum
16	Stop byte

Sample 3

68 0B 0B 68 08 0C 00 65 01 0A 0C 00 00 00 05 95 16

LPDU bytes	Explanation
68	Start byte 1: frame with variable length
0B 0B	Block length and block length copy
68	Start byte 2
08	Control field: PRM-bit not set (frame from secondary station), function code = 8 (user data)
0C 00	Link address (2 octets) = 12 dec.
65	Type identification: C_CI_NA_1 (counter interrogation)
01	Number of objects = 1
0A	Cause of transmission = 10 (activation confirmation)
0C 00	Common ASDU address (2 octets) = 12 dec.
00 00	Object address (2 octets) = 0
05	Counter interrogation request qualifier = 5 (general counter interrogation)
95	Checksum
16	Stop byte

Sample 4

68 0F 0F 68 08 0C 00 0F 01 03 0C 00 81 30 DA 16 00 00 07 DB 16

LPDU bytes	Explanation
68	Start byte 1: frame with variable length
0F 0F	Block length and block length copy
68	Start byte 2
08	Control field: PRM-bit not set (frame from secondary station), function code = 8 (user data)
0C 00	Link address (2 octets) = 12 dec.
0F	Type identification: M_IT_NA_1 (integrated total)
01	Number of objects = 1
03	Cause of transmission = 3 (spontanous)
0C 00	Common ASDU address (2 octets) = 12 dec.
81 30	Object address (2 octets)
DA 16 00 00	BCR (binary counter value)
07	Quality descriptor = 7 (sequence)
DB	Checksum
16	Stop byte

Sample 4

68 2B 2B 68 08 0C 00 0B 07 03 0C 00 10 30 BE 09 00 11 30 90 09 00 0E 30 75 00 00 28 30 25 09 00 29 30 75 00 00 0F 30 0F 0A 00 2E 30 AE 05 00 85 16

LPDU bytes	Explanation
68	Start byte 1: frame with variable length
2B 2B	Block length and block length copy
68	Start byte 2
08	Control field: PRM-bit not set (frame from secondary station), function code = 8 (user data)
0C 00	Link address (2 octets) = 12 dec.
0B	Type identification: M_ME_NB_1(measured value, scaled value)
07	Number of objects = 7
03	Cause of transmission = 3 (spontanous)
0C 00	Common ASDU address (2 octets) = 12 dec.
10 30	Object address (2 octets) of **first** information object
BE 09 00	Scaled value + QDS (quality descriptor) of **first** information object
11 30	Object address (2 octets) of **second** information object
90 09 00	Scaled value + QDS (quality descriptor) of **second** information object
0E 30	Object address (2 octets) of **third** information object
75 00 00	Scaled value + QDS (quality descriptor) of **third** information object
28 30 25 09 00 29 30 75 00 00 0F 30 0F 0A 00 2E 30 AE 05 00	Object address + Scaled value + QDS (quality descriptor) of information object four to seven
85	Checksum
16	Stop byte

4.1.2 ASDU object description

4.1.2.1 Standard IEC 60870-5-101 data types

Type	Dec	Hex	Description
ASDU_TYPEUNDEF	0	0x00	Not used
M_SP_NA_1 [▶ 120]	1	0x01	Single-point information
M_SP_TA_1 [▶ 122]	2	0x02	Single-point information with time tag
M_DP_NA_1 [▶ 126]	3	0x03	Double-point information
M_DP_TA_1 [▶ 128]	4	0x04	Double-point information with time tag
M_ST_NA_1 [▶ 132]	5	0x05	Step position information
M_ST_TA_1 [▶ 134]	6	0x06	Step position information with time tag
M_BO_NA_1 [▶ 138]	7	0x07	Bitstring of 32 bit
M_BO_TA_1 [▶ 140]	8	0x08	Bitstring of 32 bit with time tag
M_ME_NA_1 [▶ 144]	9	0x09	Measured value, normalised value
M_ME_TA_1 [▶ 146]	10	0x0A	Measured value, normalized value with time tag
M_ME_NB_1 [▶ 152]	11	0x0B	Measured value, scaled value
M_ME_TB_1 [▶ 154]	12	0x0C	Measured value, scaled value wit time tag
M_ME_NC_1 [▶ 158]	13	0x0D	Measured value, short floating point number
M_ME_TC_1 [▶ 160]	14	0x0E	Measured value, short floating point number with time tag
M_IT_NA_1 [▶ 164]	15	0x0F	Integrated totals
M_IT_TA_1 [▶ 166]	16	0x10	Integrated totals with time tag
M_EP_TA_1 [▶ 212]	17	0x11	Event of protection equipment with time tag
M_EP_TB_1 [▶ 214]	18	0x12	Packed start events of protection equipment with time tag
M_EP_TC_1 [▶ 217]	19	0x13	Packed output circuit information of protection equipment with time tag
M_PS_NA_1 [▶ 228]	20	0x14	Packed single point information with status change detection
M_ME_ND_1 [▶ 150]	21	0x15	Measured value, normalized value without quality descriptor
ASDU_TYPE_22..29	22..29	0x16..0x1D	Reserved (standard area)
M_SP_TB_1 [▶ 124]	30	0x1E	Single-point information with time tag CP56Time2a
M_DP_TB_1 [▶ 130]	31	0x1F	Double-point information with time tag CP56Time2a
M_ST_TB_1 [▶ 136]	32	0x20	Step position information with time tag CP56Time2a
M_BO_TB_1 [▶ 142]	33	0x21	Bitstring of 32 bit with time tag CP56Time2a
M_ME_TD_1 [▶ 148]	34	0x22	Measured value, normalised value with time tag CP56Time2a
M_ME_TE_1 [▶ 156]	35	0x23	Measured value, scaled value with time tag CP56Time2a
M_ME_TF_1 [▶ 162]	36	0x24	Measured value, short floating point number with time tag CP56Time2a
M_IT_TB_1 [▶ 168]	37	0x25	Integrated totals with time tag CP56Time2a
M_EP_TD_1 [▶ 220]	38	0x26	Event of protection equipment with time tag CP56Time2a
M_EP_TE_1 [▶ 222]	39	0x27	Packed start events of protection equipment with time tag CP56Time2a
M_EP_TF_1 [▶ 225]	40	0x28	Packed output circuit information of protection equipment with time tag CP56Time2a
ASDU_TYPE_41..44	41..44	0x29..0x2C	Reserved (standard area)

Type	Dec	Hex	Description
C_SC_NA_1 [▶ 170]	45	0x2D	Single command
C_DC_NA_1 [▶ 174]	46	0x2E	Double command
C_RC_NA_1 [▶ 178]	47	0x2F	Regulating step command
C_SE_NA_1 [▶ 182]	48	0x30	Set-point Command, normalised value
C_SE_NB_1 [▶ 186]	49	0x31	Set-point Command, scaled value
C_SE_NC_1 [▶ 190]	50	0x32	Set-point Command, short floating point number
C_BO_NA_1 [▶ 194]	51	0x33	Bitstring 32 bit command
ASDU_TYPE_52..57	52..57	0x34..0x39	Reserved (standard area)
C_SC_TA_1 [▶ 172]	58	0x3A	Single command with time tag CP56Time2a
C_DC_TA_1 [▶ 176]	59	0x3B	Double command with time tag CP56Time2a
C_RC_TA_1 [▶ 180]	60	0x3C	Regulating step command with time tag CP56Time2a
C_SE_TA_1 [▶ 184]	61	0x3D	Measured value, normalised value command with time tag CP56Time2a
C_SE_TB_1 [▶ 188]	62	0x3E	Measured value, scaled value command with time tag CP56Time2a
C_SE_TC_1 [▶ 192]	63	0x3F	Measured value, short floating point number command with time tag CP56Time2a
C_BO_TA_1 [▶ 196]	64	0x40	Bitstring of 32 bit command with time tag CP56Time2a
ASDU_TYPE_65..69	65..69	0x41..0x45	Reserved (standard area)
M_EI_NA_1 [▶ 200]	70	0x46	End of Initialisation
ASDU_TYPE_71..99	71..99	0x47..0x63	Reserved (standard area)
C_IC_NA_1 [▶ 204]	100	0x64	Interrogation command
C_CI_NA_1 [▶ 206]	101	0x65	Counter interrogation command
C_RD_NA_1 [▶ 210]	102	0x66	Read command
C_CS_NA_1 [▶ 202]	103	0x67	Clock synchronisation command
C_TS_NA_1 [▶ 198]	104	0x68	Test command
C_RP_NA_1 [▶ 208]	105	0x69	Reset process command
C_CD_NA_1	106	0x6A	Delay acquisition command
C_TS_TA_1	107	0x6B	Test command with time tag CP56Time2a
ASDU_TYPE_108..109	108..109	0x6C..0x6D	Reserved (standard area)
P_ME_NA_1 [▶ 230]	110	0x6E	Parameter of measured values, normalized value
P_ME_NB_1 [▶ 232]	111	0x6F	Parameter of measured values, scaled value
P_ME_NC_1 [▶ 234]	112	0x70	Parameter of measured values, short floating point number
P_AC_NA_1 [▶ 236]	113	0x71	Parameter activation
ASDU_TYPE_114..119	114..119	0x72..0x77	Reserved (standard area)
F_FR_NA_1	120	0x78	File ready
F_SR_NA_1	121	0x79	Section ready
F_SC_NA_1	122	0x7A	Call directory, select file, call file, call section
F_LS_NA_1	123	0x7B	Last section, last segment
F_FA_NA_1	124	0x7C	ACK file, ACK section
F_SG_NA_1	125	0x7D	Segment
F_DR_TA_1	126	0x7E	Directory
ASDU_TYPE_127..255	127..255	0x7F..0xFF	Reserved (user area)

4.1.2.2 Single-point information

4.1.2.2.1 M_SP_NA_1

Single-point information without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x01 (1)	= M_SP_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	SPI [▶ 628]	SIQ = Single-point information with quality descriptor
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.2.2 M_SP_TA_1

Single-point information with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x02 (2)	= M_SP_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 4									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	SPI [▶ 628]	SIQ = Single-point information with quality descriptor
			|---	data[1..3] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[4..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.2.3 M_SP_TB_1

Single-point information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x1E (30)	= M_SP_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	SPI [▶ 628]	SIQ = Single-point information with quality descriptor
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.3 Double-point information

4.1.2.3.1 M_DP_NA_1

Double-point information without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x03 (3)	= M_DP_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	DPI [▶ 629]		DIQ = Double-point information with quality descriptor
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.3.2 M_DP_TA_1

Double-point information with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x04 (4)	= M_DP_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 4									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	DPI [▶ 629]		DIQ = Double-point information with quality descriptor
			|---	data[1...3] =	CP24Time2a [▶ 636]								Three octet binary time tag
			|---	data[4...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.3.3 M_DP_TB_1

Double-point information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x1F (31)	= M_DP_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	DPI [▶ 629]		DIQ = Double-point information with quality descriptor
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.4 Step position information

4.1.2.4.1 M_ST_NA_1

Step position information without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x05 (5)	= M_ST_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	VTI [▶ 238]								Value with transient state indication
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.4.2 M_ST_TA_1

Step position information with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x06 (6)	= M_ST_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	VTI [▶ 238]								Value with transient state indication
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[2..4] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.4.3 M_ST_TB_1

Step position information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x20 (32)	= M_ST_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 9									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	VTI [▶ 238]								Value with transient state indication
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[2..8] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[9..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.5 Bitstring of 32 bits

4.1.2.5.1 M_BO_NA_1

Bitstring of 32 bits without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x07 (7)	= M_BO_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS [▶ 392] = Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.5.2 M_BO_TA_1

Bitstring of 32 bits with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x08 (8)	= M_BO_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	BSI [▶ 239]								Binary state information
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS [▶ 392] = Quality descriptor
			|---	data[5...7] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[8...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.5.3 M_BO_TB_1

Bitstring of 32 bits with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x21 (33)	= M_BO_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS [▶ 392] = Quality descriptor
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.6 Measured value, normalized value

4.1.2.6.1 M_ME_NA_1

Measured value, normalized value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x09 (9)	= M_ME_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.6.2 M_ME_TA_1

Measured value, normalized value with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0A (10)	= M_ME_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 6									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..5] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[6..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.6.3 M_ME_TD_1

Measured value, normalized value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x22 (34)	= M_ME_TD_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 10									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value;
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.6.4 M_ME_ND_1

Measured value, normalized value without quality descriptor.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x15 (21)	= M_ME_ND_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.7 Measured value, scaled value

4.1.2.7.1 M_ME_NB_1

Measured value, scaled value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0B (11)	= M_ME_NB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.7.2 M_ME_TB_1

Measured value, scaled value with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0C (12)	= M_ME_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 6									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..5] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[6..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.7.3 M_ME_TE_1

Measured value, scaled value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x23 (35)	= M_ME_TE_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.8 Measured value, short floating point value

4.1.2.8.1 M_ME_NC_1

Measured value, short floating point value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0D (13)	= M_ME_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.8.2 M_ME_TC_1

Measured value, short floating point value with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0E (14)	= M_ME_TC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[5...7] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[8...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.8.3 M_ME_TF_1

Measured value, short floating point value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x24 (36)	= M_ME_TF_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.9 Integrated totals

4.1.2.9.1 M_IT_NA_1

Integrated total without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0F (15)	= M_IT_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BCR [▶ 239]								Binary counter reading
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence					Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.9.2 M_IT_TA_1

Integrated total with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x10 (16)	= M_IT_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	BCR [▶ 239]								Binary counter reading
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence					Quality descriptor
			|---	data[5...7] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[8...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.9.3 M_IT_TB_1

Integrated total with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x25 (37)	= M_IT_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	BCR [▶ 239]								Binary counter reading
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence					Quality descriptor
			|---	data[5...11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.10 Single command

4.1.2.10.1 C_SC_NA_1

Single command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2D (45)	= C_SC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					0	SCS [▶ 622]	SCO = Single command
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.10.2 C_SC_TA_1

Single command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3A (58)	= C_SC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					0	SCS [▶ 622]	SCO = Single command
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.11 Double command

4.1.2.11.1 C_DC_NA_1

Double command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2E (46)	= C_DC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					DCS [▶ 622]		DCO = Double command
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.11.2 C_DC_TA_1

Double command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3B (59)	= C_DC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					DCS [▶ 622]		DCO = Double command
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.12 Regulating step command

4.1.2.12.1 C_RC_NA_1

Regulating step command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2F (47)	= C_RC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					RCS [▶ 628]		RCO = Regulating step command
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.12.2 C_RC_TA_1

Regulating step command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3C (60)	= C_RC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					RCS [▶ 628]		RCO = Regulating step command
			|---	data[1...7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.13 Set-point command, normalized value

4.1.2.13.1 C_SE_NA_1

Set-point command, normalized value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x30 (48)	= C_SE_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.13.2 C_SE_TA_1

Set-point command, normalized value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3D (61)	= C_SE_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 10									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.14 Set-point command, scaled value

4.1.2.14.1 C_SE_NB_1

Set-point command, scaled value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x31 (49)	= C_SE_NB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.14.2 C_SE_TB_1

Set-point command, scaled value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3E (62)	= C_SE_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 10									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.15 Set-point command, short floating value

4.1.2.15.1 C_SE_NC_1

Set-point command, short floating point value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x32 (50)	= C_SE_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[5...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.15.2 C_SE_TC_1

Set-point command, short floating point value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3F (63)	= C_SE_TC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.16 Bitstring command

4.1.2.16.1 C_BO_NA_1

Bitstring of 32 bits without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x33 (51)	= C_BO_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 4									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.16.2 C_BO_TA_1

Bitstring of 32 bits with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x40 (64)	= C_BO_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 11									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4..10] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.17 Test command

4.1.2.17.1 C_TS_NA_1

Test command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x68 (104)	= C_TS_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] = 0xAA	1	0	1	0	1	0	1	0	FBP [▶ 239] = Fixed test pattern
			|---	data[1] = 0x55	0	1	0	1	0	1	0	1	
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.18 System information in monitor direction

4.1.2.18.1 M_EI_NA_1

End of initialization.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x46 (70)	= M_EI_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	LPC [▶ 238]	COI [▶ 623]							Cause of initialization
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.19 System information in control direction

4.1.2.19.1 C_CS_NA_1

Clock synchronisation command

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x67 (103)	= C_CS_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0. .6] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[7. .IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.19.2 C_IC_NA_1

Interrogation command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x64 (100)	= C_IC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	QOI [▶ 623]								Qualifier of interrogation
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.19.3 C_CI_NA_1

Counter interrogation command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x65 (101)	= C_CI_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	FRZ [▶ 624]		RQT [▶ 625]						QCC = Qualifier of counter interrogation
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.19.4 C_RP_NA_1

Reset process command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x69 (105)	= C_RP_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	QRP [▶ 625]								Qualifier of reset process
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.19.5 C_RD_NA_1

Read command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x66 (102)	= C_RD_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.20 Protection equipment information

4.1.2.20.1 M_EP_TA_1

Event of protection equipment with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x11 (17)	= M_EP_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 6									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	ES [▶ 626]		SEP = Single event of protection equipment
			|---	data[1..2] =	CP16Time2a [▶ 636]								Elapsed time, two octets binary time
			|---	data[3..5] =	CP24Time2a [▶ 636]								Three octets binary time tag
			|---	data[6..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.20.2 M_EP_TB_1

Packed start events of protection equipment with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x12 (18)	= M_EP_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	SRD	SIE	SL3	SL2	SL1	GS	SEP = Start events of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment
			|---	data[2..3] =	CP16Time2a [▶ 636]								Relay duration time, two octets binary time
			|---	data[4..6] =	CP24Time2a [▶ 636]								Three octets binary time tag

			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.1.2.20.3 M_EP_TC_1

Packed output circuit information of protection equipment with CP24Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x13 (19)	= M_EP_TC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 7								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	0	0	CL3	CL2	CL1	GC	OCI = Output cirquit information of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment
			|---	data[2..3] =	CP16Time2a [▶ 636]								Relay operating time, two octets binary time
			|---	data[4..6] =	CP24Time2a [▶ 636]								Three octets binary time tag

			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.1.2.20.4 M_EP_TD_1

Event of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x26 (38)	= M_EP_TD_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 10									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	ES [▶ 626]		SEP = Single event of protection equipment
			|---	data[1..2] =	CP16Time2a [▶ 636]								Elapsed time, two octets binary time
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.20.5 M_EP_TE_1

Packed start events of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x27 (39)	= M_EP_TE_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 11									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	SRD	SIE	SL3	SL2	SL1	GS	SEP = Start events of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment
			|---	data[2..3] =	CP16Time2a [▶ 636]								Relay duration time, two octets binary time
			|---	data[4..10] =	CP56Time2a [▶ 633]								Seven octets binary time tag

			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.1.2.20.6 M_EP_TF_1

Packed output circuit information of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x28 (40)	= M_EP_TF_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 11								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	0	0	CL3	CL2	CL1	GC	OCI = Output cirquit information of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment
			|---	data[2..3] =	CP16Time2a [▶ 636]								Relay operating time, two octets binary time
			|---	data[4..10] =	CP56Time2a [▶ 633]								Seven octets binary time tag

			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.1.2.20.7 M_PS_NA_1

Packed single point information with status change detection.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x14 (20)	= M_PS_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	SCD								Status and status change detection (32 bit)
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]				OV [▶ 392]	QDS: Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.21 Parameter loading/activation

4.1.2.21.1 P_ME_NA_1

Parameter of measured value, normalized value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x6E (110)	= P_ME_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.21.2 P_ME_NB_1

Parameter of measured value, scaled value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x6F (111)	= P_ME_NB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.21.3 P_ME_NC_1

Parameter of measured value, short floating point value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x70 (112)	= P_ME_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0...3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[5...IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.21.4 P_AC_NA_1

Parameter activation.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x71 (113)	= P_AC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	QPA [▶ 627]								Qualifier of parameter activation
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.1.2.22 Information elements

4.1.2.22.1 LPC

Local parameter change flag.

- <0> = No change;
- <1> = Changed;

4.1.2.22.2 VTI

Value with transient state indication (8 bits).

Transient state (bit 7):

- <0> = equipment is not in transient state;
- <1> = equipment is in transient state;

Value (bits 0..6) = <-64..63>;

4.1.2.22.3 BSI

Bitstring of 32 bits.

4.1.2.22.4 NVA

Normalized value.

4.1.2.22.5 SVA

Scaled value.

4.1.2.22.6 R32

Short floating point value.

4.1.2.22.7 BCR

Binary counter reading.

4.1.2.22.8 S/E

Select/execute state.

- <0> = Ausführen (execute);
- <1> = Anwählen (select);

4.1.2.22.9 FBP

Fixed test pattern.

4.1.2.22.10 TSC

Test command counter.

4.2 IEC60870-5-102

TF6500 implements one interface for the transport of the IEC 60870-5-102 ASDUs (service data unitof the application layer) via the serial interfaces of the pc and the Beckhoff KL6xxx/EL6xxx bus terminals.

The end application is imposed on the software interface of the PLC library. Unlike the implementation of other 101/104 TwinCAT products only one "so called" 'Low level' interface is available. The reason: The IEC 60870-5-102 devices are featuring a lot of vendor specific data. The 'Low level' interface allows the access to this data. The characteristics of both interfaces are described briefly below.

'Low level' interface:

This interface starts lower down on the protocol stack and enables the sending and/or receipt of individual frames (ASDUs). This interface is placed inside the protocol structure above the link layer (2) and implements necessary transport functions and frame formats. Application functions like for example the general request or time synchronisation are not implemented in the interface, but the user can implement them by himself.

Pro: Very flexible; all properties in the ASDU frame can be changed (e.g. an own time stamp, Select/ Execute or a special command execution etc.); high performance can be achieved because only the necessary services are implemented; high performance if there are many data points;

Contra: Larger amount of programming work; all services, such as general query, counter query, time synchronisation, command execution, spontaneous data transmission etc. must be implemented by the PLC programmer himself (programmed out); the PLC programmer must be familiar with the protocol standard.

Protocol structure of endsystem:

Protocol structure of the end system:

Selection of application functions from IEC 60870-5-5	User process
Selection of ASDU (service data units of the application layer) from IEC 60870-5-3, IEC 60870-5-4 and IEC 60870-5-102	Application layer (7)
N/A	Presentation layer (6)
	Session layer (5)
	Transport layer (4)
	Network layer (3)
unbalanced IEC 60870-5-2 IEC 60870-5-1 (FT 1.2)	Link layer (2)
EIA RS485, RS232 (V.24), Fiber Optics	Physical layer (1)



Layers 3 to 6 are not used.

4.2.1 IEC 60870-5-102 telegram structure

- FT 1.2 frame format is used;
- Frames with fixed and variable block length and single control characters are used;
- Hamming distance = 4;

Frame with variable length

This frame type is used to transmit user data between controlling and controlled station

byte\bit	7	6	5	4	3	2	1	0					
0	Start byte 1 (0x68)								Header		LPCI		LPDU
1	Block length												
2	Block length (copy)												
3	Start byte 2 (0x68)												
4	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field				
5	Link address fields												
	(0, 1 or 2 octets)												
	Type identification [▶ 605]								DATA UNIT IDENTIFIER		ASDU		
	SQ	Number of object											
	T	P/N	Cause of transmission [▶ 614]										
	ASDU address fields												
	(1 or 2 octets)												
	Record address												
	Information object address								Info-object				
	Information elements												
n-1	Checksum								Tail		LPCI		
n	Stop byte (0x16)												

Frame with fixed length

This frame type is used for link layer services. Sometimes used as an confirmation.

byte\bit	7	6	5	4	3	2	1	0	
0	Start byte 1 (0x10)								
1	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field
2	Link address								
	(0, 1 or 2 octets)								
n-1	Checksum								
n	Stop byte (0x16)								

Single character data

Single characters are used to confirm link layer and user data transmissions.

- 0xE5 (positive acknowledge);

4.2.2 ASDU object description

4.2.2.1 Standard IEC 60870-5-102 data types

Type	Dec	Hex	Description
ASDU_TYPEUNDEF_2	0	0x00	Not used
M_SP_TA_2 [▶ 245]	1	0x01	Single-point information with time tag
M_IT_TA_2 [▶ 247]	2	0x02	Accounting integrated totals, 4 octets each
M_IT_TB_2 [▶ 249]	3	0x03	Accounting integrated totals, 3 octets each
M_IT_TC_2 [▶ 251]	4	0x04	Accounting integrated totals, 2 octets each
M_IT_TD_2 [▶ 253]	5	0x05	Periodical reset accounting integrated totals, 4 octets each
M_IT_TE_2 [▶ 255]	6	0x06	Periodical reset accounting integrated totals, 3 octets each
M_IT_TF_2 [▶ 257]	7	0x07	Periodical reset accounting integrated totals, 2 octets each
M_IT_TG_2 [▶ 259]	8	0x08	Operational integrated totals, 4 octets each
M_IT_TH_2 [▶ 261]	9	0x09	Operational integrated totals, 3 octets each
M_IT_TI_2 [▶ 263]	10	0x0A	Operational integrated totals, 2 octets each
M_IT_TK_2 [▶ 265]	11	0x0B	Periodical reset operational integrated totals, 4 octets each
M_IT_TL_2 [▶ 267]	12	0x0C	Periodical reset operational integrated totals, 3 octets each
M_IT_TM_2 [▶ 269]	13	0x0D	Periodical reset operational integrated totals, 2 octets each
	14..69	0x0E..0x45	Reserved (standard area)
M_EI_NA_2 [▶ 271]	70	0x46	End of initialization
P_MP_NA_2 [▶ 273]	71	0x47	Manufacturer and product specification of integrated total DTE
M_TI_TA_2 [▶ 275]	72	0x48	Current system time of integrated total DTE
	73..99	0x49..0x63	Reserved (standard area)
C_RD_NA_2 [▶ 277]	100	0x64	Read manufacturer and product specification
C_SP_NA_2 [▶ 279]	101	0x65	Read record of single-point information with time tag
C_SP_NB_2 [▶ 281]	102	0x66	Read record of single-point information with time tag of a selected time range
C_TI_NA_2 [▶ 283]	103	0x67	Read current system time of integrated total DTE
C_CI_NA_2 [▶ 285]	104	0x68	Read accounting integrated totals of the oldest integration period
C_CI_NB_2 [▶ 287]	105	0x69	Read accounting integrated totals of the oldest integration period and of a selected range of addresses
C_CI_NC_2 [▶ 289]	106	0x6A	Read accounting integrated totals of a specific past integration period
C_CI_ND_2 [▶ 291]	107	0x6B	Read accounting integrated totals of a specific past integration period and of a selected range of addresses
C_CI_NE_2 [▶ 293]	108	0x6C	Read periodical reset accounting integrated totals of the oldest integration period
C_CI_NF_2 [▶ 295]	109	0x6D	Read periodical reset accounting integrated totals of the oldest integration period and of a selected range of addresses
C_CI_NG_2 [▶ 297]	110	0x6E	Read periodical reset accounting integrated totals of a specific past integration period
C_CI_NH_2 [▶ 299]	111	0x6F	Read periodical reset accounting integrated totals of a specific past integration period and of a selected range of addresses

Type	Dec	Hex	Description
C_CI_NI_2 [▶ 301]	112	0x70	Read operational integrated totals of the oldest integration period
C_CI_NK_2 [▶ 303]	113	0x71	Read operational integrated totals of the oldest integration period and of a selected range of addresses
C_CI_NL_2 [▶ 305]	114	0x72	Read operational integrated totals of a specific past integration period
C_CI_NM_2 [▶ 307]	115	0x73	Read operational integrated totals of a specific past integration period and of a selected range of addresses
C_CI_NN_2 [▶ 309]	116	0x74	Read periodical reset operational integrated totals of the oldest integration period
C_CI_NO_2 [▶ 311]	117	0x75	Read periodical reset operational integrated totals of the oldest integration period and of a selected range of addresses
C_CI_NP_2 [▶ 313]	118	0x76	Read periodical reset operational integrated totals of a specific past integration period
C_CI_NQ_2 [▶ 315]	119	0x77	Read periodical reset operational integrated totals of a specific past integration period and of a selected range of addresses
C_CI_NR_2 [▶ 317]	120	0x78	Read accounting integrated totals of a specific past integration period of a selected time range and of a selected range of addresses
C_CI_NS_2 [▶ 320]	121	0x79	Read periodical reset accounting integrated totals of a specific past integration period of a selected time range and of a selected range of addresses
C_CI_NT_2 [▶ 323]	122	0x7A	Read operational integrated totals of a specific past integration period of a selected time range and of a selected range of addresses
C_CI_NU_2 [▶ 326]	123	0x7B	Read periodical reset operational integrated totals of a specific past integration period of a selected time range and of a selected range of addresses
	124..127	0x7C..0x7F	Reserved (standard area)
M_DS_TA_2	128	0x80	-
P_ME_NA_2	129	0x81	Parameters of the measuring point
M_DS_TB_2	130	0x82	-
M_CH_TA_2	131	0x83	-
C_PK_2	132	0x84	Load private key
C_TA_VC_2	133	0x85	Read tariff information (current values)
C_TA_VM_2	134	0x86	Read tariff information (stored values)
M_TA_VC_2	135	0x87	Tariff information (current values)
M_TA_VM_2	136	0x88	Tariff information (stored values)
C_TA_CP_2	137	0x89	Close accounting period
M_IB_TG_2	139	0x8B	Block of operational integrated totals (absolute values)
M_IB_TK_2	140	0x8C	Block of periodical reset operational integrated totals (increment values)
C_RM_NA_2	141	0x8D	Read configuration data of the meter device
M_RM_NA_2	142	0x8E	Configuration of the meter device
C_MR_NA_2	143	0x8F	Change configuration data of the meter device
C_PC_NA_2	144	0x90	-
M_PC_NA_2	145	0x91	-
C_MC_NA_2	146	0x92	-

Type	Dec	Hex	Description
C_DF_NA_2	147	0x93	-
M_DF_NA_2	148	0x94	-
C_MF_NA_2	149	0x95	-
	150..179	0x96..0xB3	Reserved
C_DS_TA_2	180	0xB4	-
C_CS_TA_2	181	0xB5	Change date and time (Time syncronization)
C_PI_NA_2	182	0xB6	Read parameters of the measuring point
C_AC_NA_2	183	0xB7	Start session and send access key
C_DS_TB_2	184	0xB8	-
C_CH_TA_2	185	0xB9	-
C_MH_TA_2	186	0xBA	-
C_FS_NA_2	187	0xBB	Finish session
C_MP_NA_2	188	0xBC	-
C_CB_NT_2	189	0xBD	Read a block of operational integrated totals of a time period and a selected address
C_CB_UN_2	190	0xBE	Read a block of periodical reset operational integrated totals of a time period and a selected address
	191..255	0xBF..0xFF	Reserved

4.2.2.2 Process information in monitor direction

4.2.2.2.1 M_SP_TA_2

Single-point information with time tag.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x1 (1)	= M_SP_TA_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * 9									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	SPA [▶ 329]	Single-point address	Information object 1
			|---	data[1] =	SPQ [▶ 329] SPI [▶ 628]	SPQ = Single-point qualifier SPI = Single-point information	
			|---	data[2..8] =	CP56Time2b [▶ 634]	Seven octets binary time tag	
			|---	data[9..length - 1] =			Information object 2..i
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved	

4.2.2.2.2 M_IT_TA_2

Accounting integrated totals, 4 octets each

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x2 (2)	= M_IT_TA_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (6 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..4] =	CR4				Counter reading, 4 octets	
			|---	data[5] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[6] =	Signature				Signature (optional)	
			|---	data[7..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.3 M_IT_TB_2

Accounting integrated totals, 3 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x3 (3)	= M_IT_TB_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (5 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..3] =	CR3				Counter reading, 3 octets	
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[5] =	Signature				Signature (optional)	
			|---	data[6..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.4 M_IT_TC_2

Accounting integrated totals, 2 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x4 (4)	= M_IT_TC_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (4 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..2] =	CR2				Counter reading, 2 octets	
			|---	data[3] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[4] =	Signature				Signature (optional)	
			|---	data[5..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.5 M_IT_TD_2

Periodical reset accounting integrated totals, 4 octets each.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x5 (5)	= M_IT_TD_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 647]
	|---	rcdAddr			Record address
	|---	stream			Information element/ object data [▶ 587]
		|---	length	= i * (6 + [1 (signature)]) + 5	
		|---	data	7 6 5 4 3 2 1 0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..4] =	CR4				Counter reading, 4 octets	
			|---	data[5] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[6] =	Signature				Signature (optional)	
			|---	data[7..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.6 M_IT_TE_2

Periodical reset accounting integrated totals, 3 octets each.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6 (6)	= M_IT_TE_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 647]
	|---	rcdAddr			Record address
	|---	stream			Information element/ object data [▶ 587]
		|---	length	= i * (5 + [1 (signature)]) + 5	
		|---	data	7 6 5 4 3 2 1 0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..3] =	CR3				Counter reading, 3 octets	
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[5] =	Signature				Signature (optional)	
			|---	data[6..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.7 M_IT_TF_2

Periodical reset accounting integrated totals, 2 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x7 (7)	= M_IT_TF_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (4 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..2] =	CR2				Counter reading, 2 octets	
			|---	data[3] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[4] =	Signature				Signature (optional)	
			|---	data[5..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.8 M_IT_TG_2

Operational integrated totals, 4 octets each.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x8 (8)	= M_IT_TG_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 647]
	|---	rcdAddr			Record address
	|---	stream			Information element/ object data [▶ 587]
		|---	length	= i * (6 + [1 (signature)]) + 5	
		|---	data	7 6 5 4 3 2 1 0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..4] =	CR4				Counter reading, 4 octets	
			|---	data[5] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[6] =	Signature				Signature (optional)	
			|---	data[7..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.9 M_IT_TH_2

Operational integrated totals, 3 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x9 (9)	= M_IT_TH_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (5 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA			Information object address	Information object 1
			|---	data[1..3] =	CR3			Counter reading, 3 octets	
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[5] =	Signature				Signature (optional)	
			|---	data[6..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.10 M_IT_TI_2

Operational integrated totals, 2 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0xA (10)	= M_IT_TI_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (4 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..2] =	CR2				Counter reading, 2 octets	
			|---	data[3] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[4] =	Signature				Signature (optional)	
			|---	data[5..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.11 M_IT_TK_2

Periodical reset operational integrated totals, 4 octets each.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0xB (11)	= M_IT_TK_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 647]
	|---	rcdAddr			Record address
	|---	stream			Information element/ object data [▶ 587]
		|---	length	= i * (6 + [1 (signature)]) + 5	
		|---	data	7 6 5 4 3 2 1 0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..4] =	CR4				Counter reading, 4 octets	
			|---	data[5] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[6] =	Signature				Signature (optional)	
			|---	data[7..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.12 M_IT_TL_2

Periodical reset operational integrated totals, 3 octets each.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0xC (12)	= M_IT_TL_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 647]
	|---	rcdAddr			Record address
	|---	stream			Information element/ object data [▶ 587]
		|---	length	= i * (5 + [1 (signature)]) + 5	
		|---	data	7 6 5 4 3 2 1 0	

			|---	data[0] =	IOA			Information object address	Information object 1
			|---	data[1..3] =	CR3			Counter reading, 3 octets	
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[5] =	Signature				Signature (optional)	
			|---	data[6..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.2.13 M_IT_TM_2

Periodical reset operational integrated totals, 2 octets each.

- obj													ASDU object [▶ 646]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0xD (13)	= M_IT_TM_2									Type identification [▶ 605]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= i									Number of objects
|	|---	bT											Test
|	|---	bPN											Positive/ negative confirmation/ activation
|	|---	eCOT											Cause of transmission [▶ 614]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= i * (4 + [1 (signature)]) + 5									
		|---	data		7	6	5	4	3	2	1	0	

			|---	data[0] =	IOA				Information object address	Information object 1
			|---	data[1..2] =	CR2				Counter reading, 2 octets	
			|---	data[3] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence	Quality descriptor	
			|---	data[4] =	Signature				Signature (optional)	
			|---	data[5..length - 6] =						Information object 2..i
			|---	data[length - 5..length - 1] =	CP40Time2a [▶ 635]				Five octets binary time tag	
			|---	data[length..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved	

4.2.2.3 System information in monitor direction

4.2.2.3.1 M_EI_NA_2

End of initialization.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x46 (70)	= M_EI_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr		= 0									Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0								Address of information object
			|---	data[1] =	LPC [▶ 238]	COI [▶ 623]							COI = Cause of initialization
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.3.2 P_MP_NA_2

Manufacturer and product specification of integrated total DTE.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x47 (71)	= P_MP_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr		= 0									Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 6									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	YEAR				MONTH				Date of standard
			|---	data[1] =									Manufacturer code
			|---	data[2..5] =									Product code
			|---	data[6..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.3.3 M_TI_TA_2

Current system time of integrated total DTE.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x48 (72)	= M_TI_TA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr		= 0									Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..6] =	CP56Time2b [▶ 634]								Seven octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4 System information in control direction

4.2.2.4.1 C_RD_NA_2

Read manufacturer and product specification.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x64 (100)	= C_RD_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr		= 0									Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.2 C_SP_NA_2

Read record of single-point information with time tag.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x65 (101)	= C_SP_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.3 C_SP_NB_2

Read record of single-point information with time tag of selected time range.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x66 (102)	= C_SP_NB_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 10									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..4] =	CP40Time2a [▶ 635]								Five octets binary time tag (from)
			|---	data[5..9] =	CP40Time2a [▶ 635]								Five octets binary time tag (until)
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.4 C_TI_NA_2

Read current system time of integrated total data terminal equipment.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x67 (103)	= C_TI_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr		= 0									Record address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.5 C_CI_NA_2

Read accounting integrated totals of the oldest integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x68 (104)	= C_CI_NA_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.6 C_CI_NB_2

Read accounting integrated totals of the oldest integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x69 (105)	= C_CI_NB_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.7 C_CI_NC_2

Read accounting integrated totals of a specific past integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6A (106)	= C_CI_NC_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..4] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.8 C_CI_ND_2

Read accounting integrated totals of a specific past integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6B (107)	= C_CI_ND_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.9 C_CI_NE_2

Read periodically reset accounting integrated totals of the oldest integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6C (108)	= C_CI_NE_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.10 C_CI_NF_2

Read periodically reset accounting integrated totals of the oldest integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6D (109)	= C_CI_NF_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.11 C_CI_NG_2

Read periodically reset accounting integrated totals of a specific past integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6E (110)	= C_CI_NG_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..4] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.12 C_CI_NH_2

Read periodically reset accounting integrated totals of a specific past integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x6F (111)	= C_CI_NH_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.13 C_CI_NI_2

Read operational integrated totals of the oldest integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x70 (112)	= C_CI_NI_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.14 C_CI_NK_2

Read operational integrated totals of the oldest integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x71 (113)	= C_CI_NK_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.15 C_CI_NL_2

Read operational integrated totals of a specific past integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x72 (114)	= C_CI_NL_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..4] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.16 C_CI_NM_2

Read operational integrated totals of a specific past integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x73 (115)	= C_CI_NM_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.17 C_CI_NN_2

Read periodically reset operational integrated totals of the oldest integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x74 (116)	= C_CI_NN_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 0	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.18 C_CI_NO_2

Read periodically reset operational integrated totals of the oldest integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x75 (117)	= C_CI_NO_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.19 C_CI_NP_2

Read periodically reset operational integrated totals of a specific past integration period.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x76 (118)	= C_CI_NP_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..4] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.20 C_CI_NQ_2

Read periodically reset operational integrated totals of a specific past integration period and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x77 (119)	= C_CI_NQ_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.2.2.4.21 C_CI_NR_2

Read accounting integrated totals of selected time and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x78 (120)	= C_CI_NR_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag (from)
			|---	data[7..11] =	CP40Time2a [▶ 635]								Five octets binary time tag (to)

			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.2.2.4.22 C_CI_NS_2

Read periodically reset accounting integrated totals of selected time and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x79 (121)	= C_CI_NS_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag (from)
			|---	data[7..11] =	CP40Time2a [▶ 635]								Five octets binary time tag (to)

			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.2.2.4.23 C_CI_NT_2

Read operational integrated totals of selected time and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x7A (122)	= C_CI_NT_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag (from)
			|---	data[7..11] =	CP40Time2a [▶ 635]								Five octets binary time tag (to)

			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.2.2.4.24 C_CI_NU_2

Read periodically reset operational integrated totals of selected time and of selected range of addresses.

- obj					ASDU object [▶ 646]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 647]
|	|---	eType	= 0x7B (123)	= C_CI_NU_2	Type identification [▶ 605]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation / activation
|	|---	eCOT			Cause of transmission [▶ 614]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]

|---	- info												INFORMATION OBJECT [▶ 647]
	|---	rcdAddr											Record address of accounting period
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =									Integrated total address (from)
			|---	data[1] =									Integrated total address (to)
			|---	data[2..6] =	CP40Time2a [▶ 635]								Five octets binary time tag (from)
			|---	data[7..11] =	CP40Time2a [▶ 635]								Five octets binary time tag (to)

			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.2.2.5 Information elements

4.2.2.5.1 SPA

Single-point address: <0..255>

- <0..127> compatible range
- <128..255> private range

4.2.2.5.2 SPQ

Single-point qualifier: <0..127>

- <0> not specified
- <1..127> vendor specific

4.2.2.5.3 Record address (rcdAddr)

The list addresses are specified as follows: <0..255>

- <0> Default, if no other value is specified
- <1> List address for counter states from the beginning of the accounting period
- <2..10> Reserved for further compatible specifications
- <11> List address for counter states from measurement period 1
- <12> List address for counter states from measurement period 2
- <13> List address for counter states from measurement period 3
- <14..20> Reserved for further compatible specifications
- <21> List address for counter states (daily values) from measurement period 1
- <22> List address for counter states (daily values) from measurement period 2
- <23> List address for counter states (daily values) from measurement period 3
- <24..30> Reserved for further compatible specifications
- <31> List address for counter states (monthly values) from measurement period 1
- <32> List address for counter states (monthly values) from measurement period 2
- <33> List address for counter states (monthly values) from measurement period 3
- <34..40> Reserved for further compatible specifications
- <41> List address for counter states (yearly values) from measurement period 1
- <42> List address for counter states (yearly values) from measurement period 2
- <43> List address for counter states (yearly values) from measurement period 3
- <44..49> Reserved for further compatible specifications
- <50> Oldest single message
- <51> Complete list with single messages
- <52> Partial list 1 with single messages
- <53> Partial list 2 with single messages

- <54> Partial list 3 with single messages
- <55> Partial list 4 with single messages
- <56..127> Reserved for further compatible specifications
- <128 - 255> For particular applications (private area)

The size of a partial list is a system parameter.

4.3 IEC60870-5-103

TF6500 implements one interface for the transport of the IEC 60870-5-103 ASDUs (service data unitof the application layer) via the serial interfaces of the pc and the Beckhoff KL6xxx/EL6xxx bus terminals.

The end application is imposed on the software interface of the PLC library. Unlike the implementation of other 101/104 TwinCAT products only one "so called" 'Low level' interface is available. The reason: The IEC 60870-5-102 devices are featuring a lot of vendor specific data. The 'Low level' interface allows the access to this data. The characteristics of both interfaces are described briefly below.

'Low level' interface:

This interface starts lower down on the protocol stack and enables the sending and/or receipt of individual frames (ASDUs). This interface is placed inside the protocol structure above the link layer (2) and implements necessary transport functions and frame formats. Application functions like for example the general request are not implemented in the interface, but the user can implement them by himself.

Pro: Very flexible; all properties in the ASDU frame can be changed (e.g. an own time stamp, Select/ Execute or a special command execution etc.); high performance can be achieved because only the necessary services are implemented; high performance if there are many data points;

Contra: Larger amount of programming work; all services, such as general query, counter query, time synchronisation, command execution, spontaneous data transmission etc. must be implemented by the PLC programmer himself (programmed out); the PLC programmer must be familiar with the protocol standard.

Protocol structure of endsystem:

Protocol structure of the end system:

Selection of application functions from IEC 60870-5-5	User process
Selection of ASDU (service data units of the application layer) from IEC 60870-5-3, IEC 60870-5-4 and IEC 60870-5-103	Application layer (7)
N/A	Presentation layer (6)
	Session layer (5)
	Transport layer (4)
	Network layer (3)
unbalanced IEC 60870-5-2 IEC 60870-5-1 (FT 1.2)	Link layer (2)
EIA RS485, RS232 (V.24), Fiber Optics	Physical layer (1)



Layers 3 to 6 are not used.

4.3.1 IEC 60870-5-103 telegram structure

- FT 1.2 frame format is used;
- Frames with fixed and variable block length and single control characters are used;
- Hamming distance = 4;

Frame with variable length

This frame type is used to transmit user data between controlling and controlled station

byte\bit	7	6	5	4	3	2	1	0					
0	Start byte 1 (0x68)								Header		LPCI		LPDU
1	Block length												
2	Block length (copy)												
3	Start byte 2 (0x68)												
4	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field				
5	Link address												
6	Type identification (monitoring-direction [▶ 607], control-direction [▶ 607])								DATA UNIT IDENTIFIER		ASDU		
7	SQ	Number of object											
8	Cause of transmission (monitoring-direction [▶ 615], control-direction [▶ 616])												
9	ASDU address												
10	Function type [▶ 388]												
11	Information number [▶ 388]								Info-object				
	Information elements												
n-1	Checksum								Tail		LPCI		
n	Stop byte (0x16)												

Frame with fixed length

This frame type is used for link layer services. Sometimes used as an confirmation.

byte\bit	7	6	5	4	3	2	1	0	
0	Start byte 1 (0x10)								
1	RES	PRM	FCB / ACD	FCV / DFC	Function code				Control field
2	Link address								
3	Checksum								
4	Stop byte (0x16)								

Single character data

Single characters are used to confirm link layer and user data transmissions.

- 0xE5 (positive acknowledge);

4.3.2 ASDU object description

4.3.2.1 Standard IEC 60870-5-103 data types

Monitoring direction	Control direction	Dec	Hex	Description
M_TYPEUNDEF_3	C_TYPEUNDEF_3	0	0x00	Not used
M_TTM_TA_3 [▶ 335]	-	1	0x01	Time-tagged message
M_TMR_TA_3 [▶ 337]	-	2	0x02	Time-tagged message with relative time
M_MEI_NA_3 [▶ 339]	-	3	0x03	Measurands I
M_TME_TA_3 [▶ 342]	-	4	0x04	Time-tagged measurands with relative time
M_IRC_NA_3 [▶ 363]	-	5	0x05	Identification
M_SYN_TA_3 [▶ 367]	C_SYN_TA_3 [▶ 373]	6	0x06	Time synchronisation
-	C_IGI_NA_3 [▶ 375]	7	0x07	Initialization of general interrogation
M_TGI_NA_3 [▶ 365]	-	8	0x08	General interrogation
M_MEII_NA_3 [▶ 344]	-	9	0x09	Measurands II
M_GD_XA_3 [▶ 371]	C_GD_NA_3 [▶ 377]	10	0x0A	Generic data
M_GI_XA_3 [▶ 369]	-	11	0x0B	Generic identification
-	-	12..19	0x0C..0x13	Reserved (standard area)
-	C_GRC_NA_3 [▶ 379]	20	0x14	General command
-	C_GC_NA_3 [▶ 381]	21	0x15	Generic command
-	-	22	0x16	Reserved (standard area)
M_LRD_TA_3 [▶ 348]	-	23	0x17	List of recorded disturbances
-	C_ODT_NA_3 [▶ 383]	24	0x18	Order for disturbance data transmission
-	C_ADT_NA_3 [▶ 385]	25	0x19	Acknowledgement for disturbance data transmission
M_RTD_TA_3 [▶ 350]	-	26	0x1A	Ready for transmission of disturbance data
M_RTC_NA_3 [▶ 352]	-	27	0x1B	Ready for transmission of channel
M_RTT_NA_3 [▶ 354]	-	28	0x1C	Ready for transmission of tags
M_TOT_NA_3 [▶ 356]	-	29	0x1D	Transmission of tags
M_TOV_NA_3 [▶ 358]	-	30	0x1E	Transmission of disturbance values
M_EOT_NA_3 [▶ 361]	-	31	0x1F	End of transmission
-	-	32..255	0x1F..0xFF	Reserved (user defined area)

4.3.2.2 Process information in monitor direction

4.3.2.2.1 M_TTM_TA_3

Time-tagged message.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x1 (1)	= M_TTM_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 6								
		|---	data		7	6	5	4	3	2	1 0	
			|---	data[0] =	0	0	0	0	0	0	DPI [▶ 629]	DPI = Double-point information
			|---	data[1..4] =	CP32Time2a [▶ 635]							Four octets binary time tag
			|---	data[5] =	SIN [▶ 392]							Supplementary information
			|---	data[6..IEC870_MAX_ASDU_DATA_BYTE] =								Reserved

4.3.2.2.2 M_TMR_TA_3

Time-tagged message with relative time.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x2 (2)	= M_TMR_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1 0	
			|---	data[0] =	0	0	0	0	0	0	DPI [▶ 629]	DPI = Double-point information
			|---	data[1..2] =	RET [▶ 391]							Relative time
			|---	data[3..4] =	FAN [▶ 390]							Fault number
			|---	data[5..8] =	CP32Time2a [▶ 635]							Four octets binary time tag
			|---	data[9] =	SIN [▶ 392]							Supplementary information
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =								Reserved

4.3.2.2.3 M_MEI_NA_3

Measurands I.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x3 (3)	= M_MEI_NA_3	Type identification [▶ 607]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 8 * i								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[1] =	MVAL [▶ 391]								Current L2
			|---	data[2] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[3] =	MVAL [▶ 391]								Voltage L1-L2
			|---	data[4] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[5] =	MVAL [▶ 391]								Active power P
			|---	data[6] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[7] =	MVAL [▶ 391]								Reactive power Q

			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.3.2.2.4 M_TME_TA_3

Time-tagged measurands with relative time

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x4 (4)	= M_TME_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 12									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short circuit location
			|---	data[4..5] =	RET [▶ 391]								Relative time
			|---	data[6..7] =	FAN [▶ 390]								Fault number
			|---	data[8..11] =	CP32Time2a [▶ 635]								Four octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.2.5 M_MEII_NA_3

Measurands II.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x9 (9)	= M_MEII_NA_3	Type identification [▶ 607]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= i	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 18 * i								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[1] =	MVAL [▶ 391]								Current L1
			|---	data[2] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[3] =	MVAL [▶ 391]								Current L2
			|---	data[4] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[5] =	MVAL [▶ 391]								Current L3
			|---	data[6] =						RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[7] =	MVAL [▶ 391]								Voltage L1-E

			|---	data[8] =		RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[9] =	MVAL [▶ 391]				Voltage L2-E
			|---	data[10] =		RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[11] =	MVAL [▶ 391]				Voltage L3-E
			|---	data[12] =		RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[13] =	MVAL [▶ 391]				Active power P
			|---	data[14] =		RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[15] =	MVAL [▶ 391]				Reactive power Q
			|---	data[16] =		RES [▶ 393]	ER [▶ 393]	OV [▶ 392]	MEA = Measurand with quality descriptor
			|---	data[17] =	MVAL [▶ 391]				Frequency f
			|---	data[18..IEC870_MAX_ASDU_DATA_BYTE] =					Reserved

4.3.2.2.6 M_LRD_TA_3

List of recorded disturbances.

- obj													ASDU object [▶ 649]	
|---	+ head												Reserved	
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]	
|	|---	eType	= 0x17 (23)	= M_LRD_TA_3									Type identification [▶ 607]	
|	|---	bSQ		= FALSE									Sequence of information objects	
|	|---	nObj		= i									Number of objects	
|	|---	eCOT											Cause of transmission [▶ 615]	
|	|---	asduAddr											Common address of asdu	
|	|---	eClass											Fifo priority class [▶ 611]	
|---	- info												INFORMATION OBJECT [▶ 649]	
	|---	fc											Function code/type [▶ 388]	
	|---	n		= 0									not used	
	|---	stream											Information element/ object data [▶ 587]	
		|---	length	= i * 10										
		|---	data		7	6	5	4	3	2	1	0		
			|---	data[0..1] =	FAN [▶ 390]								Fault number	Dataset 1
			|---	data[2] =	RES				OTEV	TEST	TM	TP	SOF = Status of fault	
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag	

			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =			Dataset 2..i

4.3.2.2.7 M_RTD_TA_3

Ready for transmission of disturbance data.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	0x1A (26)	= M_RTD_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 15									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0								not used
			|---	data[1] =	TOV [▶ 392]								Type of disturbance values
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4..5] =	NOF [▶ 391]								Number of grid faults
			|---	data[6] =	NOC [▶ 391]								Number of channels
			|---	data[7..8] =	NOE [▶ 391]								Number of information elements of a channel
			|---	data[9..10] =	INT [▶ 390]								Interval
			|---	data[11..14] =	CP32Time2a [▶ 635]								Four octets binary time tag
			|---	data[15..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.2.8 M_RTC_NA_3

Ready for transmission of channel.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x1B (27)	= M_RTC_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 17									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0								not used
			|---	data[1] =	TOV [▶ 392]								Type of disturbance value
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4] =	ACC [▶ 388]								Actual channel
			|---	data[5..8] =	R32 [▶ 239]								RPV = Rated primary value
			|---	data[9..12] =	R32 [▶ 239]								RSV = Rated secondary value
			|---	data[13..16] =	R32 [▶ 239]								RFA = Reference factor
			|---	data[17..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.2.9 M_RTT_NA_3

Ready for transmission of tags.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	0x1C (28)	= M_RTT_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 4									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0								not used
			|---	data[1] =	0								not used
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.2.10 M_TOT_NA_3

Transmission of tags.

- obj													ASDU object [▶ 649]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x1D (29)	= M_TOT_NA_3									Type identification [▶ 607]
|	|---	bSQ		= TRUE									Sequence of information objects
|	|---	nObj		= 1									Number of objects
|	|---	eCOT											Cause of transmission [▶ 615]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 649]
	|---	fc											Function code/type [▶ 388]
	|---	n		= 0									not used
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5 + (i * 3)									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	FAN [▶ 390]								Fault number
			|---	data[2] =	NOT [▶ 391]								Number of tags
			|---	data[3..4] =	TAP [▶ 392]								Tag position

			|---	data[5] =		Function code/ type	Tag 1
			|---	data[6] =		Information number	
			|---	data[7] =	0 DPI [▶ 629]	Double point information	
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =			Tag 2..i

4.3.2.2.11 M_TOV_NA_3

Transmission of disturbance values.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x1E (30)	= M_TOV_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8 + (i *2)									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0								not used
			|---	data[1] =	TOV [▶ 392]								Type of disturbance values
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4] =	ACC [▶ 388]								Actual channel
			|---	data[5] =	NDV [▶ 391]								Number of relevant disturbance values per ASDU
			|---	data[6..7] =	NFE [▶ 391]								Number of the ASDU's first information element
			|---	data[8..9] =	SDV [▶ 392]								Single disturbance value 1
			|---	data[10..11] =	SDV [▶ 392]								Single disturbance value 2

			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =	SDV [▶ 392]	Single disturbance value 3..i

4.3.2.2.12 M_EOT_NA_3

End of transmission.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x1F (31)	= M_EOT_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	TOO [▶ 392]								Type of order
			|---	data[1] =	TOV [▶ 392]								Type of disturbance values
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4] =	ACC [▶ 388]								Actual channel
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.3 System information in monitor direction

4.3.2.3.1 M_IRC_NA_3

Identification.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x5 (5)	= M_IRC_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 13								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	COL [▶ 389]								Compatibility level
			|---	data[1..8] =	ASC [▶ 389]								ASCII characters
			|---	data[9..12] = 0x20 if not used	free								Internal vendor identification
			|---	data[13..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.3.2 M_TGI_NA_3

Termination of general interrogation.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x8 (8)	= M_TGI_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc		= GLB	Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	SCN [▶ 392]								Scan number
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.3.3 M_SYN_TA_3

Time synchronisation.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x6 (6)	= M_SYN_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 615]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc		= GLB	Function code/ type [▶ 388]

	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0. .6] =	CP56Time2a [▶ 633]								Seven octets binary time format
			|---	data[7. .IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.3.4 M_GI_XA_3

Generic identification.

- obj													ASDU object [▶ 649]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0xB (11)	= M_GI_XA_3									Type identification [▶ 607]
|	|---	bSQ		= TRUE									Sequence of information objects
|	|---	nObj		= 1									Number of objects
|	|---	eCOT											Cause of transmission [▶ 615]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 649]
	|---	fc		GEN									Function code/type [▶ 388]
	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= variable									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	RII [▶ 391]								Return information identifier
			|---	data[1] =	GROUP [▶ 390]								GIN = Generic identification number
			|---	data[2] =	ENTRY [▶ 390]								
			|---	data[3] =	CONT [▶ 389]	COUNT [▶ 389]	NO [▶ 391]						NDE = Number of descriptive elements

			|---	data[4] =	KOD [▶ 390]	Kind of description	Element 1
			|---	data[5] =	DATATYPE [▶ 389]	GDD = Generic data description	
			|---	data[6] =	DATASIZE [▶ 389]		
			|---	data[7] =	CONT [▶ 389] NUMBER [▶ 392]		
			|---	data[8..8+(DATASIZE*NUMBER)] =	GID [▶ 392]	Generic identification data	
			|---	data[..IEC870_MAX_ASDU_DATA_BYTE] =			Element 2..i

4.3.2.3.5 M_GD_XA_3

Generic data.

- obj													ASDU object [▶ 649]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0xA (10)	= M_GD_XA_3									Type identification [▶ 607]
|	|---	bSQ		= TRUE									Sequence of information objects
|	|---	nObj		= 1									Number of objects
|	|---	eCOT											Cause of transmission [▶ 615]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 649]
	|---	fc		GEN									Function code/type [▶ 388]
	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= variable									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	RII [▶ 391]								Return information identifier
			|---	data[1] =	CONT [▶ 389]	COUNT [▶ 389]	NO [▶ 391]						NGD = Number of generic data sets

			|---	data[2] =	GROUP [▶ 390]	GIN = Generic identification number	Dataset 1
			|---	data[3] =	ENTRY [▶ 390]		
			|---	data[4] =	KOD [▶ 390]	Kind of description	
			|---	data[5] =	DATATYPE [▶ 389]	GDD = Generic data description	
			|---	data[6] =	DATASIZE [▶ 389]		
			|---	data[7] =	CONT [▶ 389] / NUMBER [▶ 392]		
			|---	data[8..8+(DATASIZE*NUMBER)] =	GID [▶ 392]	Generic identification data	
			|---	data[..IEC870_MAX_ASDU_DATA_BYTE] =			Dataset 2..i

4.3.2.4 System information in control dircetion

4.3.2.4.1 C_SYN_TA_3

Time synchronisation.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x6 (6)	= C_SYN_TA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 616]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc		:= GLB	Function code/ type [▶ 388]

	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0. .6] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[7. .IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.4.2 C_IGI_NA_3

General interrogation.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x7 (7)	= C_IGI_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 616]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc		:= GLB	Function code/ type [▶ 388]

	|---	n										Information number [▶ 388]
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	SCN [▶ 392]								Scan number
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.4.3 C_GD_NA_3

Generic data.

- obj													ASDU object [▶ 649]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x0A (10)	= C_GD_NA_3									Type identification [▶ 607]
|	|---	bSQ		= FALSE									Sequence of information objects
|	|---	nObj		= 1									Number of objects
|	|---	eCOT											Cause of transmission [▶ 616]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 649]
	|---	fc		:= GEN									Function code/type [▶ 388]
	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= variable									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	RII [▶ 391]								Return information identifier
			|---	data[1] =	CONT [▶ 389]	COUNT [▶ 389]	NO [▶ 391]						NGD = Number of generic data sets

			|---	data[2] =	GROUP [▶ 390]	GIN = Generic identification number	Dataset 1
			|---	data[3] =	ENTRY [▶ 390]		
			|---	data[4] =	KOD [▶ 390]	Kind of description	
			|---	data[5] =	DATATYPE [▶ 389]	GDD = Generic data description	
			|---	data[6] =	DATASIZE [▶ 389]		
			|---	data[7] =	CONT [▶ 389]]	NUMBER [▶ 392]	
			|---	data[8] =	GID [▶ 392]	Generic identification data	
			|---	data[9..IEC870_MAX_ASDU_DATA_BYTE] =			Dataset 2..i

4.3.2.4.4 C_GRC_NA_3

General command.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x14 (20)	= C_GRC_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 616]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc		:= GLB	Function code/ type [▶ 388]

	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =							DCS [▶ 622]		DCO = Double command
			|---	data[1] =	RII [▶ 391]								Return information identifier
			|---	data[2. .IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.3.2.4.5 C_GC_NA_3

Generic command.

- obj													ASDU object [▶ 649]
|---	+ head												Reserved
|---	- ident												DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x15 (21)	= C_GC_NA_3									Type identification [▶ 607]
|	|---	bSQ		= TRUE									Sequence of information objects
|	|---	nObj		= 1									Number of objects
|	|---	eCOT											Cause of transmission [▶ 616]
|	|---	asduAddr											Common address of asdu
|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 649]
	|---	fc		:= GEN									Function code/type [▶ 388]
	|---	n											Information number [▶ 388]
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 2 + (i * 3)									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	RII [▶ 391]								Return information identification
			|---	data[1] =	NOG [▶ 391]								Number of generic data sets

			|---	data[2] =	GROUP [▶ 390]	GIN = Generic identification number	Dataset 1
			|---	data[3] =	ENTRY [▶ 390]		
			|---	data[4] =	KOD [▶ 390]	Kind of description	
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =			Dataset 2..i

4.3.2.4.6 C_ODT_NA_3

Order of disturbance data transmission.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x18 (24)	= C_ODT_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 616]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	TOO [▶ 392]								Type of order
			|---	data[1] =	TOV [▶ 392]								Type of disturbance value
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4] =	ACC [▶ 388]								Actual channel
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									

4.3.2.4.7 C_ADT_NA_3

Acknowledgement for disturbance data transmission.

- obj					ASDU object [▶ 649]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 650]
|	|---	eType	= 0x19 (25)	= C_ADT_NA_3	Type identification [▶ 607]
|	|---	bSQ		= TRUE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	eCOT			Cause of transmission [▶ 616]
|	|---	asduAddr			Common address of asdu
|	|---	eClass			Fifo priority class [▶ 611]
|---	- info				INFORMATION OBJECT [▶ 649]
	|---	fc			Function code/ type [▶ 388]
	|---	n		= 0	not used

	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 5									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	TOO [▶ 392]								Type of order
			|---	data[1] =	TOV [▶ 392]								Type of disturbance value
			|---	data[2..3] =	FAN [▶ 390]								Fault number
			|---	data[4] =	ACC [▶ 388]								Actual channel
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									

4.3.2.5 Function type (code)

Typ	Beschreibung
0..127	Reserviert (privater Bereich)
128	Distanzschutz
129	Nicht benutzt (kompatibler Bereich)
130..143	Reserviert (privater Bereich)
144..145	Nicht benutzt (kompatibler Bereich)
146..159	Reserviert (privater Bereich)
160	Überstromzeitschutz
161	Nicht benutzt (kompatibler Bereich)
162..175	Reserviert (privater Bereich)
176	Transformator-Differentialschutz
177	Nicht benutzt (kompatibler Bereich)
178..191	Reserviert (privater Bereich)
192	Leitungsdifferentialschutz
193	Nicht benutzt (kompatibler Bereich)
194..207	Reserviert (privater Bereich)
208	Nicht benutzt (kompatibler Bereich)
209	Nicht benutzt (kompatibler Bereich)
210..223	Reserviert (privater Bereich)
224	Nicht benutzt (kompatibler Bereich)
225	Nicht benutzt (kompatibler Bereich)
226..239	Reserviert (privater Bereich)
240	Nicht benutzt (kompatibler Bereich)
241	Nicht benutzt (kompatibler Bereich)
242..253	Reserviert (privater Bereich)
254	Generische Funktion (GEN)
255	Globale Funktion (GLB)

4.3.2.6 Information number

Number	monitor direction	control direction
0..15	System functions	System functions
16..31	State	Common commands
32..47	Monitor	Not used
48..63	Ground leak	Not used
64..127	Short-circuits	Not used
128..143	Automatic restart	Not used
144..159	Operation measurements	Not used
160..239	Not used	Not used
240..255	Generic function	Generic function

4.3.2.7 Information elements

4.3.2.7.1 ACC

Actual channel: <0..255>

4.3.2.7.2 ASC

ASCII 8 bit code: <0..255>

4.3.2.7.3 COL

Compatibility level: <0..255>

4.3.2.7.4 CONT

Continuous flag

- <0> no following ASDU with the same RII;
- <1> following ASDU has the same RII;

4.3.2.7.5 COUNT

One bit counter for ASDU with equal RII: <0..1>

4.3.2.7.6 DATASIZE

Data size: <1..255>

4.3.2.7.7 DATATYPE

Data type: <0..255>

- <0> no data
- <1> OS8ASCII
- <2> PACKEDBITSTRING
- <3> UI
- <4> I
- <5> UF
- <6> F
- <7> R32
- <8> R64
- <9> Double poing information
- <10> Single poing information
- <11>
- <12>
- <13>
- <14>
- <15>
- <16>
- <17>
- <18>
- <19>
- <20>
- <21>

- <22>
- <23> data struct
- <24> index
- <25..255> reserved

4.3.2.7.8 ENTRY

Entry identification: <0..255>

4.3.2.7.9 FAN

Fault number: <0..65535>

4.3.2.7.10 GROUP

Group identification: <0..255>

4.3.2.7.11 INT

Interval: <1..65535> [µs]

4.3.2.7.12 KOD

Kind of description: <0..255>

- <0> no KOD specified
- <1> actual value
- <2> default value
- <3> range (min, max, step size)
- <4> reserved
- <5> precision
- <6> factor
- <7> % reference
- <8> enumeration
- <9> dimension
- <10> description
- <11> reserved
- <12> password entry
- <13> is read only
- <14> is write only
- <15> reserved
- <16> reserved
- <17> reserved
- <18> reserved
- <19> corresponding function type and information number
- <20> corresponding event
- <21> enumerated text array
- <22> enumerated value array
- <23> related entries

- <24..255> reserved

4.3.2.7.13 MVAL

Measured value:<-1..+1.-2E-12>

4.3.2.7.14 NDV

Number of relevant disturbance values per ASDU: <1..255>

- <1..25> used
- <26.255> not used

4.3.2.7.15 NFE

Number of the ASDU's first information element: <0..65535>

4.3.2.7.16 NO

Number of generic data sets: <0..63>

4.3.2.7.17 NOC

Number of channels: <0..255>

4.3.2.7.18 NOE

Number of information elements of a channel: <0..65535>

4.3.2.7.19 NOF

Number of grid faluts: <0..65535>

4.3.2.7.20 NOG

Number of generic identifications: <0..255>

4.3.2.7.21 NOT

Number of tags: <1..255>

4.3.2.7.22 RET

Relative time: <0..65535>

4.3.2.7.23 RII

Return information identifier.

USINT <0..255>

4.3.2.7.24 SCN

Scan number: <0..255>

4.3.2.7.25 SDV

Single disturbance value: <-1..+1-2E-15>

4.3.2.7.26 SIN

Supplementary information: <0..255>

4.3.2.7.27 TAP

Tag position: <0..65535>

4.3.2.7.28 TOO

Type of order: <1..255>

4.3.2.7.29 TOV

Type of disturbance value: <0..255>

4.3.2.7.30 GID

Generic identification data.

4.3.2.7.31 NUMBER

Number.

4.3.2.7.32 Other information elements

Quality descriptor

QDS

OV

Overflow quality flag:

- <0> = no overflow;
- <1> = overflow;

BL

Blocked quality flag:

- <0> = not blocked;

- <1> = blocked;

SB

Substituted quality flag:

- <0> = not substituted;
- <1> = substituted;

NT

Topical quality flag:

- <0> = topical;
- <1> = not topical;

IV

Invalid quality flag:

- <0> = valid;
- <1> = invalid;

CY

Carry flag:

- <0> = no carry;
- <1> = carry;

CA

Adjusted flag:

- <0> = Counter was not adjusted;
- <1> = Counter was adjusted;

EI

Elapsed flag:

- <0> = Elapsed time valid;
- <1> = Elapsed time not valid;

RES

ER

OTEV

TEST

TM

TP

4.4 IEC60870-5-104

TF6500 implements two interfaces for the transport of the IEC 60870-5-104 ASDUs (service data unit of the application layer) via TCP/IP.

'High level' interface:

This interface is a so-called ‘single-block solution’. All functions are encapsulated in one PLC block. The block implements the most important services and functions. This implementation is sufficient for over 90% of applications.

Pro: Very little PLC programming work is required in order to create an application; all services, such as general query, counter query, time synchronisation, command execution, spontaneous data transmission etc. are already implemented in the block and are executed automatically; the mapping of the IEC<->PLC process data and that of the data points is configured via function calls; the PLC programmer does not need to be very well acquainted with the protocol standard;

Contra: The PLC application has only a small influence on the execution of the protocol; no influence on the execution of the services – these are automatically implemented internally; time stamps are automatically generated by the block and cannot be changed (handed over by externals); only the direct command execution, for example, is possible; poorer performance if there are many data points.

This interface is recommended if you:

- are not familiar with the protocol standard;
- are implementing a simple application with few data points (<1000);
- are not placing any great performance demands on the application;
- are not sending any special command execution such as Select/Execute or data + time stamp from external devices;
- do not require any functions that are not supported according to the compatibility list;

'Low level' interface

This interface starts lower down on the protocol stack and enables the sending and/or receipt of individual frames (ASDUs). This interface is placed inside the protocol structure above the transport layer (4) and implements already the APCI functions (protocol control information of the application layer, see lower table). Application functions like for example the general request and count request are not implemented in the interface, but the user can implement them by himself.

Pro: Very flexible; all properties in the ASDU frame can be changed (e.g. an own time stamp, Select/Execute or a special command execution etc.); high performance can be achieved because only the necessary services are implemented; high performance if there are many data points;

Contra: Larger amount of programming work; all services, such as general query, counter query, time synchronisation, command execution, spontaneous data transmission etc. must be implemented by the PLC programmer himself (programmed out); the PLC programmer must be familiar with the protocol standard.

This interface is recommended if you:

- are familiar with the protocol standard;
- are implementing a protocol converter application;
- are implementing virtually all available standard functions in the application;
- are using special functions, such as the relaying of the time stamp from a Modbus device or the gaining of control over the command execution;
- require functions that are not supported according to the compatibility list;
- have many data points (>1000) and need high performance;

Protocol structure of endsystem:

Protocol structure of the end system:

Selection of application functions from IEC 60870-5-5 according to IEC 60870-5-101	User process
Selection of ASDU (service data units of the application layer) from IEC 60870-5-101 and IEC 60870-5-104	Application layer (7)
APCI (protocol control information of the application layer) Transport interface (user to TCP interface)	
Selection from the TCP/IP protocol collection (RFC 2200)	Transport layer (4)
	Network layer (3)
	Link layer (2)
	Physical layer (1)



Layers 5 and 6 are not used.

4.4.1 IEC 60870-5-104 telegram structure

APCI = Application Protocol Control Information

ASDU = Application Service Data Unit

APDU = Application Protocol Data Unit

Telegram format with variable length

This frame type is used to transmit user data between controlling and controlled station

byte\bit	7	6	5	4	3	2	1	0				
0	Start byte (0x68)									APCI		APDU
1	Length of the APDU (max. 253)											
2	Control field 1											
3	Control field 2											
4	Control field 3											
5	Control field 4											
6	Type identification [▶ 601]									ASDU		
7	SQ	Number of objects										
8	T	P/N	Cause of transmission (COT) [▶ 611]									
9	Originator address (ORG)											
10	ASDU address fields											
11	(2 octets)											
12	Information object address fields (IOA)											
13	(3 octets)											
14												
15	Object information											
...												
...												
...												
n-1												
n												

Telegram format with fixed length

byte\bit	7	6	5	4	3	2	1	0		
0	Start byte (0x68)									APCI
1	4 (Length of the APDU)									
2	Control field 1									
3	Control field 2									
4	Control field 3									
5	Control field 4									

Control field formats

Two types of control field formats: I-Format, S-Format are used to perform numbered information transfer.

The third: U-Format control field is used to perform unnumbered link layer control functions.

I-Format

byte\bit	7	6	5	4	3	2	1	0
0	Send sequence number N(S) LSB							0
1	Send sequence number N(S) MSB							
2	Receive sequence number N(R) LSB							0
3	Receive sequence number N(R) MSB							

S-Format

byte\bit	7	6	5	4	3	2	1	0
0	0						0	1
1	0							
2	Receive sequence number N(R) LSB							0
3	Receive sequence number N(R) MSB							

U-Format

byte\bit	7	6	5	4	3	2	1	0
0	TESTFR		STOPDT		STARTDT		1	1
1	0							
2	0							
3	0							

Simple samples and explanations

104 substation configuration: COT = 2 octets (includes originator address), ASDU address = 2 octets, IOA address = 3 octets

Sample 1

68 0E 4E 14 7C 00 65 01 0A 00 0C 00 00 00 00 05

LPDU bytes	Explanation
68	Start byte
0E	Length of the APDU = 14
4E	Send sequence number N(S) LSB, bit 0 = 0 => I-Format
14	Send sequence number N(S) MSB
7C	Receive sequence number N(R) LSB
00	Receive sequence number N(R) MSB
65	Type identification: C_CI_NA_1 (counter interrogation command)
01	Number of objects = 1
0A	Cause of transmission = 10 (activation termination)
00	Originator address = 0
0C 00	Common ASDU address (2 octets) = 12 dec.
00 00 00	Object address (3 octets)
05	Counter interrogation request qualifier = 5 (general counter interrogation)

Sample 2

68 34 5A 14 7C 00 0B 07 03 00 0C 00 10 30 00 BE 09 00 11 30 00 90 09 00 0E 30 00 75 00 00 28 30 00 25 09 00 29 30 00 75 00 00 0F 30 00 0F 0A 00 2E 30 00 AE 05 00

LPDU bytes	Explanation
68	Start byte
34	Length of the APDU = 52
5A	Send sequence number N(S) LSB, bit 0 = 0 => I-Format
14	Send sequence number N(S) MSB
7C	Receive sequence number N(R) LSB
00	Receive sequence number N(R) MSB
0B	Type identification: M_ME_NB_1(measured value, scaled value)
07	Number of objects = 7
03	Cause of transmission = 3 (spontanous)
00	Originator address = 0
0C 00	Common ASDU address (2 octets) = 12 dec.
10 30 00	Object address (3 octets) of **first** information object
BE 09 00	Scaled value + QDS (quality descriptor) of **first** information object
11 30 00	Object address (3 octets) of **second** information object
90 09 00	Scaled value + QDS (quality descriptor) of **second** information object
0E 30 00	Object address (3 octets) of **third** information object
75 00 00	Scaled value + QDS (quality descriptor) of **third** information object
28 30 00 25 09 00 29 30 00 75 00 00 0F 30 00 0F 0A 00 2E 30 00 AE 05 00	Object address + Scaled value + QDS (quality descriptor) of information object four to seven

Sample 3

68 04 01 00 7E 14

LPDU bytes	Explanation
68	Start byte
04	Length of the APDU = 4
01	bits 2..7 reserved, bit 0 = 1 and bit 1 = 0 => S-Format
00	reserved
7E	Receive sequence number N(R) LSB
14	Receive sequence number N(R) MSB

4.4.2 ASDU object description

4.4.2.1 Standard IEC 60870-5-104 data types

Type	Dec	Hex	Description
ASDU_TYPEUNDEF	0	0x00	Not used
M_SP_NA_1 [▶ 120]	1	0x01	Single-point information
M_SP_TA_1	2	0x02	Single-point information with time tag
M_DP_NA_1 [▶ 126]	3	0x03	Double-point information
M_DP_TA_1	4	0x04	Double-point information with time tag
M_ST_NA_1 [▶ 132]	5	0x05	Step position information
M_ST_TA_1	6	0x06	Step position information with time tag
M_BO_NA_1 [▶ 138]	7	0x07	Bitstring of 32 bit
M_BO_TA_1	8	0x08	Bitstring of 32 bit with time tag
M_ME_NA_1 [▶ 144]	9	0x09	Measured value, normalised value
M_ME_TA_1	10	0x0A	Measured value, normalized value with time tag
M_ME_NB_1 [▶ 152]	11	0x0B	Measured value, scaled value
M_ME_TB_1	12	0x0C	Measured value, scaled value wit time tag
M_ME_NC_1 [▶ 158]	13	0x0D	Measured value, short floating point number
M_ME_TC_1	14	0x0E	Measured value, short floating point number with time tag
M_IT_NA_1 [▶ 164]	15	0x0F	Integrated totals
M_IT_TA_1	16	0x10	Integrated totals with time tag
M_EP_TA_1	17	0x11	Event of protection equipment with time tag
M_EP_TB_1	18	0x12	Packed start events of protection equipment with time tag
M_EP_TC_1	19	0x13	Packed output circuit information of protection equipment with time tag
M_PS_NA_1 [▶ 228]	20	0x14	Packed single point information with status change detection
M_ME_ND_1 [▶ 150]	21	0x15	Measured value, normalized value without quality descriptor
ASDU_TYPE_22..29	22..29	0x16..0x1D	Reserved (standard area)
M_SP_TB_1 [▶ 124]	30	0x1E	Single-point information with time tag CP56Time2a
M_DP_TB_1 [▶ 130]	31	0x1F	Double-point information with time tag CP56Time2a
M_ST_TB_1 [▶ 136]	32	0x20	Step position information with time tag CP56Time2a
M_BO_TB_1 [▶ 142]	33	0x21	Bitstring of 32 bit with time tag CP56Time2a

Type	Dec	Hex	Description
M_ME_TD_1 [▶ 148]	34	0x22	Measured value, normalised value with time tag CP56Time2a
M_ME_TE_1 [▶ 156]	35	0x23	Measured value, scaled value with time tag CP56Time2a
M_ME_TF_1 [▶ 162]	36	0x24	Measured value, short floating point number with time tag CP56Time2a
M_IT_TB_1 [▶ 168]	37	0x25	Integrated totals with time tag CP56Time2a
M_EP_TD_1 [▶ 220]	38	0x26	Event of protection equipment with time tag CP56Time2a
M_EP_TE_1 [▶ 222]	39	0x27	Packed start events of protection equipment with time tag CP56Time2a
M_EP_TF_1 [▶ 225]	40	0x28	Packed output circuit information of protection equipment with time tag CP56Time2a
ASDU_TYPE_41..44	41..44	0x29..0x2C	Reserved (standard area)
C_SC_NA_1 [▶ 170]	45	0x2D	Single command
C_DC_NA_1 [▶ 174]	46	0x2E	Double command
C_RC_NA_1 [▶ 178]	47	0x2F	Regulating step command
C_SE_NA_1 [▶ 182]	48	0x30	Set-point Command, normalised value
C_SE_NB_1 [▶ 186]	49	0x31	Set-point Command, scaled value
C_SE_NC_1 [▶ 190]	50	0x32	Set-point Command, short floating point number
C_BO_NA_1 [▶ 194]	51	0x33	Bitstring 32 bit command
ASDU_TYPE_52..57	52..57	0x34..0x39	Reserved (standard area)
C_SC_TA_1 [▶ 172]	58	0x3A	Single command with time tag CP56Time2a
C_DC_TA_1 [▶ 176]	59	0x3B	Double command with time tag CP56Time2a
C_RC_TA_1 [▶ 180]	60	0x3C	Regulating step command with time tag CP56Time2a
C_SE_TA_1 [▶ 184]	61	0x3D	Measured value, normalised value command with time tag CP56Time2a
C_SE_TB_1 [▶ 188]	62	0x3E	Measured value, scaled value command with time tag CP56Time2a
C_SE_TC_1 [▶ 192]	63	0x3F	Measured value, short floating point number command with time tag CP56Time2a
C_BO_TA_1 [▶ 196]	64	0x40	Bitstring of 32 bit command with time tag CP56Time2a
ASDU_TYPE_65..69	65..69	0x41..0x45	Reserved (standard area)

Type	Dec	Hex	Description
M_EI_NA_1 [▶ 200]	70	0x46	End of Initialisation
ASDU_TYPE_71..99	71..99	0x47..0x63	Reserved (standard area)
C_IC_NA_1 [▶ 204]	100	0x64	Interrogation command
C_CI_NA_1 [▶ 206]	101	0x65	Counter interrogation command
C_RD_NA_1 [▶ 210]	102	0x66	Read command
C_CS_NA_1 [▶ 202]	103	0x67	Clock synchronisation command
C_TS_NA_1	104	0x68	Test command
C_RP_NA_1 [▶ 208]	105	0x69	Reset process command
C_CD_NA_1	106	0x6A	Delay acquisition command
C_TS_TA_1 [▶ 453]	107	0x6B	Test command with time tag CP56Time2a
ASDU_TYPE_108..109	108..109	0x6C..0x6D	Reserved (standard area)
P_ME_NA_1 [▶ 230]	110	0x6E	Parameter of measured values, normalized value
P_ME_NB_1 [▶ 232]	111	0x6F	Parameter of measured values, scaled value
P_ME_NC_1 [▶ 234]	112	0x70	Parameter of measured values, short floating point number
P_AC_NA_1 [▶ 236]	113	0x71	Parameter activation
ASDU_TYPE_114..119	114..119	0x72..0x77	Reserved (standard area)
F_FR_NA_1	120	0x78	File ready
F_SR_NA_1	121	0x79	Section ready
F_SC_NA_1	122	0x7A	Call directory, select file, call file, call section
F_LS_NA_1	123	0x7B	Last section, last segment
F_FA_NA_1	124	0x7C	ACK file, ACK section
F_SG_NA_1	125	0x7D	Segment
F_DR_TA_1	126	0x7E	Directory
ASDU_TYPE_127..255	127..255	0x7F..0xFF	Reserved (user area)

4.4.2.2 Single-point information

4.4.2.2.1 M_SP_NA_1

Single-point information without time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x01 (1)	= M_SP_NA_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	SPI 628 SIQ = Single-point information with quality descriptor
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =	Reserved							

4.4.2.2.2 M_SP_TB_1

Single-point information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x1E (30)	= M_SP_TB_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 8								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	SPI [▶ 628] SIQ = Single-point information with quality descriptor
			|---	data[1..7] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.3 Double-point information

4.4.2.3.1 M_DP_NA_1

Double-point information without time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x03 (3)	= M_DP_NA_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	DP	DIQ = Double-point information with quality descriptor
			|---	data[1...IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						



4.4.2.3.2 M_DP_TB_1

Double-point information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x1F (31)	= M_DP_TB_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 8								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	DP	DIQ = Double-point information with quality descriptor
			|---	data[1...7] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[8...IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						



4.4.2.4 Step position information

4.4.2.4.1 M_ST_NA_1

Step position information without time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x05 (5)	= M_ST_NA_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 2								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	VTI [▶ 238]	Value with transient state indication						
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392] QDS = Quality descriptor
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.4.2 M_ST_TB_1

Step position information with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x20 (32)	= M_ST_TB_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 9								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	VTI [▶ 238]	Value with transient state indication						
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392] QDS = Quality descriptor
			|---	data[2..8] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[9..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.5 Bitstring of 32 bits

4.4.2.5.1 M_BO_NA_1

Bitstring of 32 bits without time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x07 (7)	= M_BO_NA_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..3] =	BSI [▶ 239]	Binary state information						
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV	QDS [▶ 392] = Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.5.2 M_BO_TB_1

Bitstring of 32 bits with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x21 (33)	= M_BO_TB_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 12								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..3] =	BSI [▶ 239]	Binary state information						
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	QDS [▶ 392] = Quality descriptor
			|---	data[5..11] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.6 Measured value, normalized value

4.4.2.6.1 M_ME_NA_1

Measured value, normalized value without time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x09 (9)	= M_ME_NA_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..1] =	NVA [▶ 239]	Normalized value						
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392] QDS = Quality descriptor
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.6.2 M_ME_TD_1

Measured value, normalized value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x22 (34)	= M_ME_TD_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..1] =	NVA [▶ 239]	Normalized value;						
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392] QDS = Quality descriptor
			|---	data[3..9] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.6.3 M_ME_ND_1

Measured value, normalized value without quality descriptor.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x15 (21)	= M_ME_ND_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 2								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..1] =	NVA [▶ 239]	Normalized value						
			|---	data[2..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.7 Measured value, scaled value

4.4.2.7.1 M_ME_NB_1

Measured value, scaled value without time tag.

- obj					ASDU object [▶ 585]								
|---	+ head				Reserved								
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]								
|	|---	eType	= 0x0B (11)	= M_ME_NB_1	Type identification [▶ 601]								
|	|---	bSQ		= FALSE	Sequence of information objects								
|	|---	nObj		= 1	Number of objects								
|	|---	bT			Test								
|	|---	bPN			Positive/negative confirmation/activation								
|	|---	eCOT			Cause of transmission [▶ 611]								
|	|---	nORG			Originator address								
|	|---	asduAddr			Common address of asdu								
|	|---	eClass			Fifo priority class [▶ 611]								
|---	- info				INFORMATION OBJECT [▶ 587]								
	|---	objAddr			Information object address								
	|---	stream			Information element/object data [▶ 587]								
		|---	length	= 3									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]	Scaled value							
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV	QDS = Quality descriptor
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved							

4.4.2.7.2 M_ME_TE_1

Measured value, scaled value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]							
|---	+ head				Reserved							
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]							
|	|---	eType	= 0x23 (35)	= M_ME_TE_1	Type identification [▶ 601]							
|	|---	bSQ		= FALSE	Sequence of information objects							
|	|---	nObj		= 1	Number of objects							
|	|---	bT			Test							
|	|---	bPN			Positive/negative confirmation/activation							
|	|---	eCOT			Cause of transmission [▶ 611]							
|	|---	nORG			Originator address							
|	|---	asduAddr			Common address of asdu							
|	|---	eClass			Fifo priority class [▶ 611]							
|---	- info				INFORMATION OBJECT [▶ 587]							
	|---	objAddr			Information object address							
	|---	stream			Information element/object data [▶ 587]							
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0..1] =	SVA [▶ 239]	Scaled value						
			|---	data[2] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV QDS = Quality descriptor [▶ 392]
			|---	data[3..9] =	CP56Time2a [▶ 633]	Seven octets binary time tag						
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved						

4.4.2.8 Measured value, short floating point value

4.4.2.8.1 M_ME_NC_1

Measured value, short floating point value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0D (13)	= M_ME_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.8.2 M_ME_TF_1

Measured value, short floating point value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x24 (36)	= M_ME_TF_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 12								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	0	0	0	OV [▶ 392]	QDS = Quality descriptor
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.9 Integrated totals

4.4.2.9.1 M_IT_NA_1

Integrated total without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x0F (15)	= M_IT_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BCR [▶ 239]								Binary counter reading
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence					Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.9.2 M_IT_TB_1

Integrated total with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x25 (37)	= M_IT_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 12								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BCR [▶ 239]								Binary counter reading
			|---	data[4] =	IV [▶ 393]	CA [▶ 393]	CY [▶ 393]	Sequence					Quality descriptor
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.10 Single command

4.4.2.10.1 C_SC_NA_1

Single command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2D (45)	= C_SC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 1									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					0	SCS [▶ 622]	SCO = Single command
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.10.2 C_SC_TA_1

Single command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3A (58)	= C_SC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 8									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					0	SCS [▶ 622]	SCO = Single command
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.11 Double command

4.4.2.11.1 C_DC_NA_1

Double command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2E (46)	= C_DC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					DCS [▶ 622]		DCO = Double command
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.11.2 C_DC_TA_1

Double command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3B (59)	= C_DC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 8								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					DCS [▶ 622]		DCO = Double command
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.12 Regulating step command

4.4.2.12.1 C_RC_NA_1

Regulating step command without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x2F (47)	= C_RC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					RCS [▶ 628]		RCO = Regulating step command
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.12.2 C_RC_TA_1

Regulating step command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3C (60)	= C_RC_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 8								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	S/E [▶ 239]	QU [▶ 626]					RCS [▶ 628]		RCO = Regulating step command
			|---	data[1..7] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[8..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.13 Set-point command, normalized value

4.4.2.13.1 C_SE_NA_1

Set-point command, normalized value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x30 (48)	= C_SE_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.13.2 C_SE_TA_1

Set-point command, normalized value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3D (61)	= C_SE_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.14 Set-point command, scaled value

4.4.2.14.1 C_SE_NB_1

Set-point command, scaled value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x31 (49)	= C_SE_NB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.14.2 C_SE_TB_1

Set-point command, scaled value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3E (62)	= C_SE_TB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.15 Set-point command, short floating value

4.4.2.15.1 C_SE_NC_1

Set-point command, short floating point value without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x32 (50)	= C_SE_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of set-point command
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.15.2 C_SE_TC_1

Set-point command, short floating point value with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x3F (63)	= C_SE_TC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 12								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	S/E [▶ 239]	QL [▶ 624]							QOS = Qualifier of command
			|---	data[5..11] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[12..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.16 Bitstring command

4.4.2.16.1 C_BO_NA_1

Bitstring of 32 bits without time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x33 (51)	= C_BO_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 4								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.16.2 C_BO_TA_1

Bitstring of 32 bits with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x40 (64)	= C_BO_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 11								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	BSI [▶ 239]								Binary state information
			|---	data[4..10] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.17 Test command

4.4.2.17.1 C_TS_TA_1

Test command with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType		= C_TS_TA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 9									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	TSC [▶ 239]								Test counter
			|---	data[2..8] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[9..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.18 System information in monitor direction

4.4.2.18.1 M_EI_NA_1

End of initialization.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x46 (70)	= M_EI_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	LPC [▶ 238]	COI [▶ 623]							Cause of initialization
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.19 System information in control direction

4.4.2.19.1 C_CS_NA_1

Clock synchronisation command

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x67 (103)	= C_CS_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0									Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 7									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..6] =	CP56Time2a [▶ 633]								Seven octets binary time tag
			|---	data[7..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.19.2 C_IC_NA_1

Interrogation command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x64 (100)	= C_IC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	QOI [▶ 623]								Qualifier of interrogation
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.19.3 C_CI_NA_1

Counter interrogation command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x65 (101)	= C_CI_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	FRZ [▶ 624]		RQT [▶ 625]						QCC = Qualifier of counter interrogation
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.19.4 C_RP_NA_1

Reset process command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x69 (105)	= C_RP_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	QRP [▶ 625]								Qualifier of reset process
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.19.5 C_RD_NA_1

Read command.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x66 (102)	= C_RD_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 0									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.20 Protection equipment information

4.4.2.20.1 M_EP_TD_1

Event of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x26 (38)	= M_EP_TD_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 10								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	ES [▶ 626]		SEP = Single event of protection equipment
			|---	data[1..2] =	CP16Time2a [▶ 636]								Elapsed time, two octets binary time
			|---	data[3..9] =	CP56Time2a [▶ 633]								Seven t octets binary time tag
			|---	data[10..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.20.2 M_EP_TE_1

Packed start events of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x27 (39)	= M_EP_TE_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass											Fifo priority class [▶ 611]
|---	- info												INFORMATION OBJECT [▶ 587]
	|---	objAddr											Information object address
	|---	stream											Information element/ object data [▶ 587]
		|---	length	= 11									
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	SRD	SIE	SL3	SL2	SL1	GS	SEP = Start events of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment
			|---	data[2..3] =	CP16Time2a [▶ 636]								Relay duration time, two octets binary time

			|---	data[4..10] =	CP56Time2a [▶ 633]	Seven octets binary time tag
			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.4.2.20.3 M_EP_TF_1

Packed output circuit information of protection equipment with CP56Time2a time tag.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x28 (40)	= M_EP_TF_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 11								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0] =	0	0	0	0	CL3	CL2	CL1	GC	OCI = Output cirquit information of protection equipment
			|---	data[1] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]	EI [▶ 393]	0	0	0	QDP = Quality descriptor for events of protection equipment

			|---	data[2..3] =	CP16Time2a [▶ 636]	Relay operating time, two octets binary time
			|---	data[4..10] =	CP56Time2a [▶ 633]	Seven octets binary time tag
			|---	data[11..IEC870_MAX_ASDU_DATA_BYTE] =		Reserved

4.4.2.20.4 M_PS_NA_1

Packed single point information with status change detection.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x14 (20)	= M_PS_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	SCD								Status and status change detection (32 bit)
			|---	data[4] =	IV [▶ 393]	NT [▶ 393]	SB [▶ 393]	BL [▶ 392]				OV [▶ 392]	QDS: Quality descriptor
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.21 Parameter loading/activation

4.4.2.21.1 P_ME_NA_1

Parameter of measured value, normalized value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x6E (110)	= P_ME_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	NVA [▶ 239]								Normalized value
			|---	data[2] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.21.2 P_ME_NB_1

Parameter of measured value, scaled value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x6F (111)	= P_ME_NB_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 3								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..1] =	SVA [▶ 239]								Scaled value
			|---	data[2] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[3..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.21.3 P_ME_NC_1

Parameter of measured value, short floating point value.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x70 (112)	= P_ME_NC_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr										Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 5								
		|---	data		7	6	5	4	3	2	1	0	
			|---	data[0..3] =	R32 [▶ 239]								Short floating point value
			|---	data[4] =	LPC [▶ 238]	POP	KPA [▶ 627]						QPM: Qualifier of parameter of measured value
			|---	data[5..IEC870_MAX_ASDU_DATA_BYTE] =									Reserved

4.4.2.21.4 P_AC_NA_1

Parameter activation.

- obj					ASDU object [▶ 585]
|---	+ head				Reserved
|---	- ident				DATA UNIT IDENTIFIER [▶ 586]
|	|---	eType	= 0x71 (113)	= P_AC_NA_1	Type identification [▶ 601]
|	|---	bSQ		= FALSE	Sequence of information objects
|	|---	nObj		= 1	Number of objects
|	|---	bT			Test
|	|---	bPN			Positive/ negative confirmation/ activation
|	|---	eCOT			Cause of transmission [▶ 611]
|	|---	nORG			Originator address
|	|---	asduAddr			Common address of asdu

|	|---	eClass										Fifo priority class [▶ 611]
|---	- info											INFORMATION OBJECT [▶ 587]
	|---	objAddr		= 0								Information object address
	|---	stream										Information element/ object data [▶ 587]
		|---	length	= 1								
		|---	data		7	6	5	4	3	2	1	0
			|---	data[0] =	QPA [▶ 627]							Qualifier of parameter activation
			|---	data[1..IEC870_MAX_ASDU_DATA_BYTE] =								Reserved

4.4.2.22 Information elements

4.4.2.22.1 LPC

Local parameter change flag.

- <0> = No change;
- <1> = Changed;

4.4.2.22.2 VTI

Value with transient state indication (8 bits).

Transient state (bit 7):

- <0> = equipment is not in transient state;
- <1> = equipment is in transient state;

Value (bits 0..6) = <-64..63>;

4.4.2.22.3 BSI

Bitstring of 32 bits.

4.4.2.22.4 NVA

Normalized value.

4.4.2.22.5 SVA

Scaled value.

4.4.2.22.6 R32

Short floating point value.

4.4.2.22.7 BCR

Binary counter reading.

4.4.2.22.8 S/E

Select/execute state.

- <0> = Ausführen (execute);
- <1> = Anwählen (select);

4.4.2.22.9 FBP

Fixed test pattern.

4.4.2.22.10 TSC

Test command counter.

5 PLC Libraries

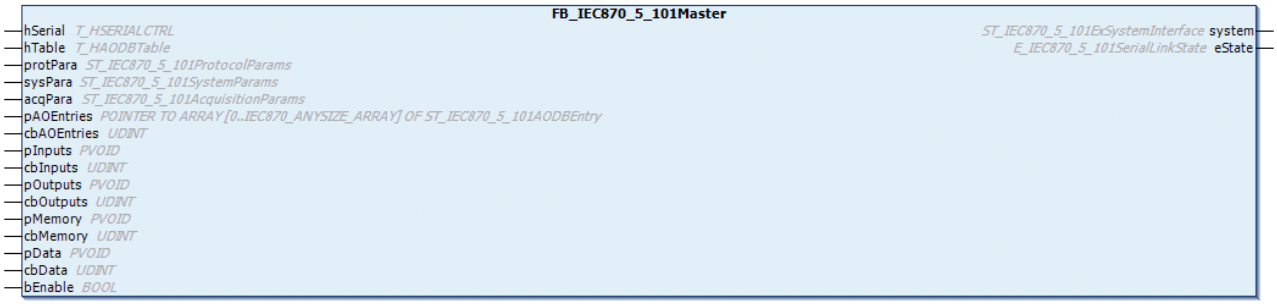
The central stations and/or substations (master/slaves) according to IEC60870-5-101/102/103/104 can be realised in the TwinCAT PLC with the PLC functions and function blocks.

The PLC library has two software interfaces. The end application is imposed on one of these interfaces. The choice of interface depends on the requirements for the end application.

5.1 Function blocks

5.1.1 High Level

5.1.1.1 FB_IEC870_5_101Master



An instance of the FB_IEC870_5_101Master function block can be used to implement an IEC60870-5-101 central station (master, High-Level) in the TwinCAT PLC. A connection to the slave is established for each instance of the function block. Normally the data exchange is started automatically once the connection is established. This is the default configuration of the function block.

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial : T_HSERIALCTRL;
    hTable  : T_HAODBTable;
END_VAR
```

hSerial: Connection handle [▶ 632] to the FB_IEC870_SerialLineCtrl [▶ 519] function block. This variable is used to exchange the data to be sent and received via the FB_IEC870_SerialLineCtrl function block.

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

VAR_INPUT

```
VAR_INPUT
    protPara        : ST_IEC870_5_101ProtocolParams :=  := ( eType := eIEC870_101_MASTER );
    sysPara         : ST_IEC870_5_101SystemParams := ( bEndOfInit := FALSE );
    acqPara         : ST_IEC870_5_101AcquisitionParams;
    pAOEntries      : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries     : UDINT := 0;
    pInputs         : PVOID := 0;
    cbInputs        : UDINT := 0;
    pOutputs        : PVOID := 0;
    cbOutputs       : UDINT := 0;
    pMemory         : PVOID := 0;
    cbMemory        : UDINT := 0;
    pData           : PVOID := 0;
    cbData          : UDINT := 0;
    bEnable         : BOOL := TRUE;
END_VAR
```

protPara: IEC 60870-5-101 protocol parameter [▶ 640].

sysPara: System parameter [▶ 588].

acqPara: Parameter for cyclic data acquisition [▶ 593].

pAOEntries: Address of the first application object database variable [▶ 584].

cbAOEntries: Byte size of the application object database variables.

pInputs: Address of the PLC process data area for the inputs.

cbInputs: Byte size of the PLC process data area for the inputs.

pOutputs: Address of the PLC process data area for the outputs.

cbOutputs: Byte size of the PLC process data area for the outputs.

pMamory: Address of the PLC process data area for the flags.

cbMamory: Byte size of the PLC process data area for the flags.

pData: Address of the PLC data area.

cbData: Byte size of the PLC data area.

bEnable: Activates/deactivates the function block (communication and connections).

The addresses can be determined with the ADR operator and the byte sizes with the SIZEOF operator.

VAR_OUTPUT

```
VAR_OUTPUT
    system  : ST_IEC870_5_101ExSystemInterface;
    eState  : E_IEC870_5_101SerialLinkState := eSERIALLINK_DISCONNECTED;
END_VAR
```

system: System interface [▶ 582]. This variable is used by other IEC application functions as a communication interface for the IEC device (here: central station).

- Member variable *system.device* is expected by the F_iecSetAOQuality [▶ 539] function as VAR_IN_OUT parameter, for example.
- Member variable *system.device.errors* is a device error FIFO. The PLC application can read and analyse registered errors.

eState: Connection status to the slave [▶ 630].

Examples (High-Level interface):

- IEC 60870-5-101 minimal central station (master) [▶ 666];
- IEC 60870-5-101 complex central station (master) [▶ 667];

Further examples:

The device error Fifo is read cyclically and the registered errors are written in the TwinCAT XAE->"Error List" window.

```
...
fbClient : FB_IEC870_5_101Master; (* IEC 60870-5-101 control station function block instance *)
...
```

```
...
REPEAT
    fbClient.system.device.errors.RemoveError();
    IF fbClient.system.device.errors.bOk THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'IEC 60870-5-101 master communication
 error: 0x%s',
                DWORD_TO_HEXSTR( fbClient.system.device.errors.getError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT fbClient.system.device.errors.bOk
END_REPEAT
...
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.1.2 FB_IEC870_5_101Slave



An instance of the function block FB_IEC870_5_101Slave can be used to implement an IEC 60870-5-101 substation (slave, high-level) in the TwinCAT PLC. Normally, the data exchange starts automatically once the connection has been established. This is the default configuration of the function block.

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial         : T_HSERIALCTRL;
END_VAR
```

hSerial: Connection handle [▶ 632] to FB_IEC870_SerialLineCtrl [▶ 519]. Via this variable the data to be send and received are exchanged with FB_IEC870_SerialLineCtrl.

VAR_INPUT

```
VAR_INPUT
    protPara        : ST_IEC870_5_101ProtocolParams;
    sysPara         : ST_IEC870_5_101SystemParams;
    pAOEntries      : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries     : UDINT := 0;
    pInputs         : PVOID := 0;
    cbInputs        : UDINT := 0;
    pOutputs        : PVOID := 0;
    cbOutputs       : UDINT := 0;
    pMemory         : PVOID := 0;
    cbMemory        : UDINT := 0;
    pData           : PVOID := 0;
    cbData          : UDINT := 0;
    bEnable         : BOOL := TRUE;
END_VAR
```

protPara: IEC 60870-5-101 protocol parameters. [▶ 640]

sysPara: System parameter. [▶ 588]

pAOEntries: Address of the first application object database variable. [▶ 584]

cbAOEntries: Byte size of the application object database variables.

pInputs: Address of the PLC process data area for the inputs.

cbInputs: Byte size of the PLC process data area for the inputs.

pOutputs: Address of the PLC process data area for the outputs.

cbOutputs: Byte size of the PLC process data area for the outputs.

pMemory: Address of the PLC process data area for the flags.

cbMemory: Byte size of the PLC process data area for the flags.

pData: Address of the PLC data area.

cbData: Byte size of the PLC data area.

bEnable: Activates/deactivates the function block (communication and connection).

The addresses can be determined with the ADR operator and the byte sizes with the SIZEOF operator.

VAR_OUTPUT

```
VAR_OUTPUT
    system      : ST_IEC870_5_101SystemInterface;
    eState      : E_IEC870_5_101SerialLinkState := eSERIALLINK_DISCONNECTED;
END_VAR
```

system: System interface [▶ 582]. This variable is used by other PLC functions or function blocks as communication interface with the IEC device (here: substation).

- Member variable *system.device* is expected by the F_iecSetAOQuality [▶ 539] function as VAR_IN_OUT parameter, for example.
- Member variable *system.device.errors* is a device error FIFO. The PLC application can read and analyses registered errors.

eState: Status of the connection [▶ 630] to the master.

Examples (high-level interface):

- IEC 60870-5-101 minimum substation (slave) [▶ 693];
- IEC 60870-5-101 complex substation (slave) [▶ 694];

Further examples:

```
PROGRAM test
VAR
    slave1AODB      : ARRAY[1..50] OF ST_IEC870_5_101AODBEntry;

    inputs AT%IB0   : ARRAY[0..999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..999] OF BYTE;
    memory AT%MB0   : ARRAY[0..999] OF BYTE;
    data            : ARRAY[0..999] OF BYTE;

    server1         : FB_IEC870_5_101Slave;

    bEnable         : BOOL;
    eState          : E_IEC870_5_101SerialLinkState;

    bError          : BOOL;
    iecError        : ST_IEC870_5_101ErrorFifoEntry;
END_VAR
```

```
server1.protPara.linkAddr := 220;
server1.protPara.eLinkMode := eIEC870_LinkMode_Unbalanced;
server1.protPara.elinkAddrSize := eIEC870_LinkAddr_TwoOctets;

server1.sysPara.asduFmt.eAsduAddrSize := eIEC870_AsduAddr_TwoOctets;
server1.sysPara.asduFmt.eObjAddrSize := eIEC870_ObjAddr_ThreeOctets;
server1.sysPara.asduFmt.eCOTSize := eIEC870_COT_TwoOctets;
server1.sysPara.asduAddr := 7;
server1.sysPara.bUsePCTime := TRUE;
server1.sysPara.bSyncTime := TRUE;
server1.sysPara.bSyncPCTime := FALSE;
server1.sysPara.bEndOfInit := TRUE;
server1.sysPara.bPerCyclic := FALSE;
server1.sysPara.tPerCyclicBase := T#5s;
server1.sysPara.bBackScan := FALSE;
server1.sysPara.tBackScanCycle := T#30s;
server1.sysPara.bPerFRZ := TRUE;
server1.sysPara.tPerFRZCycle := T#15s;
server1.sysPara.dbgMode := IEC870_DEBUGMODE_LINKLAYER;
```

```
(* OR IEC870_DEBUGMODE_DEVSTATE OR IEC870_DEBUGMODE_ASDU;*)
server1.sysPara.bTimeCOT3 := FALSE;

server1( pInputs := ADR( inputs ),
    cbInputs := SIZEOF( inputs ),
    pOutputs := ADR( outputs ),
    cbOutputs := SIZEOF( outputs ),
    pMemory := ADR( memory ),
    cbMemory := SIZEOF( memory ),
    pData := ADR( data ),
    cbData := SIZEOF( data ),
    pAOEntries := ADR( slave1AODB ),
    cbAOEntries := SIZEOF( slave1AODB ),
    hSerial := P_SerialComm_HighSpeed.hSerial,
    bEnable := bEnable,
    eState=>eState );
```

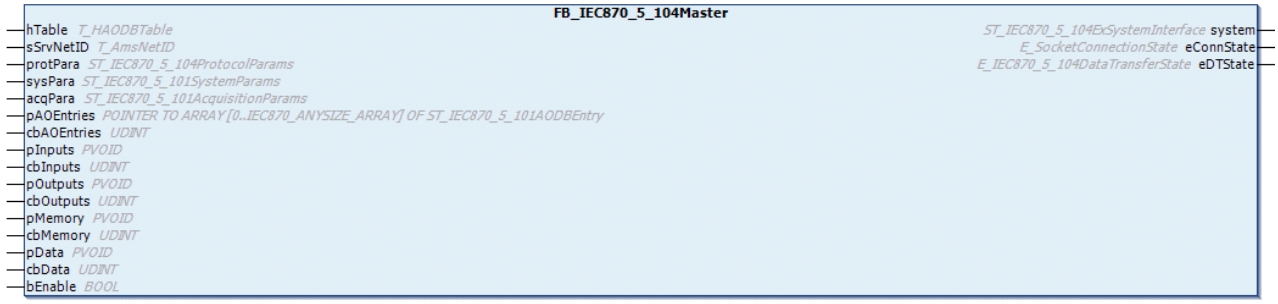
In the following example, the device error FIFO is read cyclically, and the registered errors are written to the TwinCAT XAE->"Error Log" window.

```
REPEAT
    server1.system.device.errors.RemoveError( getError=>iecError, bOk=>bError );
    IF bError THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'IEC 60870-5-101 slave error: 0x%s',
                DWORD_TO_HEXSTR( iecError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT bError
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.1.3 FB_IEC870_5_104Master



An instance of the function block FB_IEC870_5_104Master can be used to implement an IEC 60870-5-104 central station (master, high-level) in the TwinCAT PLC. A connection to the slave is established for each instance of the function block.

The function block features the following tasks;

- **STARTDT** (starts the data exchange);
- **STOPDT** (stops the data exchange);

Normally, the data exchange starts automatically once the connection has been established. This is the default configuration of the function block. If required the data exchange can be stopped or started by calling the actions.

VAR_IN_OUT

```
VAR_IN_OUT
    hTable : T_HAODBTable;
END_VAR
```

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialised once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

VAR_INPUT

```
VAR_INPUT
    sSrvNetID   : T_AmsNetID := '';
    protPara    : ST_IEC870_5_104ProtocolParams := ( bControlDT := TRUE, bDTControlled := FALSE,
                  sRemoteHost := '', nRemotePort := 2404 );
    sysPara     : ST_IEC870_5_101SystemParams := ( bEndOfInit := FALSE, asduAddr := 11, tSyncTimeut
:= T#0s );
    acqPara     : ST_IEC870_5_101AcquisitionParams;
    pAOEntries  : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries : UDINT := 0;
    pInputs     : PVOID := 0;
    cbInputs    : UDINT := 0;
    pOutputs    : PVOID := 0;
    cbOutputs   : UDINT := 0;
    pMemory     : PVOID := 0;
    cbMemory    : UDINT := 0;
    pData       : PVOID := 0;
    cbData      : UDINT := 0;
    bEnable     : BOOL := TRUE;
END_VAR
```

sSrvNetID: String containing the network address of the TwinCAT TCP/IP Connection Server. For the local computer (default) an empty string may be specified.

protPara: IEC 60870-5-104 protocol parameters [▶ 651].

sysPara: System parameter [▶ 588].

acqPara: Parameter for cyclic data acquisition [▶ 593].

pAOEntries: Address of the first application object database variable [▶ 584].

cbAOEntries: Byte size of the application object database variables.

pInputs: Address of the PLC process data area for the inputs.

cbInputs: Byte size of the PLC process data area for the inputs.

pOutputs: Address of the PLC process data area for the outputs.

cbOutputs: Byte size of the PLC process data area for the outputs.

pMamory: Address of the PLC process data area for the flags.

cbMamory: Byte size of the PLC process data area for the flags.

pData: Address of the PLC data area.

cbData: Byte size of the PLC data area.

bEnable: Activates/deactivates the function block (communication and connections).

The addresses can be determined with the ADR operator and the byte sizes with the SIZEOF operator.

VAR_OUTPUT

```
VAR_OUTPUT
    system      : ST_IEC870_5_104ExSystemInterface;
    eConnState  : E_SocketConnectionState := eSOCKET_DISCONNECTED;
    eDTState    : E_IEC870_5_104DataTransferState := eIEC870_STOPDT;
END_VAR
```

system: System interface [▶ 653]. This variable is used by other IEC application functions as a communication interface for the IEC device (here: central station).

- Member variable *system.device* is expected by the F_iecSetAOQuality [▶ 539] function as VAR_IN_OUT parameter, for example.
- Member variable *system.device.errors* is a device error FIFO. The PLC application can read and analyse registered errors.

eConnState: Status of the TCP/IP connection with the slave..

eDTState: Status of the IEC60870-5-104 data exchange [▶ 632] (STARTDT, STOPDT)

Examples (High-Level interface):

- IEC 60870-5-104 minimal central station (master) [▶ 732];
- IEC 60870-5-104 complex central station (master) [▶ 747];

Further examples:

In the following example the device error FIFO is read cyclically, and the registered errors are written into the TwinCAT XAE->"Error List" window.

```
...

fbClient : FB_IEC870_5_104Master;(* IEC 60870-5-104 master function
block instance *)

...
```

```
...

REPEAT

    fbClient.system.device.errors.RemoveError( );

    IF fbClient.system.device.errors.bOk THEN

    ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG,
'IEC60870-5-104 master error: 0x%s',

            DWORD_TO_HEXSTR(
fbClient.system.device.errors.getError.nErrId, 8, FALSE) );

    END_IF

UNTIL NOT fbClient.system.device.errors.bOk

END_REPEAT

...
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.1.4 FB_IEC870_5_104Slave



An instance of the function block FB_IEC870_5_104Slave can be used to implement an IEC 60870-5-104 substation (slave, high-level) in the TwinCAT PLC. An instance of the function block can only establish a connection with a master.

> To implement redundant systems with two or more connections to the master, please use the FB_IEC870_5_104SlaveGrp [▶ 498] function block.

VAR_IN_OUT

```
VAR_IN_OUT
    hServer : T_HSERVER;
END_VAR
```

hServer: TCP/IP server handle. The internal parameters of the server handle variable first have to be initialised via the F_CreateServerHnd function.

VAR_INPUT

```
VAR_INPUT
    protPara      : ST_IEC870_5_104ProtocolParams;
    sysPara       : ST_IEC870_5_101SystemParams;
    pAOEntries    : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries   : UDINT := 0;
    pInputs       : PVOID := 0;
    cbInputs      : UDINT := 0;
    pOutputs      : PVOID := 0;
    cbOutputs     : UDINT := 0;
    pMemory       : PVOID := 0;
    cbMemory      : UDINT := 0;
    pData         : PVOID := 0;
    cbData        : UDINT := 0;
    bEnable       : BOOL := TRUE;
END_VAR
```

protPara: IEC 60870-5-104 protocol parameters [▶ 651].

sysPara: System parameter [▶ 588].

pAOEntries: Address of the first application object database variable [▶ 584].

cbAOEntries: Byte size of the application object database variables.

pInputs: Address of the PLC process data area for the inputs.

cbInputs: Byte size of the PLC process data area for the inputs.

pOutputs: Address of the PLC process data area for the outputs.

cbOutputs: Byte size of the PLC process data area for the outputs.

pMamory: Address of the PLC process data area for the flags.

cbMamory: Byte size of the PLC process data area for the flags.

pData: Address of the PLC data area.

cbData: Byte size of the PLC data area.

bEnable: Activates/deactivates the function block (communication and connection).

The addresses can be determined with the ADR operator and the byte sizes with the SIZEOF operator.

VAR_OUTPUT

```
VAR_OUTPUT
    system      : ST_IEC870_5_104SystemInterface;
    eConnState  : E_SocketConnectionState      := eSOCKET_DISCONNECTED;
    eDTState    : E_IEC870_5_104DataTransferState  := eIEC870_STOPDT;
END_VAR
```

system: System interface [▶ 654]. This variable is used by other IEC application functions as a communication interface for the IEC device (here: substation).

- Member variable *system.device* is expected by the F_iecSetAOQuality [▶ 539] function as VAR_IN_OUT parameter, for example.
- Member variable *system.device.errors* is a device error FIFO. The PLC application can read and analyses registered errors.

eConnState: status of the TCP/IP connection with the master.

eDTState: Status of the IEC 60870-5-104 data exchange [▶ 632] (STARTDT, STOPDT)

Examples (high-level interface):

- IEC 60870-5-104 minimum substation (slave) [▶ 756];
- IEC 60870-5-104 complex substation (slave) [▶ 774];

Further examples:

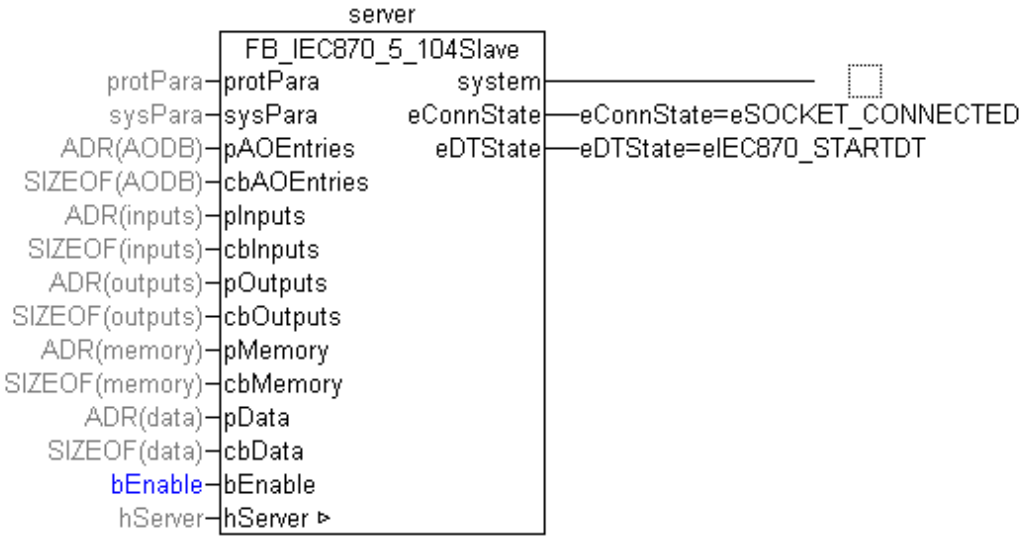
Call in FBD:

```
PROGRAM test
VAR
    AODB            : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    inputs AT%IB0   : ARRAY[0..999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..999] OF BYTE;
    memory AT%MB0   : ARRAY[0..999] OF BYTE;
    data            : ARRAY[0..999] OF BYTE;

    hServer         : T_HSERVER;
    server          : FB_IEC870_5_104Slave;


    bEnable         : BOOL := TRUE;
    protPara        : ST_IEC870_5_104ProtocolParams;
    sysPara         : ST_IEC870_5_101SystemParams := ( asduAddr := 7 );
    eConnState      : E_SocketConnectionState;
    eDTState        : E_IEC870_5_104DataTransferState;
    bError          : BOOL;
    iecError        : ST_IEC870_5_101ErrorFifoEntry;
END_VAR
```





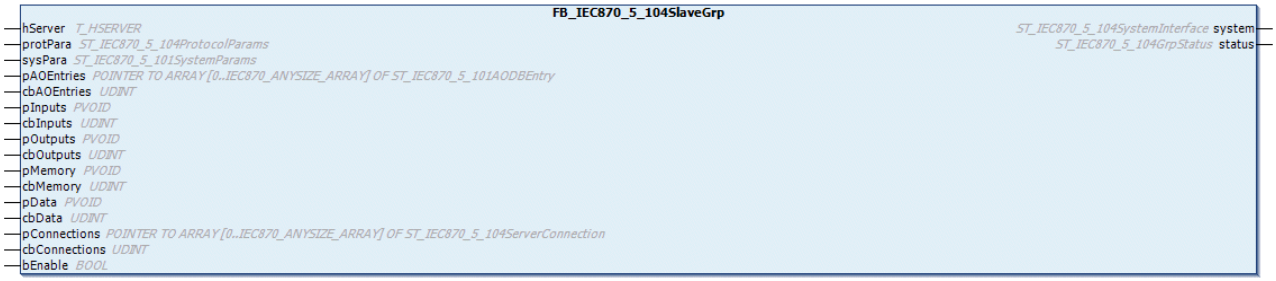
In the following example, the device error FIFO is read cyclically, and the registered errors are written to the TwinCAT XAE->"Error List" window.

```
REPEAT
    server.system.device.errors.RemoveError( getError=>iecError, bOk=>bError );
    IF bError THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'IEC 60870-5-104 slave error: 0x%s',
                DWORD_TO_HEXSTR( iecError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT bError
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.1.5 FB_IEC870_5_104SlaveGrp



An instance of the function block FB_IEC870_5_104SlaveGrp can be used to implement an IEC 60870-5-104 substation (slave, high-level) in the TwinCAT PLC. An instance can be used to implement simple systems with one connection or systems with several connections to the master (redundant system).

The maximum number of connections can be specified through the number of ST_IEC870_5_104ServerConnection [▶ 653] array elements. The address of the array variables and the byte size of the array variables must be transferred to the instance of the FB_IEC870_5_104SlaveGrp function block.

VAR_IN_OUT

```
VAR_IN_OUT
    hServer         : T_HSERVER;
END_VAR
```

hServer: TCP/IP Server-Handle. The internal parameter of the Server-Handle variable has to be initialised with the function F_CreateServerHnd.

VAR_INPUT

```
VAR_INPUT
    protPara        : ST_IEC870_5_104ProtocolParams;
    sysPara         : ST_IEC870_5_101SystemParams;
    pAOEntries      : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries     : UDINT := 0;
    pInputs         : PVOID := 0;
    cbInputs        : UDINT := 0;
    pOutputs        : PVOID := 0;
    cbOutputs       : UDINT := 0;
    pMemory         : PVOID := 0;
    cbMemory        : UDINT := 0;
    pData           : PVOID := 0;
    cbData          : UDINT := 0;
    pConnections    : POINTER TO ARRAY[0..IEC870_ANYSIZE_ARRAY] OF ST_IEC870_5_104ServerConnection :
= 0;
    cbConnections   : UDINT := 0;
    bEnable         : BOOL := TRUE;
END_VAR
```

protPara: IEC 60870-5-104 protocol parameters. [▶ 651]

sysPara: System parameter [▶ 588].

pAOEntries: Address of the first application object database variable [▶ 584].

cbAOEntries: Byte size of the application object database variables.

pInputs: Address of the PLC process data area for the inputs.

cbInputs: Byte size of the PLC process data area for the inputs.

pOutputs: Address of the PLC process data area for the outputs.

cbOutputs: Byte size of the PLC process data area for the outputs.

pMamory: Address of the PLC process data area for the flags.

cbMamory: Byte size of the PLC process data area for the flags.

pData: Address of the PLC data area.

cbData: Byte size of the PLC data area.

pConnections: Address of the ST_IEC870_5_104ServerConnection [▶ 653] array variables.

cbConnections: Byte size of the ST_IEC870_5_104ServerConnection [▶ 653] array variables.

bEnable: Activates/deactivates the function block (communication and connections).

The addresses can be determined with the ADR operator and the byte sizes with the SIZEOF operator.

VAR_OUTPUT

```
VAR_OUTPUT
    system  : ST_IEC870_5_104SystemInterface;
    status  : ST_IEC870_5_104GrpStatus;
END_VAR
```

system: System interface [▶ 654]. This variable is used by other IEC application functions as a communication interface for the IEC device (here: substation).

- Member variable *system.device* is expected by the F_iecSetAOQuality [▶ 539] function as VAR_IN_OUT parameter, for example.
- Member variable *system.device.errors* is a device error FIFO. The PLC application can read and analyses registered errors.

status: Connection and data transfer status information [▶ 654].

Example (high-level interface):

- IEC 60870-5-104 substation (slave); [▶ 775]

Further examples:

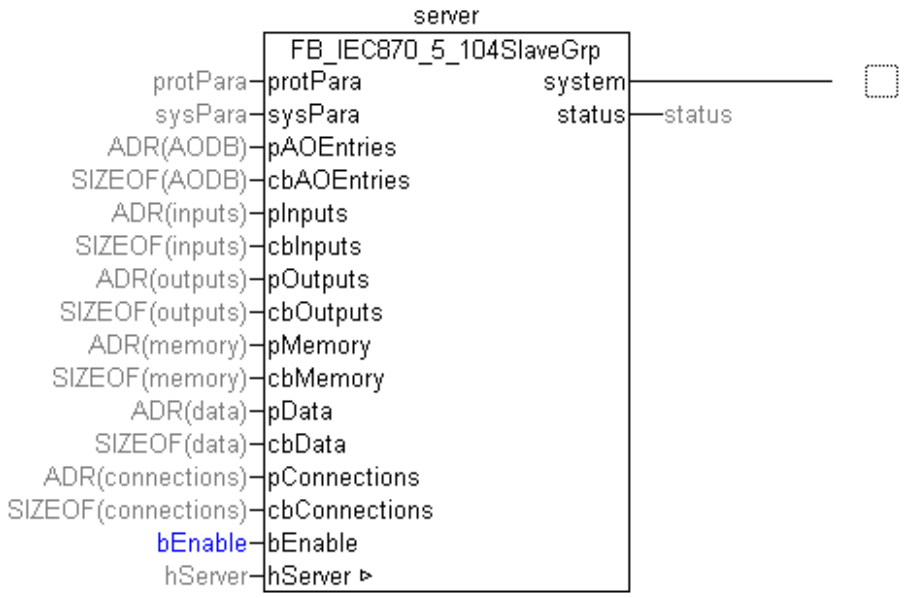
Call in FBD with up to 2 master connections:

```
PROGRAM test
VAR
    AODB            : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    inputs AT%IB0   : ARRAY[0..999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..999] OF BYTE;
    memory AT%MB0   : ARRAY[0..999] OF BYTE;
    data            : ARRAY[0..999] OF BYTE;

    hServer         : T_HSERVER;
    server          : FB_IEC870_5_104SlaveGrp;
    connections     : ARRAY[0..1] OF ST_IEC870_5_104ServerConnection; (* Two master connections *)


    bEnable         : BOOL := TRUE;
    protPara        : ST_IEC870_5_104ProtocolParams;
    sysPara         : ST_IEC870_5_101SystemParams := ( asduAddr := 7 );
    status          : ST_IEC870_5_104GrpStatus;
    bError          : BOOL;
    iecError        : ST_IEC870_5_101ErrorFifoEntry;
END_VAR
```





In the following ST example, the device error FIFO is read cyclically, and the registered errors are written to the TwinCAT XAE->"Error List" window.

```
REPEAT
    server.system.device.errors.RemoveError( getError=>iecError, bOk=>bError );
    IF bError THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'IEC 60870-5-104 slave error: 0x%s',
                DWORD_TO_HEXSTR( iecError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT bError
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.1.6 FB_IEC870_5_101TableEventHandler

Reading of the events is optional, i.e. the PLC application does not necessarily have to instance this block. This functionality is currently only supported by the high-level IEC 60870-5-101/104 master.



This function block enables the PLC application to detect certain changes in the IEC application object database and respond accordingly, if required. The changes are referred to as events. Each event type is managed in a separate internal list. The PLC application can read the pending events from one of the lists by calling the block actions. The events are counted internally since several events may occur during a PLC cycle. The counter is incremented whenever an event occurs. At the block output only the last value and the counter reading is issued.

The following events are registered by the function block:

- **OnCreate events** are reported whenever a new application object (single point, double point, measured value...) was added to the application database.
- **OnChange events** are reported when an application object is received by the lower transport layer (Rx frames) or sent (Tx frames), irrespective of whether the value of the information object has actually changed or not. For a direct command, e.g. C_SC_NA_1, in control direction (master->slave), events are usually reported for the following causes of transmission: eIEC870_COT_ACT (activation), eIEC870_COT_ACT_CON (confirmation of activation) and eIEC870_COT_TERM (completion of activation). For a data point in monitoring direction (slave->master), e.g. M_SP_NA_1, events may be reported for the following causes of transmission: eIEC870_COT_SPONTAN, eIEC870_COT_INROGEN, eIEC870_COT_BACKGROUND etc.

The function block features two tasks:

- **RemoveOnCreateEvent** (reads an entry from the OnCreate event list);
- **RemoveOnChangeEvent** (reads an entry from the OnChange event list);

VAR_IN_OUT

```
VAR_IN_OUT
    hTable : T_HAODBTable;
END_VAR
```

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialised once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

VAR_OUTPUT

```
VAR_OUTPUT
    bOk     : BOOL := FALSE;
    getObj  : ST_IEC870_5_101AOGen;
    getCfg  : ST_IEC870_5_101AOCfg;
    nEvents : DWORD := 0;
END_VAR
```

bOk: This variable becomes TRUE if a new event was read successfully. If FALSE, the event list read last is empty.

getObj: The current value of the data unit [▶ 585] (ASDU).

getCfg: The current configuration parameters of the data unit [▶ 585] (ASDU).

nEvents: Event counter (multiplier). Value range: (0 to 16#FFFFFFFF). Incrementation stops when the maximum value is reached.

Sample in ST:

In the following program section the pending events are read via REPEAT loops and written to the Windows Application log. The associated data points have already been configured as hash table entries. See function description: F_iecAddTableEntry [▶ 555].

```
PROGRAM P_LogEvents
VAR_IN_OUT
    hTable    : T_HAODBTable;
```

```
END_VAR
VAR
    fbHandler : FB_IEC870_5_101TableEventHandler;
END_VAR
```

```
REPEAT
    fbHandler.RemoveOnChangeEvent( hTable := hTable );
    IF fbHandler.bOk THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG,
            'RemoveOnChangeEvent(), IOA: %s',
            DWORD_TO_STRING( fbHandler.getObj.info.objAddr ) );
    END_IF
UNTIL NOT fbHandler.bOk
END_REPEAT

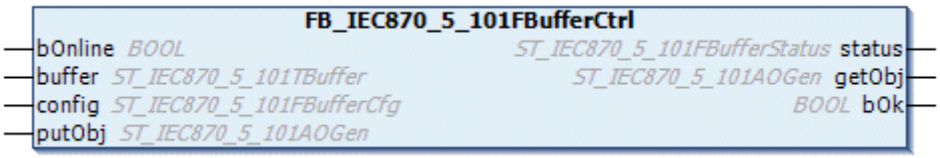

REPEAT
    fbHandler.RemoveOnCreateEvent( hTable := hTable );
    IF fbHandler.bOk THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG,
            'RemoveOnCreateEvent(), IOA: %s',
            DWORD_TO_STRING( fbHandler.getObj.info.objAddr ) );
    END_IF
UNTIL NOT fbHandler.bOk
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2 Low Level

5.1.2.1 FB_IEC870_5_101FBufferCtrl



This function block allows the contents of the TX/RX data buffer that is used for communication through the IEC 60870-5-101/104 (low level) transport interface to be manipulated. In addition, the ASDUs that are to be transmitted (only in the TX direction) are buffered in the file if the connection to the central station is interrupted (in offline mode). The functionality is similar to that of the FB_IEC870_5_101TBufferCtrl function block.

The function block features the following tasks:

- **RxRemoveObj** (removes the oldest FIFO entry from the RX FIFO);
- **RxReset** (clears all the RX FIFO entries, resets the RX FIFO);
- **TxAddObj** (inserts a new FIFO entry into the TX FIFO);
- **TxReset** (clears all the TX FIFO entries, resets the TX FIFO)

The content of the TX/RX data buffer can be changed by calling the actions listed above.

VAR_IN_OUT

```
VAR_IN_OUT
    bOnline : BOOL;
    buffer  : ST_IEC870_5_101TBuffer;
END_VAR
```

bOnline: This input tells the function block whether the connection is in offline or online mode. TRUE = online, FALSE = offline. In offline mode, the ASDUs that are to be transmitted are buffered in a file. In online mode, the ASDUs that are buffered in the file are taken out of the file and sent to the central station.

buffer: TX/RX data buffer [▶ 583]. The TX/RX buffer parameters (such as asduSize) must be configured before use.

VAR_INPUT

```
VAR_INPUT
    config  : ST_IEC870_5_101FBufferCfg;
    putObj  : ST_IEC870_5_101AOGen;
END_VAR
```

config: Offline file buffer configuration settings [▶ 638].

putObj: Data unit [▶ 585] (ASDU) that is to be transmitted.

VAR_OUTPUT

```
VAR_OUTPUT
    status  : ST_IEC870_5_101FBufferStatus;
    getObj  : ST_IEC870_5_101AOGen;
    bOk     : BOOL;
END_VAR
```

status: Offline data buffer status information [▶ 639].

getObj: Received data unit [▶ 585] (ASDU).

bOk: This variable becomes TRUE if a new entry was successfully added to or removed from the FIFO. This variable becomes FALSE if the buffer overflows and if it is not possible to remove an entry because the FIFO is already empty.

Sample in ST:

The following fragment of example code demonstrates the use of the function block actions. Approximately every 1s (tCycle) a new ASDU (M_BO_TB_1) is generated with "spontaneous" as the transmission cause and with a timestamp, and is placed in the TX FIFO.
The test command (C_TS_TA_1) and clock synchronization command (C_CS_NA_1) received are removed from the RX FIFO, and replied to with reflected ASDUs.

The VAR_IN_OUT variable *bOnline* is used to control the storage or loading of the ASDUs in the file. If *bOnline* = FALSE the file is opened, and all the TX ASDUs that arise are written into the file in the background. If bOnline = TRUE the ASDUs are taken out of the file in the background and transmitted. In order for saving and loading of the buffer file in the background to be possible, the FB_IEC870_5_101FBufferCtrl function block must be called cyclically. The insertion of new TX ASDUs or the editing of old RX ASDUs is not affected by saving or loading data into or out of the file.

```
PROGRAM P_ProcessSlaveBufferData
VAR_IN_OUT
    bOnline  : BOOL;
    buffer   : ST_IEC870_5_101TBuffer;
END_VAR
VAR
    asduAddr : DWORD := 7; (* Common asdu address *)

    fbBuffer : FB_IEC870_5_101FBufferCtrl :=( config := ( sPathName := 'c:\tmp\OfflineAsdu.dat',
                        bOverwrite := TRUE,
                    cbBuffer := 16#100000 )); (* RX/TX buffer control function block *)

    txAsdu   : ST_IEC870_5_101AOGen; (* asdu to send *)
    txTT     : T_CP56Time2a; (* time tag to send *)

    rxAsdu   : ST_IEC870_5_101AOGen; (* received asdu *)
    rxTT     : T_CP56Time2a; (* received time tag *)
```

```
    rxQOI    : BYTE; (* qualifier of interrogation command *)
    txBSI    : DWORD := 1; (* bit string value *)
    txQDS    : BYTE; (* bit string quality descriptor *)
    tCycle   : TIME := T#1s;
    bSpont   : BOOL := TRUE;
    timer    : TON;
    fbRTC    : RTC_EX2 := ( EN := TRUE, PDT := ( wYear := 2006, wMonth := 8, wDay := 17, wHour := 12
, wMinute := 23 ) );
END_VAR
```

```
timer( IN := bSpont, PT := tCycle );
IF timer.Q THEN
    timer( IN := FALSE ); timer( IN := bSpont );

    txBSI := ROL( txBSI, 1); (* Modify bit string value *)
    txQDS.7 := NOT txQDS.7; (* Toggle IV quality flag *)(* create dummy time tag *)
    fbRTC();
    txTT := SYSTEMTIME_TO_CP56Time2a( fbRTC.CDT, TRUE );

    (* create asdu *)
    txAsdu.ident.eType := M_BO_TB_1; (* Bit string with time tag *)
    txAsdu.ident.bSQ := FALSE;
    txAsdu.ident.nObj := 1;
    txAsdu.ident.eCOT := eIEC870_COT_SPONTAN;
    txAsdu.ident.nORG := 1;
    txAsdu.ident.bPN := FALSE;
    txAsdu.ident.bT := FALSE;
    txAsdu.ident.eClass := eIEC870_Class_1;
    txAsdu.ident.asduAddr := asduAddr;
    txAsdu.info.objAddr := 100;
    F_iecResetStream( 0, txAsdu.info.stream ); (* clear previous data (this sets the stream length =
 0 *)
    F_iecCopyBufferToStream( ADR( txBSI ), SIZEOF( txBSI ), txAsdu.info.stream ); (* put BSI to stre
am *)
    F_iecCopyBufferToStream( ADR( txQDS ), SIZEOF( txQDS ), txAsdu.info.stream ); (* put QDS to stre
am *)
    F_iecCopyBufferToStream( ADR( txTT ), SIZEOF( txTT ), txAsdu.info.stream ); (* put time tag to s
tream *)

    fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to the
TX fifo *)
    IF NOT fbBuffer.bOk THEN
        RETURN;
        (* TODO: Report send buffer overflow error *)
    END_IF
END_IF




REPEAT
    fbBuffer.RxRemoveObj( bOnline := bOnline, getObj=>rxAsdu, buffer := buffer ); (* Try to remove a
sdu from RX fifo *)
    IF fbBuffer.RxRemoveObj.bOk THEN (* success *)
        CASE rxAsdu.ident.eType OF

        C_TS_NA_1: (* Simple test command implementation *)

        txAsdu := rxAsdu;
        txAsdu.ident.eCOT := eIEC870_COT_ACT_CON; (* send activation confirmation *)
        fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to
the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            EXIT;
            (* TODO: Report send buffer overflow error *)
        END_IF

        C_CS_NA_1: (* Simple clock synchronisation command implementation *)

        F_iecCopyStreamToBuffer( ADR( rxTT ), SIZEOF( rxTT ), rxAsdu.info.stream );

        (*... *)

        txAsdu := rxAsdu; (* dummy old time value *)
        txAsdu.ident.eCOT := eIEC870_COT_ACT_CON; (* send activation confirmation *)
        fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to
the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            EXIT;
```

```
            (* TODO: Report send buffer overflow error *)
        END_IF

        C_IC_NA_1: (* Simple interrogation command implementation *)

        txAsdu := rxAsdu;
        txAsdu.ident.eCOT := eIEC870_COT_ACT_CON; (* send activation confirmation *)
        fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to
the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            EXIT;
            (* TODO: Report send buffer overflow error *)
        END_IF

        F_iecCopyStreamToBuffer( ADR( rxQOI ), SIZEOF(rxQOI), rxAsdu.info.stream );

        (* create asdu *)
        txAsdu.ident.eType := M_BO_NA_1; (* Bit string without time tag! *)
        txAsdu.ident.bSQ := FALSE;
        txAsdu.ident.nObj := 1;
        txAsdu.ident.eCOT := BYTE_TO_INT( rxQOI );
        txAsdu.ident.nORG := 1;
        txAsdu.ident.bPN := FALSE;
        txAsdu.ident.bT := FALSE;
        txAsdu.ident.eClass := eIEC870_Class_1;
        txAsdu.ident.asduAddr := asduAddr;
        txAsdu.info.objAddr := 100;
        F_iecResetStream( 0, txAsdu.info.stream ); (* clear previous data (this sets the stream leng
th = 0 *)
        F_iecCopyBufferToStream( ADR( txBSI ), SIZEOF( txBSI ), txAsdu.info.stream ); (* put BSI to
stream *)
        F_iecCopyBufferToStream( ADR( txQDS ), SIZEOF( txQDS ), txAsdu.info.stream ); (* put QDS to
stream *)
        fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to
the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            EXIT;
            (* TODO: Report send buffer overflow error *)
        END_IF


        txAsdu := rxAsdu;
        txAsdu.ident.eCOT := eIEC870_COT_ACT_TERM; (* send activation termination *)
        fbBuffer.TxAddObj( bOnline := bOnline, putObj := txAsdu, buffer := buffer ); (* put asdu to
the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            EXIT;
            (* TODO: Report send buffer overflow error *)
        END_IF

        ELSE
            (* TODO: Report invalid asdu type...*)
            EXIT;
        END_CASE

    END_IF
UNTIL NOT fbBuffer.bOk (* RX fifo is empty *)
END_REPEAT

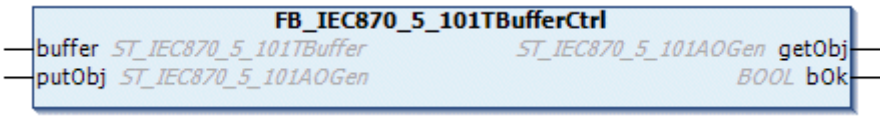


(* Offline frames are written to the file. Execute this function block in every cycle! *)
fbBuffer(bOnline := bOnline, buffer:= buffer );
IF fbBuffer.status.eState = eIEC870_FBUFFER_ERROR THEN
    (*TODO: Report file access error *)
    ;
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.2 FB_IEC870_5_101TBufferCtrl



This function block can be used to manipulate the content of the TX/RX data buffer, which is used for the communication via the IEC 60870-5-101/104 low-level interface.

The function block features the following tasks:

- **RxRemoveObj** (removes the oldest FIFO entry from the RX FIFO);
- **RxReset** (clears all the RX FIFO entries, resets the RX FIFO);
- **TxAddObj** (inserts a new FIFO entry into the TX FIFO);
- **TxReset** (clears all the TX FIFO entries, resets the TX FIFO)

The content of the TX/RX data buffer can be changed by calling the actions listed above.

VAR_IN_OUT

```
VAR_IN_OUT
    buffer  : ST_IEC870_5_101TBuffer;
END_VAR
```

buffer: TX/RX data buffer [▶ 583]. The TX/RX buffer parameters (like e.g. asduSize) have to be configured before using.

VAR_INPUT

```
VAR_INPUT
    putObj  : ST_IEC870_5_101AOGen;
END_VAR
```

putObj: Data unit [▶ 585] (ASDU) to be sent.

VAR_OUTPUT

```
VAR_OUTPUT
    getObj  : ST_IEC870_5_101AOGen;
    bOk     : BOOL;
END_VAR
```

getObj: Received data unit [▶ 585] (ASDU).

bOk: This variable becomes TRUE if a new entry was successfully added to or removed from the FIFO. This variable becomes FALSE if the buffer overflows and if it is not possible to remove an entry because the FIFO is already empty.

Examples (low-level interface):

- IEC 60870-5-101 central station (master, unbalanced mode); [▶ 687]
- IEC 60870-5-101 substation (slave, unbalanced mode) [▶ 718];

- IEC 60870-5-101 central station (master, balanced mode) [▶ 689];
- IEC 60870-5-101 substation (slave, balanced mode) [▶ 720];

Further examples:

The following fragment of example code demonstrates the use of the function block actions. Approximately every 100 ms (tCycle) a new ASDU (M_BO_TB_1) is generated with "spontaneous" as the transmission cause and with a timestamp, and is placed in the TX FIFO.
The test command (C_TS_TA_1) and clock synchronization command (C_CS_NA_1) received are removed from the RX FIFO, and replied to with reflected ASDUs.

```
PROGRAM P_SAMPLE_1ms
VAR
    (* TX/RX data buffer *)
    fbBuffer : FB_IEC870_5_101TBufferCtrl;
    buffer   : ST_IEC870_5_101TBuffer := ( asduSize := 253 );

    tCycle   : TIME := T#100ms;
    timer    : TON;
    dtStart  : DT := DT#2006-07-05-12:34:56;

    txAsdu   : ST_IEC870_5_101AOGen;
    rxAsdu   : ST_IEC870_5_101AOGen;

    txBSI    : DWORD := 1;
    txQDS    : BYTE;
    txTT     : T_CP56Time2a;
    rxTT     : T_CP56Time2a;
END_VAR
```

```
timer( IN := TRUE, PT := tCycle );
IF timer.Q THEN
    timer( IN := FALSE );

    txAsdu.ident.eType := M_BO_TB_1; (* Bit string with time tag *)
    txAsdu.ident.bSQ := FALSE;
    txAsdu.ident.nObj := 1;
    txAsdu.ident.eCOT := eIEC870_COT_SPONTAN;
    txAsdu.ident.nORG := 1;
    txAsdu.ident.bPN := FALSE;
    txAsdu.ident.bT := FALSE;
    txAsdu.ident.eClass := eIEC870_Class_1;
    txAsdu.ident.asduAddr := 7;
    txAsdu.info.objAddr := 100;


    txBSI := ROL( txBSI, 1); (* Modify bit string value *)
    txQDS.7 := NOT txQDS.7; (* Toggle IV quality flag *)(* create dummy time tag *)
    dtStart := dtStart + tCycle;
    txTT := SYSTEMTIME_TO_CP56Time2a( DT_TO_SYSTEMTIME( dtStart ), TRUE );

    F_iecResetStream( 0, txAsdu.info.stream ); (* clear previous data (this sets the stream length =
 0 *)
    F_iecCopyBufferToStream( ADR( txBSI ), SIZEOF( txBSI ), txAsdu.info.stream ); (* put BSI to stre
am *)
    F_iecCopyBufferToStream( ADR( txQDS ), SIZEOF( txQDS ), txAsdu.info.stream ); (* put QDS to stre
am *)
    F_iecCopyBufferToStream( ADR( txTT ), SIZEOF( txTT ), txAsdu.info.stream ); (* put time tag to s
tream *)

    fbBuffer.TxAddObj( putObj := txAsdu, buffer := buffer ); (* put asdu to the TX fifo *)
    IF NOT fbBuffer.bOk THEN
        ;(* Report send buffer overflow error *)
    END_IF
END_IF



REPEAT
fbBuffer.RxRemoveObj( getObj=>rxAsdu, buffer := buffer ); (* Try to remove asdu from RX fifo *)
IF fbBuffer.bOk THEN (* success *)

    CASE rxAsdu.ident.eType OF
    C_TS_TA_1: (* Test command *)

        txAsdu := rxAsdu;
        txAsdu.ident.eCOT := eIEC870_COT_ACT_CON; (* send activation confirmation *)
```

```
        fbBuffer.TxAddObj( putObj := txAsdu, buffer := buffer ); (* put asdu to the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            ;(* Report send buffer overflow error *)
        END_IF


    C_CS_NA_1: (* clock synchronisation *)

        F_iecCopyStreamToBuffer( ADR( rxTT ), SIZEOF( rxTT ), rxAsdu.info.stream );

        (*... *)

        txAsdu := rxAsdu; (* dummy old time value *)
        txAsdu.ident.eCOT := eIEC870_COT_ACT_CON; (* send activation confirmation *)

        fbBuffer.TxAddObj( putObj := txAsdu, buffer := buffer ); (* put asdu to the TX fifo *)
        IF NOT fbBuffer.bOk THEN
            ;(* Report send buffer overflow error *)
        END_IF

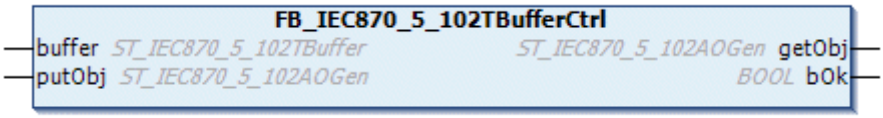
    END_CASE

END_IF
UNTIL NOT fbBuffer.bOk (* RX fifo is empty *)
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.3 FB_IEC870_5_102TBufferCtrl



This function block can be used to manipulate the content of the TX/RX data buffer, which is used for the communication via the IEC60870-5-102 low-level interface.

The function block features the following tasks:

- **RxRemoveObj** (removes the oldest FIFO entry from the RX FIFO);
- **RxReset** (clears all the RX FIFO entries, resets the RX FIFO);
- **TxAddObj** (inserts a new FIFO entry into the TX FIFO);
- **TxReset** (clears all the TX FIFO entries, resets the TX FIFO)

The content of the TX/RX data buffer can be changed by calling the actions listed above.

VAR_IN_OUT

```
VAR_IN_OUT
    buffer  : ST_IEC870_5_102TBuffer;
END_VAR
```

buffer: TX/RX data buffer [▶ 645]. The TX/RX buffer parameters (like e.g. asduSize) have to be configured before using.

VAR_INPUT

```
VAR_INPUT
    putObj   : ST_IEC870_5_102AOGen;
END_VAR
```

putObj: Data unit [▶ 646] (ASDU) that is to be transmitted.

VAR_OUTPUT

```
VAR_OUTPUT
    getObj   : ST_IEC870_5_102AOGen;
    bOk      : BOOL;
END_VAR
```

getObj: Received data unit [▶ 646] (ASDU).

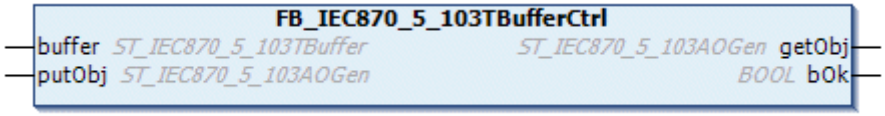
bOk: This variable becomes TRUE if a new entry was successfully added to or removed from the FIFO. This variable becomes FALSE if the buffer overflows and if it is not possible to remove an entry because the FIFO is already empty.

Example (low-level interface): IEC 60870-5-102 central station (master) [▶ 724];

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.4 FB_IEC870_5_103TBufferCtrl



This function block can be used to manipulate the content of the TX/RX data buffer, which is used for the communication via the IEC60870-5-103 low-level interface.

The function block features the following tasks:

- **RxRemoveObj** (removes the oldest FIFO entry from the RX FIFO);
- **RxReset** (clears all the RX FIFO entries, resets the RX FIFO);
- **TxAddObj** (inserts a new FIFO entry into the TX FIFO);
- **TxReset** (clears all the TX FIFO entries, resets the TX FIFO)

The content of the TX/RX data buffer can be changed by calling the actions listed above.

VAR_IN_OUT

```
VAR_IN_OUT
    buffer   : ST_IEC870_5_103TBuffer;
END_VAR
```

buffer: TX/RX data buffer [▶ 648]. The TX/RX buffer parameters (like e.g. asduSize) have to be configured before using.

VAR_INPUT

```
VAR_INPUT
    putObj   : ST_IEC870_5_103AOGen;
END_VAR
```

putObj: Data unit [▶ 649] (ASDU) to be sent.

VAR_OUTPUT

```
VAR_OUTPUT
    getObj   : ST_IEC870_5_103AOGen;
    bOk      : BOOL;
END_VAR
```

getObj: Received data unit [▶ 649] (ASDU).

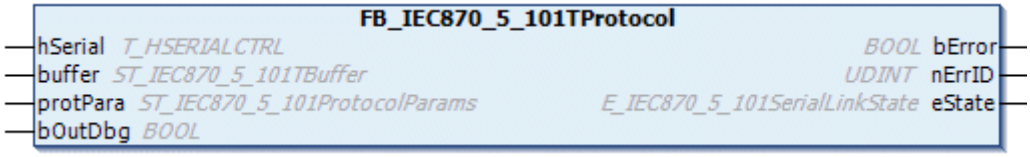
bOk: This variable becomes TRUE, if a new entry was inserted or removed successfully from the Fifo. This variable becomes FALSE at a buffer overflow and if no entry could be removed, because the Fifo was already empty.

Example (Low-Level interface): IEC 60870-5-103 central station (master) [▶ 727]

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.5 FB_IEC870_5_101TProtocol



The communication block FB_IEC870_5_101TProtocol implements the transfer procedures of the link layer according to the IEC 60870-5-1 and IEC 60870-5-2 standards.
In the event of a protocol error an associated error code is issued at the function block output and the data transfer is interrupted. The data exchange can be reactivated by calling the INIT task. The TX/RX data buffers are reset. The communication block expects a TX/RX data buffer variable. This variable must be transferred to the block via VAR_IN_OUT.

The function block features the following tasks:

- **INIT** (initializes the function block);

Protocol configuration

The communication block has a structured protPara variable. Protocol parameters such as timeout times or connection addresses etc. can be configured via this variable

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial  : T_HSERIALCTRL;
    buffer   : ST_IEC870_5_101TBuffer;
END_VAR
```

hSerial: Connection handle [▶ 632] to FB_IEC870_SerialLineCtrl [▶ 519].. The data to be sent and received are excahged with the FB_IEC870_SerialLineCtrl function block.

buffer: TX/RX data buffer [▶ 583].

VAR_INPUT

```
VAR_INPUT
    protPara    : ST_IEC870_5_101ProtocolParams;
    bOutDbg     : BOOL;
END_VAR
```

protPara: IEC 60870-5-101 protocol parameters [▶ 640]

bOutDbg: Activates/deactivates the debug output of the frames in the TwinCAT XAE->"Error List" window.

VAR_OUTPUT

```
VAR_OUTPUT
    bError  : BOOL;
    nErrID  : UDINT;
    eState  : E_IEC870_5_101SerialLinkState := eSERIALLINK_DISCONNECTED;
END_VAR
```

bError: This output is switched to TRUE if an error occurs during data transfer.

nErrID: Returns an error code [▶ 784] if the bError output is set.

eState: Connection status [▶ 630] to the other communication partner.

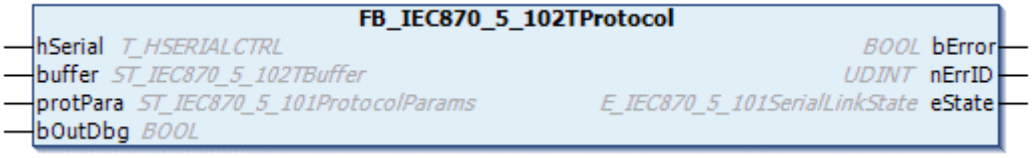
Examples (Low-Level Interface):

- IEC 60870-5-101 central station (master, unbalanced mode) [▶ 687];
- IEC 60870-5-101 substation (slave, unbalanced mode) [▶ 718];
- IEC 60870-5-101 central station (master, balanced mode) [▶ 689];
- IEC 60870-5-101 substation (slave, balanced mode) [▶ 720];

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.6 FB_IEC870_5_102TProtocol



The communication block FB_IEC870_5_102TProtocol implements the transfer procedures of the link layer according to the IEC 60870-5-1 and IEC 60870-5-2 standards.
In the event of a protocol error an associated error code is issued at the function block output and the data transfer is interrupted. The data exchange can be reactivated by calling the INIT task. The communication block expects a TX/RX data buffer variable. This variable must be transferred to the block via VAR_IN_OUT.

The function block features the following tasks:

- **INIT** (initializes the function block). In the default configuration, the TX/RX data buffers are reset. Clearing of the buffers can be prevented by setting the bRetainBuffer variables in the protocol parameter structure.

Protocol configuration

The communication block has a structured protPara variable. Protocol parameters such as timeout times or connection addresses etc. can be configured via this variable.

VAR_IN_OUT

```
VAR_IN_OUT
        hSerial : T_HSERIALCTRL;
    buffer  : ST_IEC870_5_102TBuffer;
END_VAR
```

hSerial: Connection handle [▶ 632] to FB_IEC870_SerialLineCtrl [▶ 519] function block. The data to be sent and received are exchanged with the FB_IEC870_SerialLineCtrl function block.

buffer: TX/RX data buffer [▶ 645].

VAR_INPUT

```
VAR_INPUT
    protPara    : ST_IEC870_5_101ProtocolParams;
    bOutDbg     : BOOL;
END_VAR
```

protPara: IEC 60870-5-101/102 protocol parameters [▶ 640].

bOutDbg:: Activates/deactivates the debugging output of the frames in the TwinCAT XAE->"Error List" window.

VAR_OUTPUT

```
VAR_OUTPUT
    bError  : BOOL;
    nErrID  : UDINT;
    eState  : E_IEC870_5_101SerialLinkState := eSERIALLINK_DISCONNECTED;
END_VAR
```

bError: This output is switched to TRUE if an error occurs during data transfer.

nErrID: Returns an error code [▶ 793] if the bError output is set.

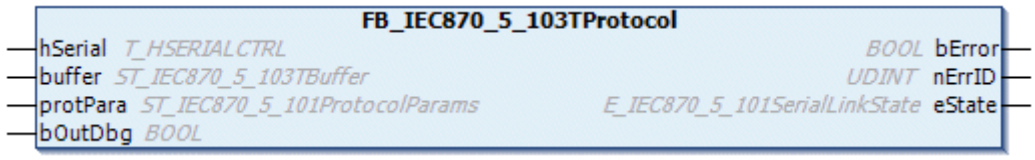
eState: Connection status [▶ 630] to the communication partner.

Example (Low-Level interface): IEC 60870-5-102 central station (master) [▶ 724];

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.7 FB_IEC870_5_103TProtocol



The communication block FB_IEC870_5_103TProtocol implements the transfer procedures of the link layer according to the IEC 60870-5-1 and IEC 60870-5-2 standards.
In the event of a protocol error an associated error code is issued at the function block output and the data transfer is interrupted. The data exchange can be reactivated by calling the INIT task. The communication block expects a TX/RX data buffer variable. This variable must be transferred to the block via VAR_IN_OUT.

The function block features the following tasks:

- **INIT** (initializes the function block). In the default configuration, the TX/RX data buffers are reset. Clearing of the buffers can be prevented by setting the bRetainBuffer variables in the protocol parameter structure.

Protocol configuration

The communication block has a structured protPara variable. Protocol parameters such as timeout times or connection addresses etc. can be configured via this variable.

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial : T_HSERIALCTRL;
    buffer  : ST_IEC870_5_103TBuffer;
END_VAR
```

hSerial: Connection handle [▶ 632] to FB_IEC870_SerialLineCtrl [▶ 519] function block. The data to be sent and received are excahged with the FB_IEC870_SerialLineCtrl function block.

buffer: TX/RX data buffer [▶ 648].

VAR_INPUT

```
VAR_INPUT
    protPara    : ST_IEC870_5_101ProtocolParams;
    bOutDbg : BOOL;
END_VAR
```

protPara: IEC 60870-5-101/103 protocol parameters [▶ 640].

bOutDbg: Activates/deactivates the debug output of the frames in the TwinCAT XAE->"Error List" window.

VAR_OUTPUT

```
VAR_OUTPUT
    bError   : BOOL;
    nErrID   : UDINT;
    eDTState : E_IEC870_5_101SerialLinkState := eSERIALLINK_DISCONNECTED;
END_VAR
```

bError: This output is switched to TRUE if an error occurs during data transfer.

nErrID: Returns an error code [▶ 799] if the bError output is set.

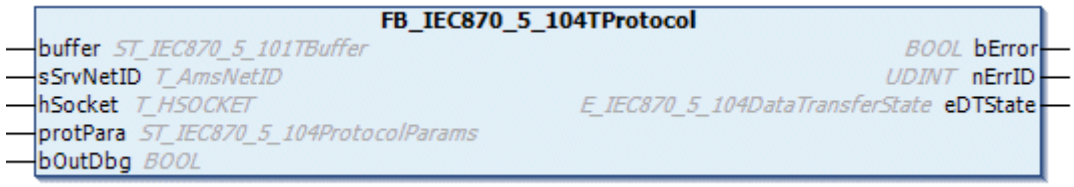
eState: Connection status [▶ 630] vis-à-vis the other communication partner.

Example (low-level interface): IEC 60870-5-103 central station (master) [▶ 727];

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.2.8 FB_IEC870_5_104TProtocol



The communication block FB_IEC870_5_104TProtocol implements the ACPI functions of the IEC 60870-5-104 standard (start/stop data transfer, test frames, send/receive frame counter etc.). In the event of a protocol error an associated error code is issued at the function block output and the data transfer is interrupted. The data exchange can be reactivated by calling the INIT task. The frame counters and the send and TX/RX data buffers are reset. The communication block expects a TX/RX data buffer variable. This variable must be transferred to the block via VAR_IN_OUT.

The function block features the following tasks:

- **INIT** (initializes the function block);
- **STARTDT** (sends a start data transfer frame to the communication partner);
- **STOPDT** (sends a stop data transfer frame to the communication partner);

Establishing the connection

The TCP/IP connection has to be established and closed via a separate block, e.g. FB_ServerClientConnection. The PLC application can respond independently to possible protocol errors and close the connection or implement the process reset service, for example. At the output this block provides a connection handle, the connection status and information about errors that have occurred during establishment/closing of the connection.
The connection handle is required by the communication block.

Protocol configuration

The communication block has a structured protPara variable. Protocol parameters such as iK, iW, start/stop data transfer behaviour etc. can be configured via this variable.

VAR_IN_OUT

```
VAR_IN_OUT
    buffer  : ST_IEC870_5_101TBuffer;
END_VAR
```

buffer: TX/RX data buffer [▶ 583].

VAR_INPUT

```
VAR_INPUT
    sSrvNetID : T_AmsNetID;
    hSocket   : T_HSOCKET;
    protPara  : ST_IEC870_5_104ProtocolParams;
    bOutDbg   : BOOL;
END_VAR
```

sSrvNetID: String containing the network address of the TwinCAT TCP/IP Connection Server. For the local computer (default) an empty string may be specified.

hSocket: TCP/IP connection handle of the communication partner for receiving or sending data.

protPara: IEC 60870-5-104 protocol parameters [▶ 651].

bOutDbg:: Activates/deactivates the debug output of the TCP/IP frames in the TwinCAT System Manager logger view.

VAR_OUTPUT

```
VAR_OUTPUT
    bError   : BOOL;
    nErrID   : UDINT;
    eDTState : E_IEC870_5_104DataTransferState := eIEC870_STOPDT;
END_VAR
```

bError: This output is switched to TRUE if an error occurs during data transfer.

nErrID: Returns an error code [▶ 805] if the bError output is set;

eDTState: Status of the IEC 60870-5-104 data exchange [▶ 632] (STARTDT, STOPDT).

Examples:

- IEC 60870-5-104 central station (master); [▶ 751]
- IEC 60870-5-104 substation (slave); [▶ 779]

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.3 FB_IEC870_PartyLineCtrl



The function block FB_IEC870_PartyLineCtrl can be used for the data exchange with the central station in line mode (party line). The function block has to be called cyclically in the PLC task. The *eMode* input can be used to activate/deactivate line mode. Line mode is not used if you do not use the function block (default setting).

The hSerial variable is a structure and is used for internal data exchange between the fast and slow communication task. Every time the IEC slave wants to transmit, the *bRTS* output (request to send) is first set to TRUE. Once transmit mode has been switched on the IEC slave is notified by setting the *bCTS* input (clear to send) to TRUE. The IEC slave then starts transmitting. Once the data have been sent (internal hardware buffers are empty) the IEC slave resets the *bRTS* output to FALSE. The send line can now be released for another device. Once this has occurred the IEC slave must be notified by setting the *bCTS* input to FALSE. In other words, the state of the *bCTS* input always follows the state of the *bRTS* output.

In the serial PC interface, the *bRTS* output is not set to FALSE (data sent) until the query of the internal hardware send buffer returns zero bytes in the buffer.

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial     : T_HSERIALCTRL;
END_VAR
```

hSerial: Connection handle [▶ 632] to FB_IEC870_SerialLineCtrl [▶ 519]. Via this variable the data to be sent and received are exchanged with FB_IEC870_SerialLineCtrl.

VAR_INPUT

```
VAR_INPUT
    eMode    : E_IEC870_5_101PartylineMode := eIEC870_PartylineMode_Off;
    deviceID : UDINT := 0;
    bCTS     : BOOL := FALSE;
END_VAR
```

eMode: Party line activation mode [▶ 630].

deviceID: This parameter is not used in TwinCAT 3.1 and does not have to be set.

bCTS: Clear to send (for the IEC link layer).

VAR_OUTPUT

```
VAR_IN_OUT
    bError  : BOOL;
    nErrID  : UDINT;
    bRTS    : BOOL := FALSE;
END_VAR
```

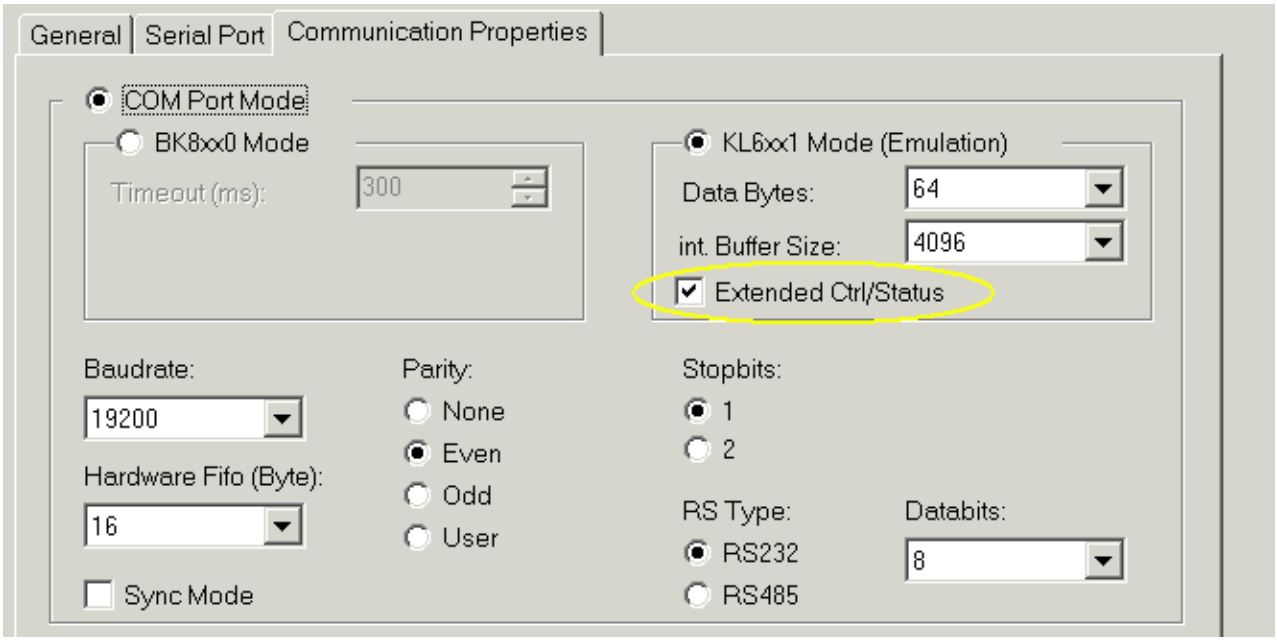
bError: Becomes TRUE as soon as an error has occurred.

nErrID: Returns the error code [▶ 784], if the *bError* output is set.

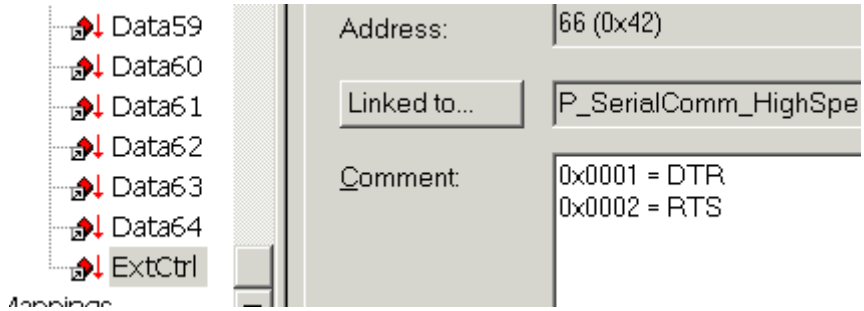
bRTS: Request to send (from the IEC link layer).

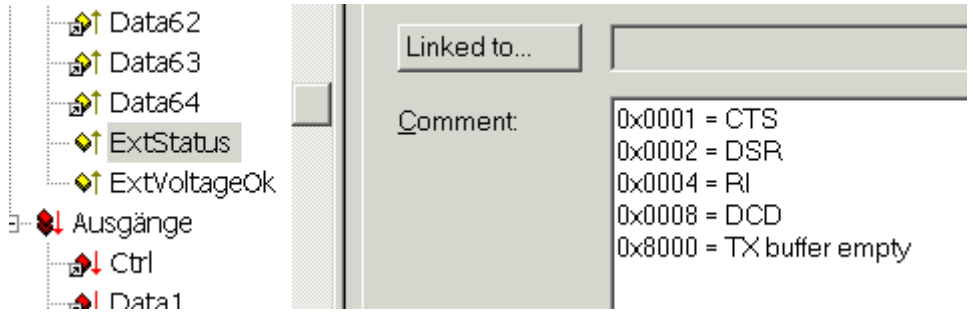
Example for line mode configuration:

The RTS line can be mapped directly as an IO output in the PLC. In the TwinCAT System Manager the option **"Extended Ctrl/Status"** must be activated. The eMode parameter at the function block must be set to *eMode := eIEC870_PartylineMode_Ext_On*.



The "**serial_ExtCtrl**" variable must be linked with "**ExtCtrl**" (RTS output) and the "**serial_ExtStatus**" variable with "**ExtStatus**".






```
PROGRAM P_SerialComm_HighSpeed
VAR
    serial_In         AT%I*   : PcComInData;
    serial_Out        AT%Q*   : PcComOutData;
    KL6_In            AT%I*   : KL6inData5B;
    KL6_Out           AT%Q*   : KL6outData5B;
    serial_ExtStatus  AT%I*   : WORD;
    serial_ExtCtrl    AT%Q*   : WORD;

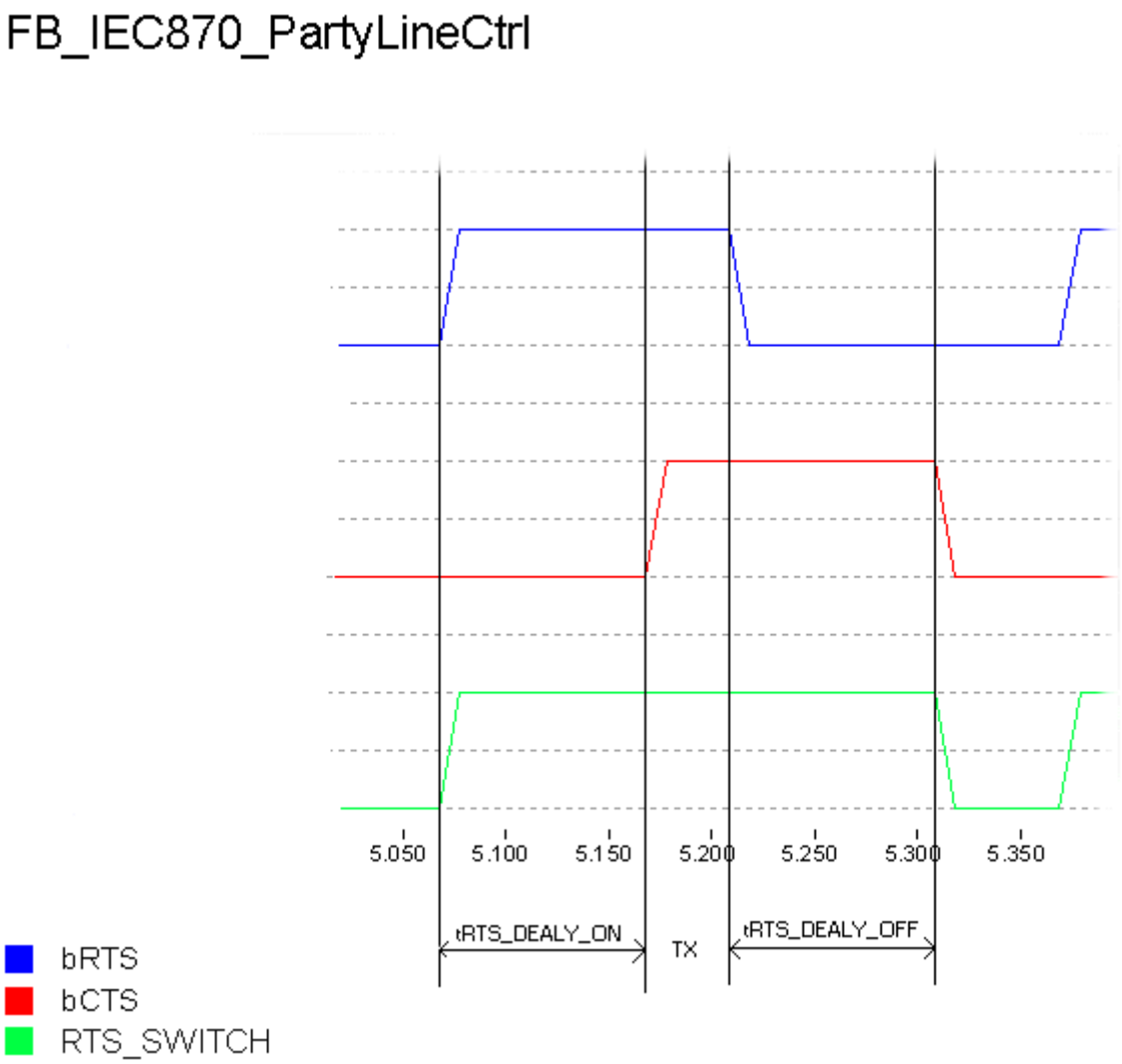
    hSerial          : T_HSERIALCTRL;(* Serial connection control handle *)
    fbSerialLineCtrl : FB_IEC870_SerialLineCtrl := ( Mode := SERIALLINEMODE_PC_COM_PORT(*SERIALLINEM
ODE_KL6_5B_STANDARD *),
                       Baudrate := 19200,
                       NoDatabits := 8,
                       Parity := PARITY_EVEN,
                       Stopbits := 1,
                       Handshake := HANDSHAKE_NONE,
                       ContinousMode := FALSE );

    fbPartyLineCtrl  : FB_IEC870_PartyLineCtrl := ( eMode := eIEC870_PartylineMode_Ext_On );
    delay            : TON;
    tRTS_DEALY_ON    : TIME := T#100ms;
    tRTS_DELAY_OFF   : TIME := T#100MS;
END_VAR
```

```
fbSerialLineCtrl(  pComIn := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR( seria
l_In ), ADR( KL6_In ) ),
        pComOut := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR( serial_Out ), A
DR( KL6_Out ) ),
        SizeComIn :=  SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD, SIZEOF( serial_In
 ), SIZEOF( KL6_In ) ),
        hSerial  := hSerial );

fbPartyLineCtrl( hSerial:= hSerial );
IF fbPartyLineCtrl.bRTS <> fbPartyLineCtrl.bCTS THEN
    IF fbPartyLineCtrl.bRTS THEN
        serial_ExtCtrl.1  := TRUE;(* Switch RTS line ON *)
        delay( in := TRUE, PT := tRTS_DEALY_ON );(* Wait until line enabled *)
     IF delay.Q THEN
            delay( in := FALSE );
            fbPartyLineCtrl.bCTS := TRUE;(* Set clear to send *)
        END_IF
    ELSE
        IF serial_ExtStatus.15  THEN
            delay( in := TRUE, PT := tRTS_DELAY_OFF );(* Wait until all data send *)
            IF delay.Q THEN
                delay( in := FALSE );
                serial_ExtCtrl.1  := FALSE;(* Switch RTS line OFF *)
                fbPartyLineCtrl.bCTS := FALSE;(* Reset clear to send *)
            END_IF
        END_IF
    END_IF
END_IF
```

Implementation of line mode in the fast communication task: The *RTS_SWITCH* variable **(serial_ExtCtrl.1)** is used to switch the line for transmit mode ON and OFF. The *tRTS_DEALY_ON* delay time (lead time) ensures that activation of the line for the IEC slave was complete, the *tRTS_DELAY_OFF* delay time (overrun time) ensures that all data bytes sent were received by the central station, including the last one **(serial_ExtStatus.15)**.



Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.4 FB_IEC870_SerialLineCtrl



The function block FB_IEC870_SerialLineCtrl deals with the communication between a serial interface (KL60xx, EL60xx or COM interface) and the IEC60870-5-10x PLC function blocks.
If a **KL60xx** serial terminal is used for the communication, the Bus Terminal is initialized and configured first by the function block (baud rate, parity, etc.). However, the **PC-COM interface** and the **EL60xx** terminal must be configured in the TwinCAT System Manager. The data received and to be sent are held in the internal buffers of the hSerial variable. The function block has to be called cyclically in the PLC task.

VAR_IN_OUT

```
VAR_IN_OUT
    hSerial     : T_HSERIALCTRL;
END_VAR
```

hSerial : Connection handle [▶ 632] to other IEC 60870-5-10x protocol function blocks. Via this variable the data to be sent and received are exchanged with FB_IEC870_SerialLineCtrl.

VAR_INPUT

```
VAR_INPUT
    Mode          : ComSerialLineMode_t   := SERIALLINEMODE_PC_COM_PORT;
    Baudrate      : UDINT       := 19200;
    NoDatabits    : BYTE        := 8;
    Parity        : ComParity_t:= PARITY_EVEN;
    Stopbits      : BYTE        := 1;
    Handshake     : ComHandshake_t    := HANDSHAKE_NONE;
    ContinousMode: BOOL        := FALSE;
    pComIn        : PVOID       := 0;
    pComOut       : PVOID       := 0;
    SizeComIn     : UINT        := 0;
END_VAR
```

Mode: The Mode input specifies unambiguously which serial hardware is being used.

Baudrate: The baud rate, as far as it is supported by the serial hardware (only in KL60xx mode: 115200, 57600, 38400, 19200, 9600, 4800, 2400, 1200).

NoDatabits: Number of user data bits in one data byte (only in KL60xx mode: 7 or 8).

Parity: Type of the parity bit of a data byte (only in KL60xx mode: PARITY_NONE=0, PARITY_EVEN=1, PARITY_ODD=2).

Stopbits: Number of stop bits per data byte (only in KL60xx mode: 1 or 2).

Handshake: Type of handshake used, as far as it is supported by the serial hardware (only in KL60xx mode: HANDSHAKE_NONE=0, HANDSHAKE_RTSCTS=1, HANDSHAKE_XONXOFF=2).

ContinousMode: Switches on continuous transmission, provided this is supported by the serial hardware.

If ContinousMode is TRUE, transmitted data is not sent out by the serial hardware until the hardware transmit buffer is full. This means that there are no time gaps in the transmission, provided the quantity of data is similar in size to the hardware transmit buffer. Continuous mode is only necessary in special cases in which the end device reacts to time gaps with a time-out.

pComIn: Universal pointer to the input variable of the process data for the serial hardware (data types KL6inData, KL6inData5b, PcComInData, EL6inData22B). The pointer is assigned with the *ADR*() function.

pComOut: Universal pointer to the output variable of the process data for the serial hardware (data types KL6outData, KL6outData5b, PcComOutData, EL6outData22B). The pointer is assigned with the *ADR*() function.

SizeComIn: Size of the input process image of the serial hardware being used. The size is determined and assigned with the *SIZEOF*() function.

Example 1:

The example shows an ST call. Switching between two communication routes can be affected by setting the mode variables.

For mode = SERIALLINEMODE_PC_COM_PORT communication is via a serial COM port of the PC, for mode = SERIALLINEMODE_KL6_5B_STANDARD via a KL6001 Bus Terminal (5-byte mode).

```
PROGRAM P_SerialComm_HighSpeed
VAR
    fbSerialLineCtrl    : FB_IEC870_SerialLineCtrl;
    Mode                : ComSerialLineMode_t := SERIALLINEMODE_KL6_5B_STANDARD; (* SERIALLINEMODE_PC
_COM_PORT *)

    serial_in AT%I*     : PcComInData;
    serial_out AT%Q*    : PcComOutData;
    KL6_in AT%I*        : KL6inData5B;
    KL6_out AT%Q*       : KL6outData5B;

    hSerial             : T_HSERIALCTRL;
END_VAR
```

```
fbSerialLineCtrl( Mode := Mode,
        Baudrate := 19200,
        NoDatabits := 8,
        Parity := PARITY_EVEN,
        Stopbits := 1,
        Handshake := HANDSHAKE_NONE,
        ContinousMode := FALSE,
        pComIn := SEL( Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR( serial_in ), ADR( KL6_in ) ),
        pComOut := SEL( Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR( serial_out ), ADR( KL6_out ) ),
        SizeComIn := SEL( Mode = SERIALLINEMODE_KL6_5B_STANDARD, SIZEOF( serial_in ), SIZEOF( KL6_in
 ) ),
        hSerial := hSerial );
```

Example 2:

In this example the communication takes place via an EL6001 (22-byte mode). The configuration of the EL6001 terminal (baud rate, parity etc.) must be carried out in the TwinCAT System Manager.

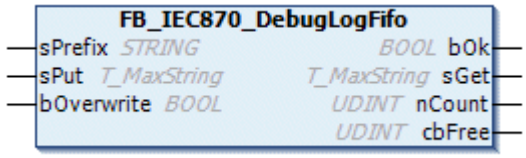
```
PROGRAM P_SerialComm_HighSpeed
VAR
    fbSerialLineCtrl    : FB_IEC870_SerialLineCtrl := ( Mode := SERIALLINEMODE_EL6_22B );
    EL6_in AT%I*        : EL6inData22B;
    EL6_out AT%Q*       : EL6outData22B;
    hSerial             : T_HSERIALCTRL;
END_VAR
```

```
fbSerialLineCtrl( pComIn := ADR( EL6_in ), pComOut := ADR( EL6_out ), SizeComIn := SIZEOF( EL6_in ),
 hSerial := hSerial );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.5 FB_IEC870_DebugLogFifo



This function block is a FIFO for status/debugging/error messages. In the default setting, the oldest entry keeps being overwritten. The FIFO has a constant internal buffer size. The size is determined by the constant: MAX_IEC870_DEBUGLOG_FIFOSIZE (default: 10000 bytes).

The function block features the following tasks:

- **A_Add** (adds a new message to the FIFO);
- **A_Remove** (removes the oldest message from the FIFO);
- A_**reset** (clears all messages, resets the FIFO);
- **A_LogHint** (adds a new message to the FIFO and also writes this message as a note in the TwinCAT XAE->"Error List" window);
- **A_LogWarning** (adds a new message to the FIFO and also writes this message as a warning in the TwinCAT XAE->"Error List" window);
- **A_LogError** (adds a new message to the FIFO and also writes this message as an error in the TwinCAT XAE->"Error List" window);

VAR_INPUT

```
VAR_INPUT
    sPrefix    : STRING;
    sPut       : T_MaxString;
    bOverwrite : BOOL := TRUE;
END_VAR
```

sPrefix: Additional message prefix.

sPut: Message to be added to the FIFO.

bOverwrite: TRUE => the oldest entries are overwritten, FALSE => the oldest entries are not overwritten.

VAR_OUTPUT

```
VAR_OUTPUT
    bOk     : BOOL;
    sGet    : T_MaxString;
    nCount  : UDINT;
    cbFree  : UDINT;
END_VAR
```

bOk: This variable becomes TRUE if a new entry was successfully added to or removed from the FIFO. This variable becomes FALSE on buffer overflow (overwriting of oldest entries disabled).

sGet: Message to be removed from the FIFO.

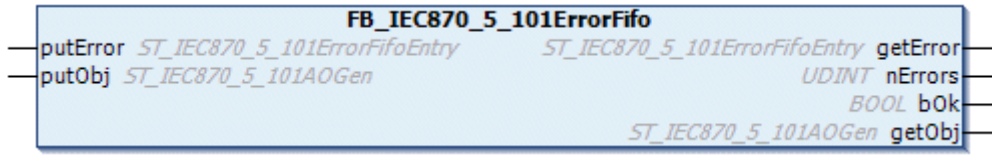
nCount: Number of current fifo entries.

cbFree : Number of free FIFO data bytes.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.1.6 FB_IEC870_5_101ErrorFifo



IEC 60870-5-10x error FIFO. The oldest entry is overwritten. The FIFO has a constant size. The size is determined by the constant: *IEC870_MAX_ERROR_FIFO_SIZE* (default: 10 elements).

The function block features three tasks:

- **AddError** (adds a new error message to the FIFO);
- **RemoveError** (removes the oldest error message from the FIFO);
- **Reset** (clears all error messages and resets the FIFO);

Error messages are normally added to the FIFO via the internal IEC 60870-5-10x device functions. The PLC application can read and analyses these error messages by calling the task: **RemoveError**.

VAR_INPUT

```
VAR_INPUT
    putError : ST_IEC870_5_101ErrorFifoEntry;
    putObj   : ST_IEC870_5_101AOGen;
END_VAR
```

putError: Error message [▶ 592] to be added to the FIFO.

putObj: Additional information about the application object (data point), to which the error message relates. This information is optional and not always present.

VAR_OUTPUT

```
VAR_OUTPUT
    getError: ST_IEC870_5_101ErrorFifoEntry;
    nErrors : UDINT;
    bOk     : BOOL;
    getObj  : ST_IEC870_5_101AOGen;
END_VAR
```

getError: Error message [▶ 592] to be removed from the FIFO.

nErrors: Returns the current number of FIFO entries (error messages in the FIFO).

bOk: This variable becomes TRUE if a new entry was successfully added to or removed from the FIFO.

getObj: Additional information about the application object (data point), to which the error message relates. This information is optional and not always present.

Sample in ST:

In the following ST example, the device error FIFO is read, and the registered errors are written to the TwinCAT XAE->"Error List".

```
PROGRAM P_LogErrors
VAR_IN_OUT
    fbErrors : FB_IEC870_5_101ErrorFifo;
END_VAR
```

```
REPEAT
    fbErrors.RemoveError();
    IF fbErrors.bOk THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'IEC 60870-5-10x device error: 0x%s',

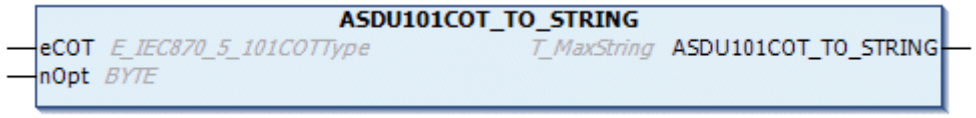
                DWORD_TO_HEXSTR( fbErrors.getError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT fbErrors.bOk
END_REPEAT
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2 Functions

5.2.1 ASDU101COT_TO_STRING



The function converts the IEC 60870-5-101/104 cause of transfer into a string (for debugging/ troubleshooting).

Syntax

```
FUNCTION ASDU101COT_TO_STRING: T_MaxString
VAR_INPUT
    eCOT : E_IEC870_101COTType;
    nOpt : BYTE;
END_VAR
```

Inputs

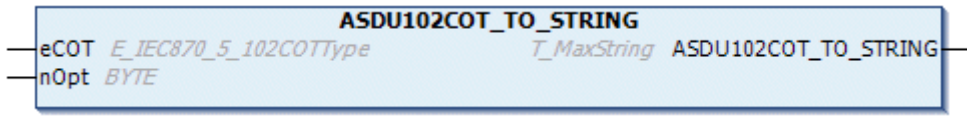
eCOT: Cause of transfer [▶ 611].

nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.2 ASDU102COT_TO_STRING



The function converts the IEC 60870-5-102 cause of transfer into a string (for debugging/troubleshooting).

Syntax

```
FUNCTION ASDU102COT_TO_STRING: T_MaxString
VAR_INPUT
    eCOT : E_IEC870_102COTType;
    nOpt : BYTE;
END_VAR
```

Inputs

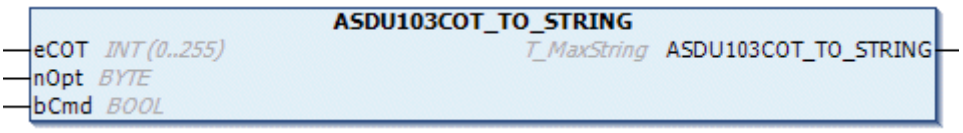
eCOT: Cause of transfer [▶ 614].

nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.3 ASDU103COT_TO_STRING



The function converts the IEC 60870-5-103 cause of transfer into a string (for debugging/troubleshooting).

Syntax

```
FUNCTION ASDU103COT_TO_STRING: T_MaxString
VAR_INPUT
    eCOT : INT(0..255);
    nOpt : BYTE;
    bCmd : BOOL;
END_VAR
```

Inputs

eCOT: Cause of transfer in monitoring direction [▶ 615] or in control direction [▶ 616].

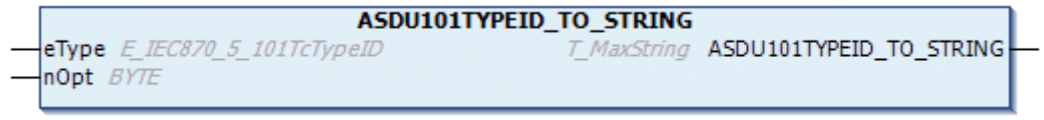
nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

bCmd: FALSE=>monitoring direction, TRUE=>control direction.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.4 ASDU101TYPEID_TO_STRING



The function converts the IEC 60870-5-101/104 ASDU type identifier into a string (for debugging/ troubleshooting).

Syntax

```
FUNCTION ASDU101TYPEID_TO_STRING: T_MaxString
VAR_INPUT
    eType : E_IEC870_101TcTypeID;
    nOpt  : BYTE;
END_VAR
```

Inputs

eCOT: ASDU type identifier. [▶ 601]

nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.5 ASDU102TYPEID_TO_STRING



The function converts the IEC 60870-5-102 ASDU type identifier into a string (for debugging/ troubleshooting).

Syntax

```
FUNCTION ASDU102TYPEID_TO_STRING: T_MaxString
VAR_INPUT
    eType : E_IEC870_102TcTypeID;
    nOpt  : BYTE;
END_VAR
```

Inputs

eCOT: ASDU type identifier [▶ 605].

nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.6 ASDU103TYPEID_TO_STRING



The function converts the IEC 60870-5-103 ASDU type identifier into a string (for debugging/ troubleshooting).

Syntax

```
FUNCTION ASDU103TYPEID_TO_STRING: T_MaxString
VAR_INPUT
    eType : INT(0..255);
    nOpt  : BYTE;
    bCmd  : BOOL;
END_VAR
```

Inputs

eCOT: ASDU type identifier in monitoring direction [▶ 607] or in control direction [▶ 607].

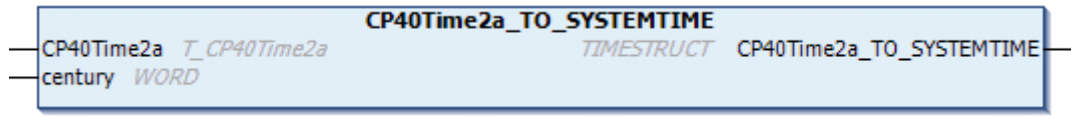
nOpt: Options. Bit 0 := FALSE => exact type identifier, bit 0 := TRUE => short type identifier. Bit 1 := TRUE => numeric hexadecimal/decimal value, bit 2 := TRUE => additional descriptive text.

bCmd: FALSE => monitoring direction, TRUE => control direction.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.7 CP40Time2a_TO_SYSTEMTIME



The function converts the CP40Time2a time format [▶ 635] to Windows system time format. The SU flag is not used.

Syntax

```
FUNCTION CP40Time2a_TO_SYSTEMTIME: TIMESTRUCT
VAR_INPUT
    CP40Time2a  : T_CP40Time2a;
    century     : WORD;
END_VAR
```

Inputs

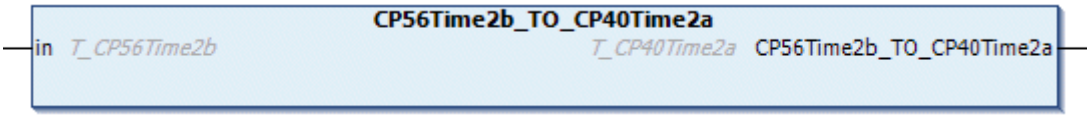
CP40Time2a: The time to be converted in CP40Time2a format.

century: The century (e.g. 20 for the year 2005). This information is not included in the CP40Time2a format and therefore has to be provided separately.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.8 CP56Time2b_TO_CP40Time2a



The function converts the CP56Time2b time format [▶ 634] to CP40Time2a time format [▶ 635].

Syntax

```
FUNCTION CP56Time2b_TO_CP40Time2a: T_CP40Time2a
VAR_INPUT
    in : T_CP56Time2b;
END_VAR
```

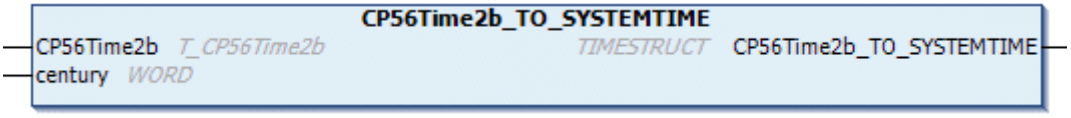
Inputs

in: The time to be converted in CP56Time2b format.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.9 CP56Time2b_TO_SYSTEMTIME



The function converts the CP56Time2b time format [▶ 634] to Windows system time format. The SU flag is not used.

Syntax

```
FUNCTION CP56Time2b_TO_SYSTEMTIME: TIMESTRUCT
VAR_INPUT
    CP56Time2b : T_CP56Time2b;
    century    : WORD;
END_VAR
```

Inputs

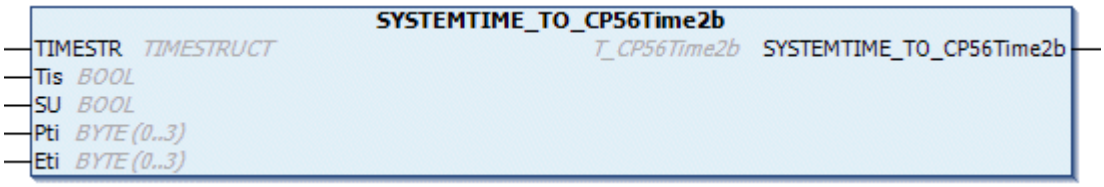
CP56Time2b: The time to be converted in CP56Time2b format.

century: The century (e.g. 20 for the year 2005). This information is not included in the CP56Time2b format and therefore has to be provided separately.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.10 SYSTEMTIME_TO_CP56Time2b



The function converts the Windows system time format to the CP56Time2b time format [▶ 634]. All reserved bits are null.

Syntax

```
FUNCTION SYSTEMTIME_TO_CP56Time2b: T_CP56Time2b
VAR_INPUT
    TIMESTR : TIMESTRUCT;
    Tis     : BOOL;
    SU      : BOOL;
    Pti     : BYTE(0..3);
    Eti     : BYTE(0..3);
END_VAR
```

Inputs

TIMESTR: The system time to be converted.

Tis: Tariff information.

SU: Summer/winter time format. This information is not available in the TIMESTR format and must be specified additionally. TRUE = summer time, FALSE = winter time (normal time).

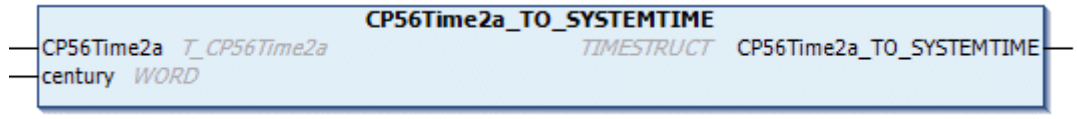
Pti: Power tariff information.

Eti: Energy tariff information.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.11 CP56Time2a_TO_SYSTEMTIME



The function converts the CP56Time2a time format [▶ 633] to Windows system time format. The SU flag is not used.

Syntax

```
FUNCTION CP56Time2a_TO_SYSTEMTIME: TIMESTRUCT
VAR_INPUT
    CP56Time2a : T_CP56Time2a;
    century    : WORD;
END_VAR
```

Inputs

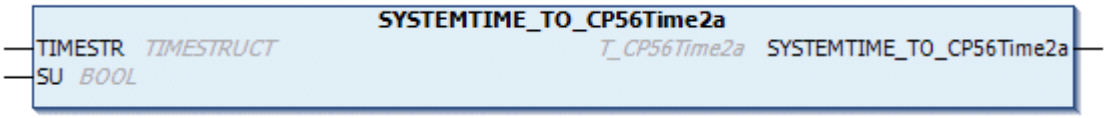
CP56Time2a: The time to be converted in CP56Time2a format.

century: The century (e.g. 20 for the year 2005). This information is not included in the CP56Time2a format and therefore has to be provided separately.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.12 SYSTEMTIME_TO_CP56Time2a



The function converts the Windows system time format to the CP56Time2a time format [▶ 633]. All reserved bits are null.

Syntax

```
FUNCTION SYSTEMTIME_TO_CP56Time2a: T_CP56Time2a
VAR_INPUT
    TIMESTR : TIMESTRUCT;
    SU      : BOOL;
END_VAR
```

Inputs

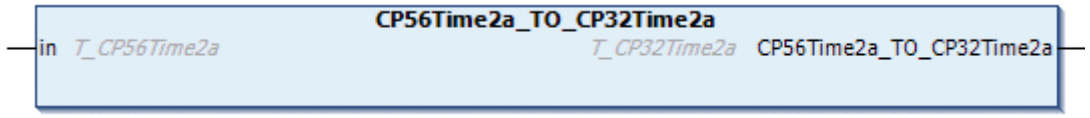
TIMESTR: The system time to be converted.

SU: Summer/winter time format. This information is not available in the TIMESTR format and must be specified additionally. TRUE = summer time, FALSE = winter time (normal time).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.13 CP56Time2a_TO_CP32Time2a



The function converts the CP56Time2a time format [▶ 633] to CP32Time2a time format [▶ 635].

Syntax

```
FUNCTION CP56Time2a_TO_CP32Time2a: T_CP32Time2a
VAR_INPUT
    in : T_CP56Time2a;
END_VAR
```

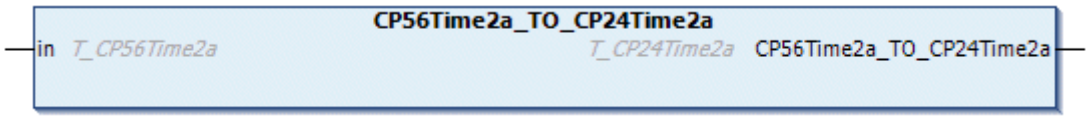
Inputs

in: The time to be converted in CP56Time2a format.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.14 CP56Time2a_TO_CP24Time2a



The function converts the CP56Time2a time format [▶ 633] to CP24Time2a time format [▶ 636].

Syntax

```
FUNCTION CP56Time2a_TO_CP24Time2a: T_CP24Time2a
VAR_INPUT
    in : T_CP56Time2a;
END_VAR
```

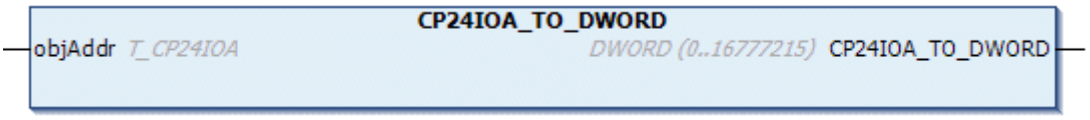
Inputs

in: The time to be converted in CP56Time2a format.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.15 CP24IOA_TO_DWORD



The function generates a structured TwinCAT object address (3 octets). See also: DWORD_TO_CP24IOA [▶ 533].

Syntax

```
FUNCTION CP24IOA_TO_DWORD: DWORD
VAR_INPUT
    objAddr : T_CP24IOA;
END_VAR
```

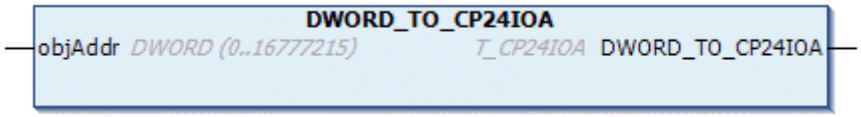
Inputs

objAddr: Parameter of the structured TwinCAT object address [▶ 637].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.16 DWORD_TO_CP24IOA



The function converts a structured TwinCAT object address [▶ 637] into individual address parameters. See also: CP24IOA_TO_DWORD [▶ 532].

Syntax

```
FUNCTION DWORD_TO_CP24IOA: T_CP24IOA
VAR_INPUT
    objAddr : DWORD(0..16777215);
END_VAR
```

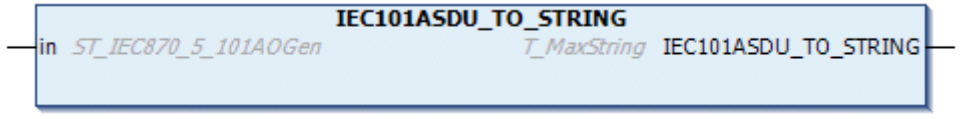
Inputs

objAddr: structured TwinCAT object address (3 octets).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.17 IEC101ASDU_TO_STRING



The function converts the information of an IEC 60870-5-101/104 ASDU to a string (for debugging purposes/ troubleshooting).

Syntax

```
FUNCTION IEC101ASDU_TO_STRING: T_MaxString
VAR_INPUT
    in : ST_IEC870_5_101AOGen;
END_VAR
```

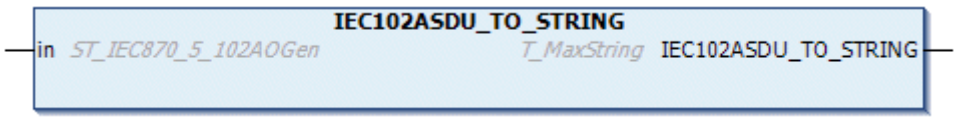
Inputs

in: Structured variable with the ASDU to be converted.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.18 IEC102ASDU_TO_STRING



The function converts the information of an IEC 60870-5-102 ASDU to a string (for debugging purposes/ troubleshooting).

Syntax

```
FUNCTION IEC102ASDU_TO_STRING: T_MaxString
VAR_INPUT
    in : ST_IEC870_5_102AOGen;
END_VAR
```

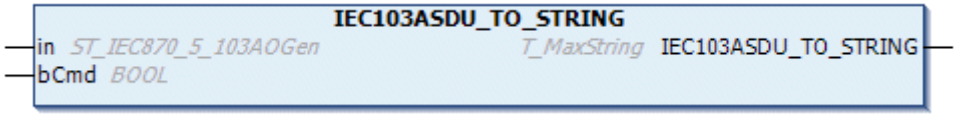
Inputs

in: Structured variable with the ASDU to be converted.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.19 IEC103ASDU_TO_STRING



The function converts the information of an IEC 60870-5-103 ASDU to a string (for debugging purposes/ troubleshooting).

Syntax

```
FUNCTION IEC103ASDU_TO_STRING: T_MaxString
VAR_INPUT
    in   : ST_IEC870_5_103AOGen;
    bCmd : BOOL;
END_VAR
```

Inputs

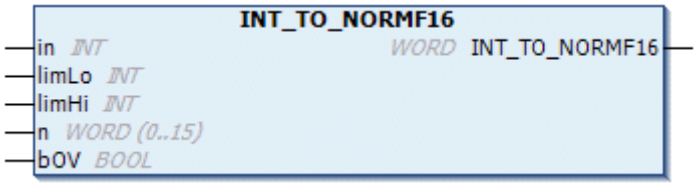
in: Structured variable with the ASDU to be converted.

bCmd: FALSE => ASDU in monitoring direction, TRUE => ASDU in control direction.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.20 INT_TO_NORMF16



This function normalizes a 16-bit integer value and converts it to an NVA value (normalized value) in fixed-point number format.

Syntax

```
FUNCTION INT_TO_NORMF16: WORD
VAR_INPUT
    in    : INT;
    limLo : INT;
    limHi : INT;
    n     : WORD(0..15) := 15;
    bOV   : BOOL;
END_VAR
```

Inputs

in: Value to be converted.

limLo: Min. value limit.

limHi: Max. value limit.

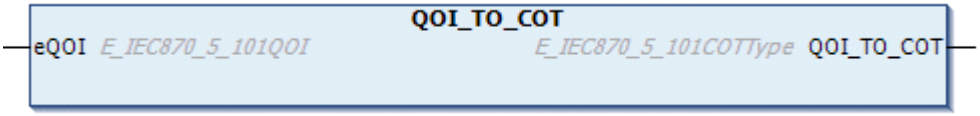
n: Number of decimal places of the fixed-point number.

bOV: TRUE => Value overrange or underrange.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.21 QOI_TO_COT



The function converts the "qualifier" of the station interrogation command to the expected cause of transfer for the interrogated data.

Syntax

```
FUNCTION QOI_TO_COT: E_IEC870_5_101COTType
VAR_INPUT
    eQOI : E_IEC870_5_101QOI;
END_VAR
```

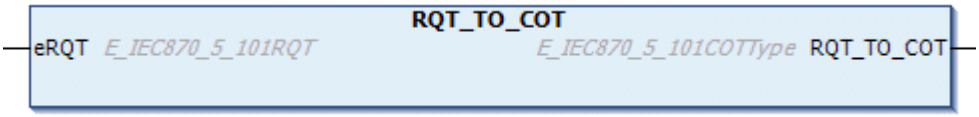
Inputs

eQOI: Qualifier of the station interrogation command [▶ 623].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.22 RQT_TO_COT



The function converts the "qualifier" of the counter interrogation command to the expected cause of transfer for the interrogated data.

Syntax

```
FUNCTION RQT_TO_COT: E_IEC870_5_101COTType
VAR_INPUT
    eRQT : E_IEC870_5_101RQT;
END_VAR
```

Inputs

eRQT: Qualifier of the counter interrogation command [▶ 625].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.23 F_iecInitAOEntry



The function F_iecInitAOEntry configures the application objects (single points, double points, measured values...) in the application database as linear table entries. The table element (array element) to be configured must be transferred to the function as a VAR_IN_OUT function parameter.

Syntax

```
FUNCTION F_iecInitAOEntry: UDINT
VAR_INPUT
    eType       : E_IEC870_5_101TcTypeID := ASDU_TYPEUNDEF;
    objAddr     : DWORD := 0;
    group       : DWORD := 0;
    multiplier  : BYTE := 0;
    ioMapType   : E_IEC870_5_101IOMappingType := MAP_AREA_NONE;
    byteOffs    : UXINT := 0;
    bitOffs     : UDINT := 0;
END_VAR
VAR_IN_OUT
    dbEntry     : ST_IEC870_5_101AODBEntry;
END_VAR
```

Inputs

eType: Application object type [▶ 601], ASDU identifier (e.g.: M_SP_NA_1 for single-point or M_DP_NA_1 for double point).

objAddr: Object address, freely selectable.

group: Object group configuration flags. A description of all group flags [▶ 655] can be found here. The flags can be combined with an OR operation. Not all combinations are meaningful!

multiplier: Basic cycle time multiplier for cyclic/periodic data transfer. 0 = deactivated. The basic cycle time can be configured via the *tPerCyclicBase* parameter in the system parameters [▶ 588].

ioMapType: TwinCAT PLC process data area [▶ 608]. This parameter defines the mapping for the TwinCAT PLC and IEC application object process data.

byteOffs: TwinCAT PLC process data byte offset (x86 platform => 32 bit, x64 platform => 64 bit).

bitOffs: TwinCAT PLC process data bit offset.

Inputs/outputs

dbEntry: The table element [▶ 584] (array element) to be configured.

Return value

Return parameter	Description
0	No error.
<> 0	Error: IEC60870-5-10x error code [▶ 658]

Sample in ST

In the following sample some data points are configured as linear table entries.

eType	objAddr	group	multiplier	ioMapType	byteOffs	bitOffs
M_SP_NA_1	100	IEC870_GRP_INRO1	0	MAP_AREA_MEMORY	100	0
M_SP_TB_1	101	IEC870_GRP_INRO1	0	MAP_AREA_MEMORY	101	0
M_DP_NA_1	200	IEC870_GRP_INROGEN	0	MAP_AREA_DATA	200	0
M_IT_NA_1	800	IEC870_GRP_REQCOGEN	0	MAP_AREA_MEMORY	800	0

```
VAR_GLOBAL CONSTANT
    MAX_TABLE_IDX : INT := 49;
END_VAR
```

```
PROGRAM P_iecInitAOEntry
VAR_IN_OUT
    AODB    : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    bInit   : BOOL := TRUE;
    nError  : UDINT;
END_VAR
```

```
IF bInit THEN
    bInit := FALSE;
    IF ( nError := F_iecInitAOEntry ( eType := M_SP_NA_1,
                    objAddr     := 100,
                    group       := IEC870_GRP_INRO1,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 100,
                    bitOffs     := 0,
                    dbEntry     := AODB[1] ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecInitAOEntry() error: %s',
                    DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecInitAOEntry ( eType := M_SP_TB_1,
                    objAddr     := 101,
                    group       := IEC870_GRP_INRO1,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 101,
                    bitOffs     := 0,
                    dbEntry     := AODB[2] ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecInitAOEntry() error: %s',
                    DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecInitAOEntry ( eType := M_DP_NA_1,
                    objAddr     := 200,
                    group       := IEC870_GRP_INROGEN,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_DATA,
```

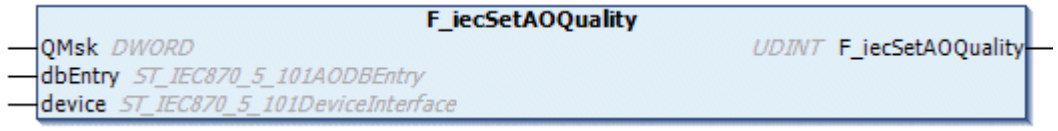
```
                    byteOffs    := 200,
                    bitOffs     := 0,
                    dbEntry     := AODB[3] ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecInitAOEntry() error: %s',
                    DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecInitAOEntry ( eType := M_IT_NA_1,
                    objAddr     := 800,
                    group       := IEC870_GRP_REQCOGEN,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 800,
                    bitOffs     := 0,
                    dbEntry     := AODB[4] ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecInitAOEntry() error: %s',
                    DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.24 F_iecSetAOQuality



This function can be used to set/reset quality flags of an application object to a particular value.

Syntax

```
FUNCTION F_iecSetAOQuality: UDINT
VAR_INPUT
    QMsk    : DWORD;
END_VAR
VAR_IN_OUT
    dbEntry : ST_IEC870_5_101AODBEntry;
    device  : ST_IEC870_5_101DeviceInterface;
END_VAR
```

Inputs

QMsk: Quality flags. The quality flags can be combined with OR links. If QMsk = null no flags are set/reset. A list of available quality flags can be found here: Quality flags [▶ 657].

Inputs/outputs

dbEntry: Application object [▶ 584] whose quality flag status is to be set.

device: Communication interface [▶ 591] of the IEC device.

Return value

Return parameter	Meaning
0	No error.
<> 0	Error: IEC60870-5-10x error code [▶ 658]

Sample in ST

```
VAR_GLOBAL CONSTANT
    MAX_TABLE_IDX : INT := 49;
END_VAR
```

```
PROGRAM P_iecGetAOQuality
VAR_IN_OUT
    AODB          : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    server1       : FB_IEC870_5_104Slave;

    bBlock        : BOOL;
    bUnblock      : BOOL;
    bIsBlocked    : BOOL;

    qualityFlags : DWORD;
END_VAR
```

```
IF bBlock THEN
    bBlock := FALSE;
    F_iecSetAOQuality( IECQ_BL_ON, AODB[1], server1.system.device );
END_IF

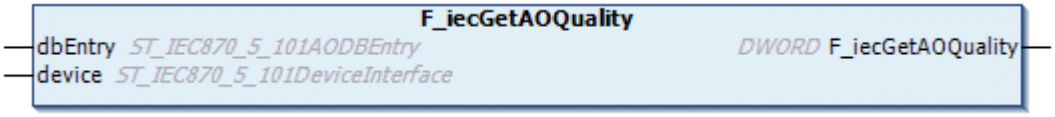
IF bUnblock THEN
    bUnblock := FALSE;
    F_iecSetAOQuality( IECQ_BL_OFF, AODB[1], server1.system.device );
END_IF

qualityFlags := F_iecGetAOQuality ( AODB[1], server1.system.device );
bIsBlocked := SEL( (qualityFlags AND IECQ_BL_ON ) = IECQ_BL_ON, FALSE, TRUE );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.25 F_iecGetAOQuality



The function can be used to read the status of the quality flags of an application object.

Syntax

```
FUNCTION F_iecGetAOQuality: DWORD
VAR_IN_OUT
    dbEntry : ST_IEC870_5_101AODBEntry;
    device  : ST_IEC870_5_101DeviceInterface;
END_VAR
```

Inputs/outputs

dbEntry: Application object [▶ 584] whose quality flags are to be read.

device: Communication interface [▶ 591] with the IEC device.

Return value

Return parameter	Meaning
0	Error, no quality flags available for this application object.
<> 0	No error. The return parameter returns the status of the quality flags. A list of available quality flags can be found here: Quality flags [▶ 657].

Sample in ST

```
VAR_GLOBAL CONSTANT
    MAX_TABLE_IDX : INT := 49;
END_VAR
```

```
PROGRAM P_iecGetAOQuality
VAR_IN_OUT
    AODB          : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    server1       : FB_IEC870_5_104Slave;

    bBlock        : BOOL;
    bUnblock      : BOOL;
    bIsBlocked    : BOOL;

    qualityFlags : DWORD;
END_VAR
```

```
IF bBlock THEN
    bBlock := FALSE;
    F_iecSetAOQuality( IECQ_BL_ON, AODB[1], server1.system.device );
END_IF

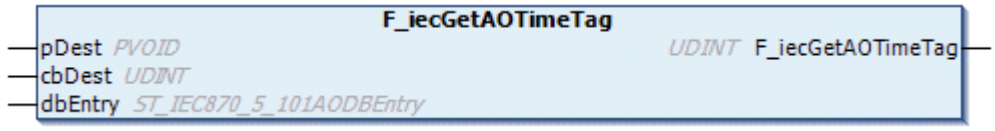
IF bUnblock THEN
    bUnblock := FALSE;
    F_iecSetAOQuality( IECQ_BL_OFF, AODB[1], server1.system.device );
END_IF

qualityFlags := F_iecGetAOQuality ( AODB[1], server1.system.device );
bIsBlocked := SEL( (qualityFlags AND IECQ_BL_ON ) = IECQ_BL_ON, FALSE, TRUE );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.26 F_iecGetAOTimeTag



This function can be used to read the current timestamp of the application object in a byte buffer.

Syntax

```
FUNCTION F_iecGetAOTimeTag: UDINT
VAR_INPUT
    pDest   : PVOID;
    cbDest  : UDINT;
END_VAR
VAR_IN_OUT
    dbEntry : ST_IEC870_5_101AODBEntry;
END_VAR
```

Inputs

pDest: Target buffer address.

cbDest: Byte size of the target buffer.

Inputs/outputs

dbEntry: Application object [▶ 584] whose timestamp is to be read.

Return value

Return parameter	Meaning
0	Error; the application object has no timestamp.
<> 0	Number of successfully copied timestamp data bytes. The CP24Time2a timestamp format has 3 bytes, for example, the CP56Time2a timestamp format has 7 bytes.

Sample in ST

```
PROGRAM P_iecGetAOTimeTag
VAR_IN_OUT
    AODB     : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    bGet     : BOOL := TRUE;
    index    : DINT;
    cbTagSize: UDINT;
    tTag     : T_CP56Time2a;
END_VAR
```

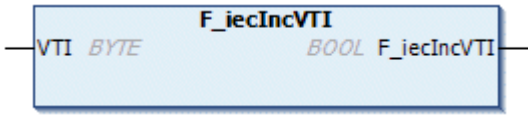
```
IF bGet THEN
    bGet    := FALSE;

    FOR index := 1 TO MAX_TABLE_IDX BY 1 DO
        cbTagSize := F_iecGetAOTimeTag ( ADR( tTag ), SIZEOF( tTag ), AODB[index] );
    END_FOR
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.27 F_iecIncVTI



This function increments the step size, which controls the INT7. The transient bit is not changed.

Syntax

```
FUNCTION F_iecIncVTI: BOOL
VAR_IN_OUT
    VTI : BYTE;
END_VAR
```

Inputs/outputs

VTI: The byte to be incremented.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.28 F_iecDecVTI



This function decrements the step size, which controls the INT7. The transient bit is not changed.

Syntax

```
FUNCTION F_iecDecVTI: BOOL
VAR_IN_OUT
    VTI     : BYTE;
END_VAR
```

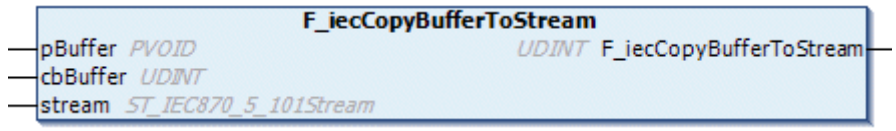
Inputs/outputs

VTI: The byte to be decremented.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.29 F_iecCopyBufferToStream



This function copies data bytes from an external buffer variable to the *stream* variable. The memory content of the *stream* variable is increased. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecCopyBufferToStream: UDINT
VAR_INPUT
    pBuffer : PVOID;
    cbBuffer: UDINT;
END_VAR
VAR_IN_OUT
    stream  : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

pBuffer: Pointer (address) of an external buffer variable.

cbBuffer: Number of data bytes to be copied from the external buffer variable to the *stream* variable.

Inputs/outputs

stream: Target data buffer [▶ 587].

Sample in ST

A rising edge at *bTx* results in copying of 2 x 4 data bytes from the *txBuffer* variable to the *stream* variable.

```
PROGRAM P_iecCopyBufferToStream
VAR
    stream   : ST_IEC870_5_101Stream;
    txBuffer : ARRAY[0..3] OF BYTE := [1, 2, 3, 4];
    cbResult : UDINT;
    bTx      : BOOL;
END_VAR
```

```
IF bTx THEN
    bTx := FALSE;
    cbResult := F_iecResetStream( 0, stream );
    cbResult := F_iecCopyBufferToStream ( ADR( txBuffer ), SIZEOF( txBuffer ), stream );
    cbResult := F_iecCopyBufferToStream ( ADR( txBuffer ), SIZEOF( txBuffer ), stream );
END_IF
```

Memory representation of the *stream* variable before the first function call:

length	data											
0	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

Memory representation of the *stream* variable after the first function call:

length	data										
4	16#01	16#02	16#03	16#04	16#00	16#00	16#00	16#00	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

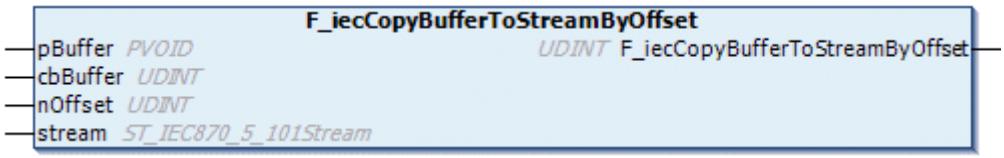
Memory representation of the *stream* variable after the second function call:

length	data										
8	16#01	16#02	16#03	16#04	16#01	16#02	16#03	16#04	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.30 F_iecCopyBufferToStreamByOffset



This function copies data bytes from an external buffer variable to the *stream* variable, starting from a byte offset position. The memory content of the *stream* variable is increased, depending on the previous length and the specified byte offset. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecCopyBufferToStreamByOffset: UDINT
VAR_INPUT
    pBuffer  : PVOID;
    cbBuffer : UDINT;
    nOffset  : UDINT;
END_VAR
VAR_IN_OUT
    stream   : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

pBuffer: Pointer (address) of an external buffer variable.

cbBuffer: Number of data bytes to be copied from the external buffer variable to the *stream* variable.

nOffset: Byte offset position in the target data buffer.

Inputs/outputs

stream: Target data buffer [▶ 587].

Sample in ST

A rising edge at *bTx* results in copying of 4 data bytes from the *txAtOffs* variable to the *stream* variable (from byte offset position: 5).

```
PROGRAM P_iecCopyBufferToStreamByOffset
VAR
    bTx      : BOOL;
    stream   : ST_IEC870_5_101Stream := (length := 8, data := [16#01, 16#02, 16#03, 16#04, 16#05, 16
#06, 16#07, 16#08] );
    txAtOffs : DWORD := 16#DDCCBBAA;
    cbResult : UDINT;
END_VAR
```

```
IF bTx THEN
    bTx := FALSE;
    cbResult := F_iecCopyBufferToStreamByOffset ( ADR( txAtOffs), SIZEOF(txAtOffs), 5, stream );
      (* stream.length == 9 *)
END_IF
```

Memory representation of the *stream* variable after the program start:

length	data										
8	16#01	16#02	16#03	16#04	16#05	16#06	16#07	16#08	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

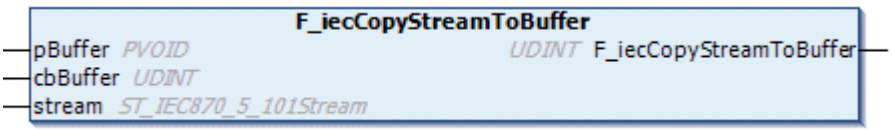
Memory representation of the *stream* variable after the F_iecCopyBufferToStreamByOffset function call:

length	data										
9	16#01	16#02	16#03	16#04	16#05	16#AA	16#BB	16#CC	16#DD	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.31 F_iecCopyStreamToBuffer



This function copies data bytes from the *stream* variable to an external buffer variable. The memory content of the *stream* variable remains unchanged. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecCopyStreamToBuffer: UDINT
VAR_INPUT
    pBuffer : PVOID;
    cbBuffer: UDINT;
END_VAR
VAR_IN_OUT
    stream  : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

pBuffer: Pointer (address) of an external buffer variable.

cbBuffer: Maximum number of data bytes to be copied from the *stream* variable to the external buffer.

Inputs/outputs

stream: Source data buffer [▶ 587].

Sample in ST

A rising edge at bRx results in copying of the first four data bytes of the *stream* variable to the *rxBufferA* and *rxBufferB* variables.

```
PROGRAM P_iecCopyStreamToBuffer
VAR
    stream    : ST_IEC870_5_101Stream := ( length := 4, data := [16#01, 16#02, 16#03, 16#04] );
    rxBufferA : ARRAY[0..3] OF BYTE;
    rxBufferB : ARRAY[0..3] OF BYTE;
    cbResult  : UDINT;
    bRx       : BOOL;
END_VAR
```

```
IF bRx THEN
    bRx := FALSE;
    cbResult := F_iecCopyStreamToBuffer ( ADR( rxBufferA ), SIZEOF( rxBufferA ), stream );
    cbResult := F_iecCopyStreamToBuffer ( ADR( rxBufferB ), SIZEOF( rxBufferB ), stream );
END_IF
```

Memory representation of the *stream* variable after the program start:

length	data											
4	16#01	16#02	16#03	16#04	...	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Memory representation of the *stream* variable after the first and second F_CopyStreamToBuffer function calls:

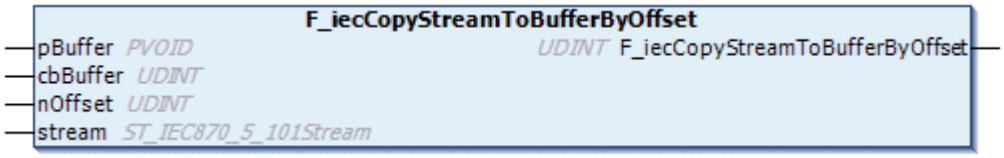
length	data											
4	16#01	16#02	16#03	16#04	...	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

The *rxBufferA* and *rxBufferB* array variables have the following values after the call: [16#01, 16#02, 16#03, 16#04].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.32 F_iecCopyStreamToBufferByOffset



This function copies data bytes from the *stream* variables into an external buffer variable from a byte offset position. The memory content of the *stream* variable remains unchanged. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecCopyStreamToBufferByOffset: UDINT
VAR_INPUT
    pBuffer : PVOID;
    cbBuffer: UDINT;
    nOffset : UDINT;
END_VAR
VAR_IN_OUT
    stream  : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

pBuffer: Pointer (address) of an external buffer variable.

cbBuffer: Maximum number of data bytes to be copied from the *stream* variable to the external buffer.

nOffset: Byte offset position of the source data buffer.

Inputs/outputs

stream: Source data buffer [▶ 587].

Sample in ST

A rising edge at bRx results in copying of four data bytes from byte offset 1 of the *stream* variable into the *rxAtOffs* variable.

```
PROGRAM P_iecCopyStreamToBufferByOffset
VAR
    stream   : ST_IEC870_5_101Stream := ( length := 8, data := [16#01, 16#AA, 16#BB, 16#CC, 16#DD, 1
6#06, 16#07, 16#08] );
    bRx      : BOOL;
    rxAtOffs : DWORD := 0;
    cbResult : UDINT := 0;
END_VAR
```

```
IF bRx THEN
    bRx := FALSE;
    cbResult := F_iecCopyStreamToBufferByOffset ( ADR( rxAtOffs ), SIZEOF( rxAtOffs ), 1, stream );
END_IF
```

Memory representation of the *stream* variable after the program start:

length	data										
8	16#01	16#AA	16#BB	16#CC	16#DD	16#06	16#07	16#08	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

Memory representation of the *stream* variable after the F_iecCopyStreamToBufferByOffset function call:

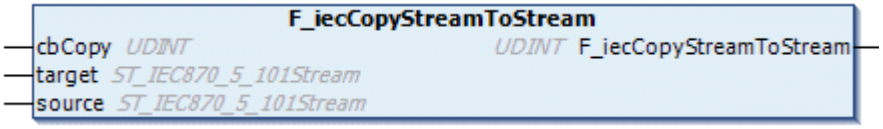
length	data										
8	16#01	16#AA	16#BB	16#CC	16#DD	16#06	16#07	16#08	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

The variable *rxAtOffs* has the value: 16#DDCCBBAA.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.33 F_iecCopyStreamToStream



This function copies data bytes from the *source* variable to the *target* variable. The memory content of the *source* variable remains unchanged. The memory content of the *target* variable is increased. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecCopyStreamToStream: UDINT
VAR_INPUT
    cbCopy  : UDINT;
END_VAR
VAR_IN_OUT
    target  : ST_IEC870_5_101Stream;
    source  : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

cbCopy: Number of bytes to be copied from the *source* variable to the *target* variable.

Inputs/outputs

target: Target data buffer [▶ 587].

source: Source data buffer [▶ 587].

Sample in ST

A rising edge at *bCopy* results in incrementing of *srcValue*, followed by copying to *srcStream*. Then, the first 4 data bytes are copied from *srcStream* to *dstStream*. Finally, the first 4 data bytes are copied from *dstStream* to the *dstValue* variable.

```
PROGRAM P_iecCopyStreamToStream
VAR
    srcStream   : ST_IEC870_5_101Stream;
    srcValue    : DWORD;

    dstStream   : ST_IEC870_5_101Stream;
    dstValue    : DWORD;

    bCopy       : BOOL;
END_VAR
```

```
IF bCopy THEN
    bCopy := FALSE;
    srcValue := srcValue + 1;
    F_iecCopyBufferToStream( ADR( srcValue ), SIZEOF( srcValue ), srcStream );
    F_iecCopyStreamToStream ( SIZEOF( srcValue ), dstStream, srcStream );
    F_iecCopyStreamToBuffer( ADR( dstValue ), SIZEOF( dstValue ), dstStream );
END_IF
```

Memory representation of the *srcStream* and *dstStream* variables after the first F_iecCopyStreamToStream function call:

Table 80: srcStream:

length	data											
4	16#01	16#00	16#00	16#00	...	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Table 81: dstStream:

length	data											
4	16#01	16#00	16#00	16#00	...	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Memory representation of the *srcStream* and *dstStream* variables after the second F_iecCopyStreamToStream function call:

Table 82: srcStream:

length	data											
8	16#01	16#00	16#00	16#00	16#02	16#00	16#00	16#00	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

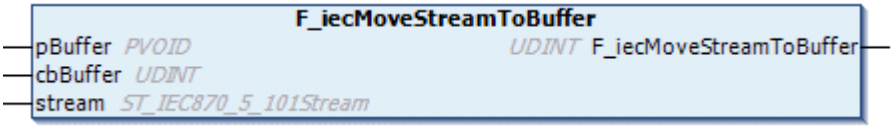
Table 83: dstStream:

length	data											
8	16#01	16#00	16#00	16#00	16#01	16#00	16#00	16#00	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.34 F_iecMoveStreamToBuffer



This function copies data bytes from the *stream* variable into an external buffer variable and then deletes the copied data bytes from the *stream* variable. The memory content of the *stream* variable is reduced. The return parameter of the function returns the number of successfully copied data bytes.

Syntax

```
FUNCTION F_iecMoveStreamToBuffer : UDINT
VAR_INPUT
    pBuffer  : PVOID;
    cbBuffer : UDINT;
END_VAR
VAR_IN_OUT
    stream   : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

pBuffer: Pointer (address) of an external buffer variable.

cbBuffer: Maximum number of data bytes to be copied from the *stream* variable to the external buffer.

Inputs/outputs

stream: Source data buffer [▶ 587].

Sample in ST

After the program start, 8 byte values are initialized in the *stream* variable. A rising edge at *bRx* results in copying of 4 data bytes from the *stream* variable into the *rxBuffer* variable.

```
PROGRAM P_iecMoveStreamToBuffer
VAR
    stream   : ST_IEC870_5_101Stream := ( length := 8, data := [16#78, 16#56, 16#34, 16#12, 16#01, 1
6#EF, 16#CD, 16#AB] );
    rxBuffer : DWORD;
    cbResult : UDINT;
    bRx      : BOOL := TRUE;
END_VAR
```

```
IF bRx THEN
    bRx := FALSE;
    cbResult := F_iecMoveStreamToBuffer( ADR( rxBuffer ), SIZEOF( rxBuffer ), stream );
END_IF
```

Memory representation of the stream variable after the program start:

length	data											
8	16#78	16#56	16#34	16#12	16#01	16#EF	16#CD	16#AB	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Memory representation of the stream variable after the first F_iecMoveStreamToBuffer function call:

length	data											
4	16#01	16#EF	16#CD	16#AB	16#01	16#EF	16#CD	16#AB	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

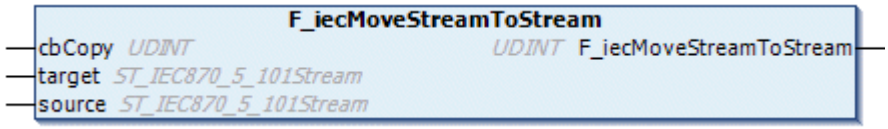
Memory representation of the stream variable after the second F_iecMoveStreamToBuffer function call:

length	data											
0	16#01	16#EF	16#CD	16#AB	16#01	16#EF	16#CD	16#AB	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.35 F_iecMoveStreamToStream



This function copies data bytes from the *source* variable to the *target* variable and then deletes the copied data bytes from the *source* variable. The memory content of the *source* variable is reduced, the content of the *target* variable is increased.

Syntax

```
FUNCTION F_iecMoveStreamToStream : UDINT
VAR_INPUT
    cbCopy  : UDINT;
END_VAR
VAR_IN_OUT
    target  : ST_IEC870_5_101Stream;
    source  : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

cbCopy: Number of data bytes to be copied from the *source variable* to the *target variable.*

Inputs/outputs

target: Target data buffer [▶ 587].

source: Source data buffer [▶ 587].

Sample in ST

```
PROGRAM P_iecMoveStreamToStream
VAR
    target : ST_IEC870_5_101Stream := ( length := 2, data := [16#AA, 16#BB] );
    source : ST_IEC870_5_101Stream := ( length := 8, data := [1,2,3,4,5,6,7,8] );
    cbReturn : UDINT;
    bMove : BOOL := TRUE;
END_VAR
```

```
IF bMove THEN
    bMove := FALSE;
    cbReturn := F_iecMoveStreamToStream( source.length, target,  source );
END_IF
```

Content of the *target* variable after program start:

length	data											
10	16#AA	16#BB	16#01	16#02	16#03	16#04	16#05	16#06	16#07	16#08	...	IEC870_MAX_ASDU_DATA_BYTE

Content of the *source* variable after the program start:

length	data											
0	16#01	16#02	16#03	16#04	16#05	16#06	16#07	16#08	16#00	16#00	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.36 F_iecResetStream



The function initializes and resets the *stream* variable. Optionally, the internal data buffer of the *stream* variable can be initialized with a certain number of zero bytes. The return value of the function returns the number of successfully initialized zero bytes.

Syntax

```
FUNCTION F_iecResetStream: UDINT
VAR_INPUT
    cbLength : UDINT;
```

```
END_VAR
VAR_IN_OUT
    stream   : ST_IEC870_5_101Stream;
END_VAR
```

Inputs

cbLength: Number of zero bytes to be initialized.

Inputs/outputs

stream: Buffer variable [▶ 587] to be initialized.

Sample in ST

After the program start, the internal buffer of the *stream* variable is reset and initialized with 5 zero bytes.

```
PROGRAM P_ResetStream
VAR
    stream : ST_IEC870_5_101Stream;
    bReset : BOOL := TRUE;
END_VAR
```

```
IF bReset THEN
    bReset := FALSE;
    F_iecResetStream( 5, stream );
END_IF
```

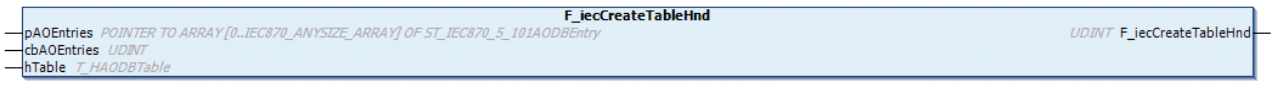
Memory content of the stream variable after the program start:

length	data											
5	16#00	16#00	16#00	16#00	16#00	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.37 F_iecCreateTableHnd



The function F_iecCreateTableHnd initializes the application object database handle (hash table handle). The table handle must be initialized once, before it can be used.

Syntax

```
FUNCTION F_iecCreateTableHnd: UDINT
VAR_INPUT
    pAOEntries  : POINTER TO ST_IEC870_5_101AODBEntry := 0;
    cbAOEntries : UDINT := 0;
END_VAR
```

```
VAR_IN_OUT
    hTable      : T_HAODBTable;
ENd_VAR
```

Inputs

pAOEntries: Address of the first application object database variable.

cbAOEntries: Byte size of the application object database variables.

Inputs/outputs

hTable: The application object database handle [▶ 633] (hash table handle) to be initialized.

Return value

Return parameter	Description
0	No error.
<> 0	Error: IEC60870-5-10x error code [▶ 658]

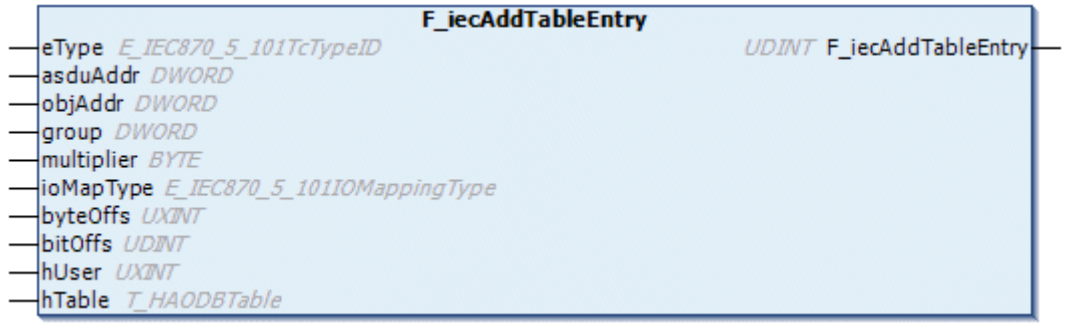
Sample in ST

See description of the F_iecAddTableEntry [▶ 555]function.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.38 F_iecAddTableEntry



The function F_iecAddTableEntry configures the application objects (single points, double points, measured values...) in the application database as hash table entries. The function automatically searches for a free, unused table element (array element) and sets its configuration parameters.

Syntax

```
FUNCTION F_iecAddTableEntry: UDINT
VAR_INPUT
    eType       : E_IEC870_5_101TcTypeID := ASDU_TYPEUNDEF;
    asduAddr    : DWORD := 0;
    objAddr     : DWORD := 0;
```

```
    group       : DWORD := 0;
    multiplier  : BYTE := 0;
    ioMapType   : E_IEC870_5_101IOMappingType := MAP_AREA_NONE;
    byteOffs    : UXINT := 0;
    bitOffs     : UDINT := 0;
    hUser       : UXINT := 0;
END_VAR
VAR_IN_OUT
    hTable      : T_HAODBTable;
ENd_VAR
```

Inputs

eType: Application object type, ASDU identifier [▶ 601] (e.g.: M_SP_NA_1 for single-point or M_DP_NA_1 for double point).

asduAddr: Common ASDU address.

objAddr: Object address, freely selectable.

group: Object group configuration flags. A description of all group flags [▶ 655] can be found here. The flags can be combined with an OR operation. Not all combinations are meaningful!

multiplier: Basic cycle time multiplier for cyclic/periodic data transfer. 0 = deactivated. The basic cycle time can be configured via the *tPerCyclicBase* parameter in the system parameters [▶ 588].

ioMapType: TwinCAT PLC process data area [▶ 608]. This parameter defines the mapping for the TwinCAT PLC and IEC application object process data.

byteOffs: TwinCAT PLC process data byte offset (x86 platform => 32 bit, x64 platform => 64 bit).

bitOffs: TwinCAT PLC process data bit offset.

hUser: Freely definable value, also pointer (x86 platform => 32 bit, x64 platform => 64 bit). This value is stored in the configuration data of the application object.

Inputs/outputs

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

Return value

Return parameter	Description
0	No error.
<> 0	Error: IEC60870-5-10x error code [▶ 658]

Sample in ST

In the following sample, some data points are added to the application database as hash table entries:

eType	asduAddr	objAddr	group	multiplier	ioMap-Type	byteOffs	bitOffs	hUser
M_SP_NA_1	11	100	IEC870_GRP_INRO1	0	MAP_AREA_MEMORY	100	0	16#00BECF01
M_SP_TB_1	11	101	IEC870_GRP_INRO1	0	MAP_AREA_MEMORY	101	0	16#00BECF11
M_DP_NA_1	11	200	IEC870_GRP_INROGEN	0	MAP_AREA_DATA	200	0	16#00BECF02
M_IT_NA_1	11	800	IEC870_GRP_REQCOGEN	0	MAP_AREA_MEMORY	800	0	16#00BECF03

```
VAR_GLOBAL CONSTANT
    MAX_TABLE_IDX   : INT := 49;
END_VAR
```

```
PROGRAM P_iecAddTableEntry
VAR_IN_OUT
    hTable  : T_HAODBTable;
    AODB    : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    bAdd    : BOOL := TRUE;
    nError  : UDINT;
END_VAR
```

```
IF bAdd THEN
    bAdd := FALSE;

    IF ( nError := F_iecCreateTableHnd( ADR( AODB ), SIZEOF( AODB ), hTable )) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecCreateTableHnd() error: %s',
                DWORD_TO_HEXSTR( nError, 8, FALSE ) );
        RETURN;
    END_IF

    IF ( nError := F_iecAddTableEntry( eType := M_SP_NA_1,
                    asduAddr    := 11,
                    objAddr     := 100,
                    group       := IEC870_GRP_INRO1,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 100,
                    bitOffs     := 0,
                    hUser       := 16#00BECF01,
                    hTable      := hTable ) ) <> 0 THEN

        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecAddTableEntry() error: %s',
                DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecAddTableEntry( eType := M_SP_TB_1,
                    asduAddr    := 11,
                    objAddr     := 101,
                    group       := IEC870_GRP_INRO1,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 101,
                    bitOffs     := 0,
                    hUser       := 16#00BECF11,
                    hTable      := hTable ) ) <> 0 THEN

        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecAddTableEntry() error: %s',
                DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecAddTableEntry( eType := M_DP_NA_1,
                    asduAddr    := 11,
                    objAddr     := 200,
                    group       := IEC870_GRP_INROGEN,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_DATA,
                    byteOffs    := 200,
```

```
                    bitOffs     := 0,
                    hUser       := 16#00BECF02,
                    hTable  := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecAddTableEntry() error: %s',
                DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF

    IF ( nError := F_iecAddTableEntry( eType := M_IT_NA_1,
                    asduAddr    := 11,
                    objAddr     := 800,
                    group       := IEC870_GRP_REQCOGEN,
                    multiplier  := 0,
                    ioMapType   := MAP_AREA_MEMORY,
                    byteOffs    := 800,
                    bitOffs     := 0,
                    hUser       := 16#00BECF03,
                    hTable       := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecAddTableEntry() error: %s',
                DWORD_TO_HEXSTR( nError, 8, FALSE ) );
    END_IF
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.39 F_iecGetPosOfTableEntry



The function F_iecGetPosOfTableEntry supplies the table index (array index) for a hash table entry that matches the lookup key. The first array element has position number one (non-zero based array position).

Syntax

```
FUNCTION F_iecGetPosOfTableEntry: UDINT
VAR_INPUT
    key    : ST_IEC870_5_101HashTableKey;
END_VAR
VAR_IN_OUT
    hTable : T_HAODBTable;
ENd_VAR
```

Inputs

key: Lookup key [▶ 599].

Inputs/outputs

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

Return value

Return parameter	Description
0	No table entry matching the key was found.
<> 0	No error. The return parameter supplies the required table index (non-zero-based array position).

Sample in ST

The system searches for the linear table index for three data points. The associated data points have already been configured as hash table entries. See the description of the function: F_iecAddTableEntry [▶ 555].

eType	objAddr	asduAddr	group
M_SP_NA_1	100	11	IEC870_GRP_INRO1
M_SP_TB_1	101	11	IEC870_GRP_INRO1
M_DP_NA_1	200	11	IEC870_GRP_INROGEN
M_IT_NA_1	800	11	IEC870_GRP_REQCOGEN

```
VAR_GLOBAL CONSTANT
    MAX_TABLE_IDX : INT := 49;
END_VAR
```

```
VAR_IN_OUT
    hTable  : T_HAODBTable;
    AODB    : ARRAY[1..MAX_TABLE_IDX] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

```
VAR
    bGet : BOOL := TRUE;
    key1 : ST_IEC870_5_101HashTableKey := ( eType := M_SP_NA_1, asduAddr := 11, objAddr := 100, grou
p := IEC870_GRP_INRO1, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key2 : ST_IEC870_5_101HashTableKey := ( eType := M_SP_TB_1, asduAddr := 11, objAddr := 101, grou
p := IEC870_GRP_INRO1, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key3 : ST_IEC870_5_101HashTableKey := ( eType := M_DP_NA_1, asduAddr := 11, objAddr := 200, grou
p := IEC870_GRP_INROGEN, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key4 : ST_IEC870_5_101HashTableKey := ( eType := M_IT_NA_1, asduAddr := 11, objAddr := 800, grou
p := IEC870_GRP_REQCOGEN, lookup := IEC870_LOOKUP_KEY_ALL_ON );

    nPosition : UDINT;
    hUser1  : UXINT;
    hUser2  : UXINT;
    hUser3  : UXINT;
    hUser4  : UXINT;
END_VAR
```

```
IF bGet    THEN
    bGet := FALSE;

    IF ( nPosition := F_iecGetPosOfTableEntry( key := key1, hTable := hTable ) ) = 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecGetPosOfTableEntry(key1) failed
! %s', '' );
    ELSE
        hUser1 := AODB[nPosition].aObj.cfg.hUser;
    END_IF

    IF ( nPosition := F_iecGetPosOfTableEntry( key := key2, hTable := hTable ) ) = 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecGetPosOfTableEntry(key2) failed
! %s', '' );
    ELSE
        hUser2 := AODB[nPosition].aObj.cfg.hUser;
    END_IF

    IF ( nPosition := F_iecGetPosOfTableEntry( key := key3, hTable := hTable ) ) = 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecGetPosOfTableEntry(key3) failed
! %s', '' );
    ELSE
        hUser3 := AODB[nPosition].aObj.cfg.hUser;
    END_IF

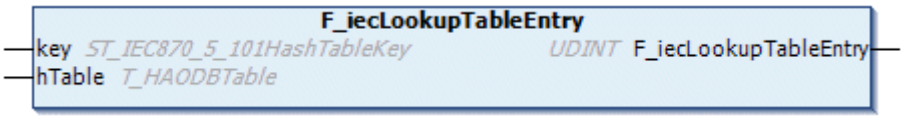
    IF ( nPosition := F_iecGetPosOfTableEntry( key := key4, hTable := hTable ) ) = 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecGetPosOfTableEntry(key4) failed
! %s', '' );
    ELSE
```

```
        hUser4 := AODB[nPosition].aObj.cfg.hUser;
    END_IF
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.40 F_iecLookupTableEntry



The function F_iecLookupTableEntry checks whether a table entry matching the key already exists.

Syntax

```
FUNCTION F_iecLookupTableEntry: UDINT
VAR_INPUT
    key    : ST_IEC870_5_101HashTableKey;
END_VAR
VAR_IN_OUT
    hTable : T_HAODBTable;
ENd_VAR
```

Inputs

key: Lookup key [▶ 599].

Inputs/outputs

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

Return value

Return parameter	Description
0	No error. A table entry matching the key exists.
<> 0	No table entry was found. Error: IEC60870-5-10x error code [▶ 658]

Sample in ST

The existence of data points in the application database is checked. The associated data points have already been configured as hash table entries. See the description of the function: F_iecAddTableEntry [▶ 555].

eType	objAddr	asduAddr	group
M_SP_NA_1	100	11	IEC870_GRP_INRO1
M_SP_TB_1	101	11	IEC870_GRP_INRO1
M_DP_NA_1	200	11	IEC870_GRP_INROGEN
M_IT_NA_1	800	11	IEC870_GRP_REQCOGEN

```
PROGRAM P_iecLookupTableEntry
VAR_IN_OUT
    hTable  : T_HAODBTable;
END_VAR
```

```
VAR
    bLookup : BOOL := TRUE;
    key1    : ST_IEC870_5_101HashTableKey := ( eType := M_SP_NA_1, asduAddr := 11, objAddr := 100, g
roup := IEC870_GRP_INRO1, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key2 : ST_IEC870_5_101HashTableKey := ( eType := M_SP_TB_1, asduAddr := 11, objAddr := 101, grou
p := IEC870_GRP_INRO1, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key3 : ST_IEC870_5_101HashTableKey := ( eType := M_DP_NA_1, asduAddr := 11, objAddr := 200, grou
p := IEC870_GRP_INROGEN, lookup := IEC870_LOOKUP_KEY_ALL_ON );
    key4 : ST_IEC870_5_101HashTableKey := ( eType := M_IT_NA_1, asduAddr := 11, objAddr := 800, grou
p := IEC870_GRP_REQCOGEN, lookup := IEC870_LOOKUP_KEY_ALL_ON );

    nError  : UDINT;
END_VAR
```

```
IF bLookup THEN
    bLookup := FALSE;

    IF ( nError := F_iecLookupTableEntry( key := key1, hTable := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecLookupTableEntry(key1) error: %
s',
                   DWORD_TO_HEXSTR( nError, 8, FALSE) );
    END_IF

    IF ( nError := F_iecLookupTableEntry( key := key2, hTable := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecLookupTableEntry(key2) error: %
s',
                   DWORD_TO_HEXSTR( nError, 8, FALSE) );
    END_IF

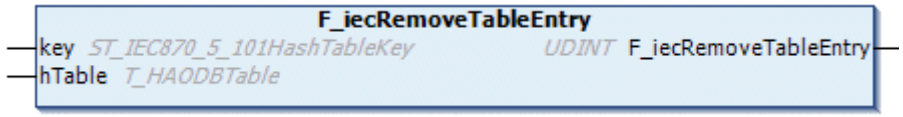
    IF ( nError := F_iecLookupTableEntry( key := key3, hTable := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecLookupTableEntry(key3) error: %
s',
                   DWORD_TO_HEXSTR( nError, 8, FALSE) );
    END_IF

    IF ( nError := F_iecLookupTableEntry( key := key4, hTable := hTable ) ) <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'F_iecLookupTableEntry(key4) error: %
s',
                   DWORD_TO_HEXSTR( nError, 8, FALSE) );
    END_IF
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.41 F_iecRemoveTableEntry



The function F_iecRemoveTableEntry removes a hash table entry matching the key.

Syntax

```
FUNCTION F_iecRemoveTableEntry: UDINT
VAR_INPUT
    key     : ST_IEC870_5_101HashTableKey;
END_VAR
VAR_IN_OUT
    hTable  : T_HAODBTable;
ENd_VAR
```

Inputs

key: Lookup key [▶ 599].

Inputs/outputs

hTable: Application object database handle [▶ 633] (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

Return value

Return parameter	Description
0	No error, the table entry was successfully removed.
<> 0	Error: IEC60870-5-10x error code [▶ 658]

Sample in ST

Three hash table entries are removed from the application database. The associated data points have already been configured as hash table entries. See the description of the function: F_iecAddTableEntry [▶ 555].

eType	objAddr	asduAddr	group
M_SP_NA_1	100	11	IEC870_GRP_INRO1
M_DP_NA_1	200	11	IEC870_GRP_INROGEN
M_IT_NA_1	800	11	IEC870_GRP_REQCOGEN

```
PROGRAM P_RemoveEntry
VAR_IN_OUT
    hTable  : T_HAODBTable;
END_VAR
```

```
VAR
    key     : ST_IEC870_5_101HashTableKey;
    bRemove : BOOL;
    nError  : UDINT;
END_VAR
```

```
IF bRemove THEN
    bRemove      := FALSE;

    key.eType    := M_SP_NA_1;
    key.asduAddr := 11;
    key.objAddr  := 100;
    key.group    := IEC870_GRP_INRO1;
    key.lookup   := IEC870_LOOKUP_KEY_ALL_ON;
    nError       := F_iecRemoveTableEntry( key := key, hTable := hTable );
```

```
    IF nError <> 0 THEN
        RETURN;
    END_IF

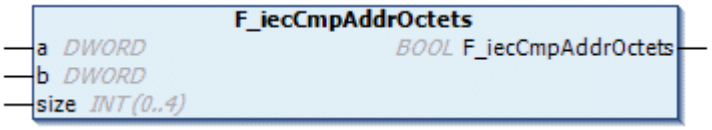
    key.eType    := M_DP_NA_1;
    key.asduAddr := 11;
    key.objAddr  := 200;
    key.group    := IEC870_GRP_INROGEN;
    key.lookup   := IEC870_LOOKUP_KEY_ALL_ON;
    nError       := F_iecRemoveTableEntry( key := key, hTable := hTable );
    IF nError <> 0 THEN
        RETURN;
    END_IF

    key.eType    := M_IT_NA_1;
    key.asduAddr := 11;
    key.objAddr  := 800;
    key.group    := IEC870_GRP_REQCOGEN;
    key.lookup   := IEC870_LOOKUP_KEY_ALL_ON;
    nError       := F_iecRemoveTableEntry( key := key, hTable := hTable );
    IF nError <> 0 THEN
        RETURN;
    END_IF
END_IF
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.42 F_iecCmpAddrOctets



This function compares two addresses (e.g. the connection address, object address or the common ASDU address).

Syntax

```
FUNCTION F_iecCmpAddrOctets: BOOL
VAR_INPUT
    a    : DWORD;
    b    : DWORD;
    size : INT(0..4);
END_VAR
```

Inputs

a: First address.

b: Second address.

size: Byte size of the address (0..3, 4 is reserved).

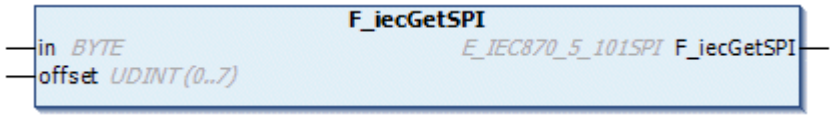
Return value

Return parameter	Meaning
TRUE	(a and b are identical) or (b is an address to all (broadcast)) or (size is 0),
FALSE	All other cases.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.43 F_iecGetSPI



This help function can be used to mask the single point information [▶ 628] of a byte variable. In the TwinCAT PLC the single point information occupies 1 bit of process data. In a byte variable, the information of up to 8 single points can therefore be mapped.

Syntax

```
FUNCTION F_iecGetSPI: E_IEC870_5_101SPI
VAR_INPUT
    in     : BYTE;
    offset : UDINT(0..7);
END_VAR
```

Inputs

in: Byte variable, from which the single point information is to be masked.

offset: Bit offset of the single point.

Sample in ST

The information of the four single points is masked.

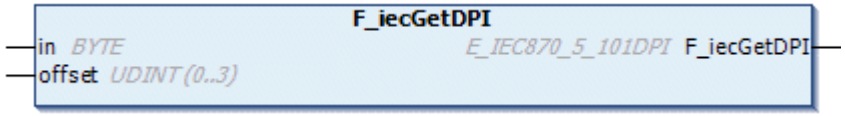
```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
    eSPI          : E_IEC870_5_101SPI;
ENd_VAR
```

```
eSPI := F_iecGetSPI( memarea[0], 0 );
eSPI := F_iecGetSPI( memarea[0], 1 );
eSPI := F_iecGetSPI( memarea[0], 2 );
eSPI := F_iecGetSPI( memarea[0], 3 );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.44 F_iecGetDPI



This help function can be used to mask the double point information [▶ 629] of a byte variable. In the TwinCAT PLC the double point information occupies 2 bit of process data. In a byte variable, the information of up to 4 double points can therefore be mapped.

Syntax

```
FUNCTION F_iecGetDPI: E_IEC870_5_101DPI
VAR_INPUT
    in     : BYTE;
    offset : UDINT(0..3);
END_VAR
```

Inputs

in: Variable, from which the double point information is to be masked.

offset: Bit offset of a double point information.

Sample in ST

The information of the four double points is masked.

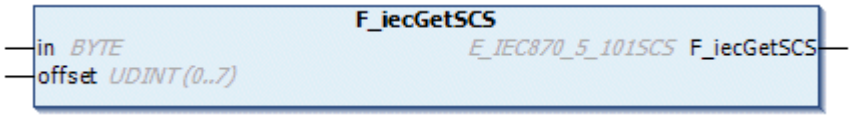
```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
    eDPI          : E_IEC870_5_101DPI;
ENd_VAR
```

```
eDPI := F_iecGetDPI( memarea[0], 0 );
eDPI := F_iecGetDPI( memarea[0], 1 );
eDPI := F_iecGetDPI( memarea[0], 2 );
eDPI := F_iecGetDPI( memarea[0], 3 );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.45 F_iecGetSCS



This help function can be used to mask the single command state [▶ 622] in a byte variable. In the TwinCAT PLC the single command state occupies one bit of process data. In a byte variable, the state of up to 8 single commands can therefore be mapped.

Syntax

```
FUNCTION F_iecGetSCS: E_IEC870_5_101SCS
VAR_INPUT
    in     : BYTE;
    offset : UDINT(0..7);
END_VAR
```

Inputs

in: Byte variable, from which the single command state is to be extracted.

offset: Bit offset of the single command.

Sample in ST

The state of the four single commands is masked.

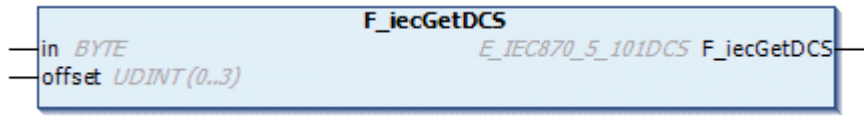
```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
    eSCS          : E_IEC870_5_101SCS;
ENd_VAR
```

```
eSCS := F_iecGetSCS( memarea[0], 0 );
eSCS := F_iecGetSCS( memarea[0], 1 );
eSCS := F_iecGetSCS( memarea[0], 2 );
eSCS := F_iecGetSCS( memarea[0], 3 );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.46 F_iecGetDCS



This help function can be used to mask the double command state [▶ 622] in a byte variable. In the TwinCAT PLC the double command state occupies 2 bits of process data. In a byte variable, the state of up to 4 double commands can therefore be mapped.

Syntax

```
FUNCTION F_iecGetDCS: E_IEC870_5_101DCS
VAR_INPUT
    in     : BYTE;
    offset : UDINT(0..3);
END_VAR
```

Inputs

in: Variable, from which the double command state is to be masked.

offset: Bit offset of the double command state.

Sample in ST

The state of the four double commands is masked.

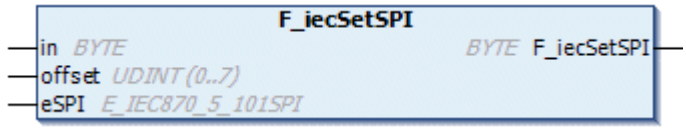
```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
    eDCS          : E_IEC870_5_101DCS;
ENd_VAR
```

```
eDCS := F_iecGetDCS( memarea[0], 0 );
eDCS := F_iecGetDCS( memarea[0], 1 );
eDCS := F_iecGetDCS( memarea[0], 2 );
eDCS := F_iecGetDCS( memarea[0], 3 );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.47 F_iecSetSPI



This help function copies single point information [▶ 628] to a byte variable. In the TwinCAT PLC the single point information occupies 1 bit of process data. In a byte variable, the information of up to 8 single points can therefore be mapped.

Syntax

```
FUNCTION F_iecSetSPI: BYTE
VAR_INPUT
    in      : BYTE;
    offset  : UDINT(0..7);
    eSPI    : E_IEC870_5_101SPI;
END_VAR
```

Inputs

in: Target variable into which the single point information is to be copied.

offset: Bit offset of the information of the single point.

eSPI: The new value of the information of the single point.

Sample in ST

The information of the four single points is set to ON.

```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
END_VAR
```

```
memarea[0] := F_iecSetSPI( memarea[0], 0, eIEC870_SPI_ON );
memarea[0] := F_iecSetSPI( memarea[0], 1, eIEC870_SPI_ON );
memarea[0] := F_iecSetSPI( memarea[0], 2, eIEC870_SPI_ON );
memarea[0] := F_iecSetSPI( memarea[0], 3, eIEC870_SPI_ON );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.48 F_iecSetDPI



This help function copies double point information [▶ 629] to a byte variable. In the TwinCAT PLC the double point information occupies 2 bit of process data. In a byte variable, the information of up to 4 double points can therefore be mapped.

Syntax

```
FUNCTION F_iecSetDPI: BYTE
VAR_INPUT
    in      : BYTE;
    offset  : UDINT(0..3);
    eDPI    : E_IEC870_5_101DPI;
END_VAR
```

Inputs

in: Target variable into which the double point information is to be copied.

offset: Bit offset of the double point information.

eDPI: The new value of the double point information.

Sample in ST

The information of the four double points is set to ON.

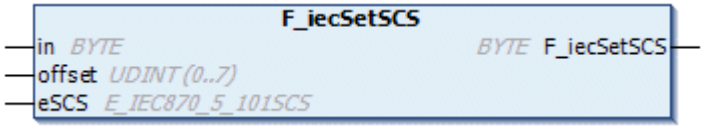
```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
END_VAR
```

```
memarea[0] := F_iecSetDPI( memarea[0], 0, eIEC870_DPI_ON );
memarea[0] := F_iecSetDPI( memarea[0], 1, eIEC870_DPI_ON );
memarea[0] := F_iecSetDPI( memarea[0], 2, eIEC870_DPI_ON );
memarea[0] := F_iecSetDPI( memarea[0], 3, eIEC870_DPI_ON );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.49 F_iecSetSCS



This help function copies the single command state [▶ 622] to a byte variable. In the TwinCAT PLC the single command state occupies one bit of process data. In a byte variable, the state of up to 8 single commands can therefore be mapped.

Syntax

```
FUNCTION F_iecSetSCS: BYTE
VAR_INPUT
    in      : BYTE;
    offset  : UDINT(0..7);
    eSCS    : E_IEC870_5_101SCS;
END_VAR
```

Inputs

in: Target variable into which the single command state is to be copied.

offset: Bit offset of the single command state.

eSCS: The new value of the single command state.

Sample in ST

The state of the four single commands is set to ON.

```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
END_VAR
```

```
memarea[0] := F_iecSetSCS( memarea[0], 0, eIEC870_SCS_ON );
memarea[0] := F_iecSetSCS( memarea[0], 1, eIEC870_SCS_ON );
memarea[0] := F_iecSetSCS( memarea[0], 2, eIEC870_SCS_ON );
memarea[0] := F_iecSetSCS( memarea[0], 3, eIEC870_SCS_ON );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.50 F_iecSetDCS



This help function copies the double command state [▶ 622] to a byte variable. In the TwinCAT PLC the double command state occupies 2 bits of process data. In a byte variable, the state of up to 4 double commands can therefore be mapped.

Syntax

```
FUNCTION F_iecSetDCS: BYTE
VAR_INPUT
    in      : BYTE;
    offset  : UDINT(0..3);
    eDCS    : E_IEC870_5_101DCS;
END_VAR
```

Inputs

in: Target variable into which the double command state is to be copied.

offset: Bit offset of the double command state.

eDCS: The new value of the double command [▶ 622] state.

Sample in ST

The state of the four double commands is set to ON.

```
PROGRAM MAIN
VAR
    memarea AT%M* : ARRAY[0..10] OF BYTE;
END_VAR
```

```
memarea[0] := F_iecSetDCS( memarea[0], 0, eIEC870_DCS_ON );
memarea[0] := F_iecSetDCS( memarea[0], 1, eIEC870_DCS_ON );
memarea[0] := F_iecSetDCS( memarea[0], 2, eIEC870_DCS_ON );
memarea[0] := F_iecSetDCS( memarea[0], 3, eIEC870_DCS_ON );
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.51 F_iecChangeLinkLayerMode



With this function the default behavior of each individual ASDU frame at the Link-Layer level in the primary station can be set or changed. The ASDU frame to be sent can be marked as a SEND/NO REPLY frame (function 4) or as a broadcast frame. The frames are marked before depositing the frame in the TX buffer. I.e. before calling the FB_IEC870_5_101TBufferCtrl [▶ 506].TxAddObj action.

If you do not use the function, then all frames in the primary station are sent as SEND/CONFIRM frames (function 3). In this case the configured station address is used as the link address.

- In unbalanced mode the use of this function has no meaning in the substation. The substation never acts as a primary station in this mode.
- The function has no meaning when using the IEC 6087-5-104 protocol.
- This functionality is supported only when using the 'low level' interface.

Syntax

```
FUNCTION F_iecChangeLinkLayerMode: BOOL
VAR_INPUT
    sendNoReplay   : BOOL;
    globalLinkAddr : BOOL;
END_VAR
VAR_IN_OUT
    asdu           : ST_IEC870_5_101AOGen;
END_VAR
```

Inputs

sendNoReplay: This parameter specifies whether the Link-Layer function SEND/NO REPLY (TRUE = function 4) or SEND/CONFIRM (FALSE = function 3) is to be used when sending the frame;

globalLinkAddr: This parameter specifies whether a global (broadcast) address is to be used instead of the configured station link address when sending the frame. If the value is TRUE, 16#FF or 16#FFFF (one octet size, two octets size link address) is used as the link address in the sent frame;

Inputs/outputs

asdu: The data unit [▶ 585] to be sent as VAR_IN_OUT variable;

Return value

Return parameter	Description
FALSE	Function failed.
TRUE	No error.

Sample 1 in ST (extract)

The spontaneous data of a bit string are to be sent to the central station with the aid of the SEND/NO REPLY function (balanced mode).

```
...

(* Send spontanous bitstring data *)
IF ( txQDS <> BITSTRING_QUALITY_100 ) OR ( txBSI <> BITSTRING_100 ) THEN

    txBSI                   := BITSTRING_100;
    txQDS                   := BITSTRING_QUALITY_100;(* Get quality *)
    txTT                    := SYSTEMTIME_TO_CP56Time2a( fbRTC.CDT, TRUE );
(* Get current time stamp *)(* create asdu *)
    txAsdu.ident.eType      := M_BO_TB_1;  (* Bit string with time tag *)
    txAsdu.ident.bSQ        := FALSE;
    txAsdu.ident.nObj       := 1;
    txAsdu.ident.eCOT       := eIEC870_COT_SPONTAN;
    txAsdu.ident.nORG       := sysPara.nOrg;(* Set originator address *)
    txAsdu.ident.bPN        := FALSE;
    txAsdu.ident.bT         := FALSE;
    txAsdu.ident.eClass     := eIEC870_Class_1;(* Put to the high priority tx buffer *)
    txAsdu.ident.asduAddr   := sysPara.asduAddr;(* Set common asdu address *)
    txAsdu.info.objAddr     := 100;(* Set information object address *)
    F_iecResetStream( 0, txAsdu.info.stream ); (* clear previous data (this sets the stream length =
 0 *)
    F_iecCopyBufferToStream( ADR( txBSI ), SIZEOF( txBSI ), txAsdu.info.stream ); (* put BSI to stre
am *)
    F_iecCopyBufferToStream( ADR( txQDS ), SIZEOF( txQDS ), txAsdu.info.stream ); (* put QDS to stre
am *)
```

```
    F_iecCopyBufferToStream( ADR( txTT ), 7 (*SIZEOF( txTT )-1*), txAsdu.info.stream ); (* put time
tag to stream *)F_iecChangeLinkLayerMode( TRUE, FALSE, txAsdu );

    fbBuffer.TxAddObj( putObj := txAsdu, buffer := buffer ); (* put asdu to the TX fifo *)F_iecChang
eLinkLayerMode( FALSE, FALSE, txAsdu );

    IF fbBuffer.bOk THEN
        fbLog( put := CONCAT( '<=', IEC101ASDU_TO_STRING(txAsdu) ) );
    ELSE(* Report send buffer overflow error *)
        fbLog( put := 'TX buffer overflow (spontanous bitstring data)!' );
    END_IF
END_IF

...
```

The txAsdu variable is used to send further data points. The default configuration of the txAsdu frame is produced by a further F_iecChangeLinkLayerMode (FALSE, FALSE,…) function call. Other ASDUs are not to be sent as SEND/NO REPLY telegrams.

Sample 2 in ST (extract)

A single command is to be sent to the substation with the aid of the SEND/NO REPLY function (unbalanced mode).

```
...

(* Send one single command *)
IF SND_SCS_2100 THEN
    SND_SCS_2100 := FALSE; (* Reset flag *)

    txAsdu.ident.eType      := C_SC_NA_1; (* Single command *)
    txAsdu.ident.bSQ        := FALSE;
    txAsdu.ident.nObj       := 1;
    txAsdu.ident.eCOT       := eIEC870_COT_ACT; (* Command activation *)
    txAsdu.ident.nORG       := sysPara.nOrg;(* Set originator address *)
    txAsdu.ident.bPN        := FALSE;
    txAsdu.ident.bT         := FALSE;
    txAsdu.ident.eClass     := eIEC870_Class_1;(* Put to the high priority tx buffer *)
    txAsdu.ident.asduAddr   := sysPara.asduAddr;(* Set common asdu address *)
    txAsdu.info.objAddr     := 2100;(* Set information object address *)
    tmpByte                 := INT_TO_BYTE(SCS_2100);(* Set single command state *)
    tmpByte.7               := 0;(* Set select/execute bit *)

    F_iecResetStream( 0, txAsdu.info.stream ); (* Clear previous data (this sets the stream length =
 0 *)
    F_iecCopyBufferToStream( ADR( tmpByte ), SIZEOF( tmpByte ), txAsdu.info.stream ); (* put QCC to
stream *)     F_iecChangeLinkLayerMode( TRUE, FALSE, txAsdu );

    fbBuffer.TxAddObj( putObj := txAsdu, buffer := buffer ); (* put asdu to the TX fifo *)
     F_iecChangeLinkLayerMode( FALSE, FALSE, txAsdu );

    IF fbBuffer.bOk THEN
        timerCON( IN := FALSE );(* Reset timer *)
        timerTERM( IN := FALSE );(* Reset timer *)
        fbLog( put := CONCAT( '<=', IEC101ASDU_TO_STRING(txAsdu) ) );
        state := 80;(* Wait for command confirmation *)
    ELSE(* Report send buffer overflow error *)
        fbLog( put := 'TX buffer overflow (single command)!' );
        state := 1;
    END_IF

END_IF

...
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.52 F_iecApdu101ToAsduLen



The function calculates the maximum available ASDU octet length for the IEC 60870-5-101 protocol by means of the configured APDU telegram length and the address field length of the link layer. The maximum available ASDU length is required for the configuration of the ST_IEC870_5_101TBuffer variable, for example. This data structure (TX/RX data buffer) is used for the data exchange via the IEC60870-5-101 low-level interface.

Syntax

```
FUNCTION F_iecApdu101ToAsduLen: BYTE
VAR_INPUT
    FRAMELength    : BYTE;
    eLinkAddrSize : E_IEC870_5_101LinkAddrSize;
END_VAR
```

Inputs

FRAMELength: The max. available APDU telegram length (see compatibility list).

eLinkAddrSize: Address field length [▶ 609] of the link layer.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.53 F_iecApdu102ToAsduLen



The function calculates the maximum available ASDU octet length for the IEC 60870-5-102 protocol by means of the configured APDU telegram length and the address field length of the link layer. The maximum available ASDU length is required for the configuration of the ST_IEC870_5_102TBuffer variable, for example. This data structure (TX/RX data buffer) is used for the data exchange via the IEC60870-5-102 low-level interface.

Syntax

```
FUNCTION F_iecApdu102ToAsduLen: BYTE
VAR_INPUT
    FRAMELength   : BYTE;
    eLinkAddrSize : E_IEC870_5_101LinkAddrSize;
END_VAR
```

Inputs

FRAMELength: The max. available APDU telegram length (see compatibility list).

eLinkAddrSize: Address field length [▶ 609] of the link layer.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.54 F_iecApdu103ToAsduLen



The function calculates the maximum available ASDU octet length for the IEC 60870-5-103 protocol by means of the configured APDU telegram length and the address field length of the link layer. The maximum available ASDU length is required for the configuration of the ST_IEC870_5_103TBuffer variable, for example. This data structure (TX/RX data buffer) is used for the data exchange via the IEC60870-5-103 low-level interface.

Syntax

```
FUNCTION F_iecApdu103ToAsduLen: BYTE
VAR_INPUT
    FRAMELength   : BYTE;
    eLinkAddrSize : E_IEC870_5_101LinkAddrSize;
END_VAR
```

Inputs

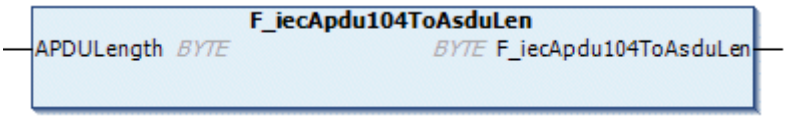
FRAMELength: The max. available APDU telegram length (see compatibility list).

eLinkAddrSize: Address field length [▶ 609] of the link layer.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.55 F_iecApdu104ToAsduLen



The function calculates the maximum available ASDU octet length for the IEC 60870-5-104 protocol by means of the configured APDU length. The maximum available ASDU length is required for the configuration of the ST_IEC870_5_101TBuffer variable, for example. This data structure (TX/RX data buffer) is used for the data exchange via the IEC60870-5-104 low-level interface.

Syntax

```
FUNCTION F_iecApdu104ToAsduLen: BYTE
VAR_INPUT
    APDULength : BYTE;
END_VAR
```

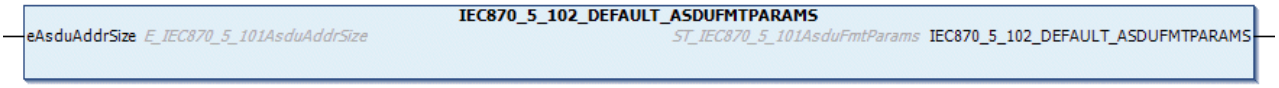
Inputs

APDULength: The available APDU length (see compatibility list).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.56 IEC870_5_102_DEFAULT_ASDUFMTPARAMS



This function sets the frame format parameters [▶ 591] specified by the IEC 60870-5-102 standard:

- Byte length of the cause of transfer;
- Byte length of the common ASDU address;
- Byte length the information object address;

Syntax

```
FUNCTION IEC870_5_102_DEFAULT_ASDUFMTPARAMS: ST_IEC870_5_101AsduFmtParams
VAR_INPUT
    eAsduAddrSize : E_IEC870_5_101AsduAddrSize;
END_VAR
```

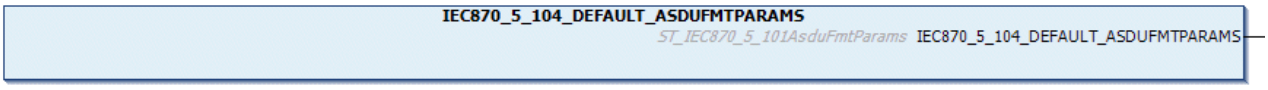
Inputs

eAsduAddrSize: Octet length of the common ASDU address [▶ 608].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.57 IEC870_5_104_DEFAULT_ASDUFMTPARAMS



This function sets the frame format parameters [▶ 591] specified by the IEC 60870-5-104 standard to the default values:

- Cause of transfer: 2 bytes;
- Common ASDU address: 2 bytes;
- Information object address: 3 bytes.

Syntax

```
FUNCTION IEC870_5_104_DEFAULT_ASDUFMTPARAMS: ST_IEC870_5_101AsduFmtParams
VAR_INPUT
(* keine Parameter *)
END_VAR
```

Inputs

None.

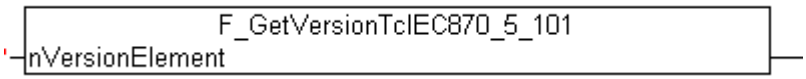
Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58 Obsolete

5.2.58.1 F_GetVersionTcIEC870_5_101

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_101: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

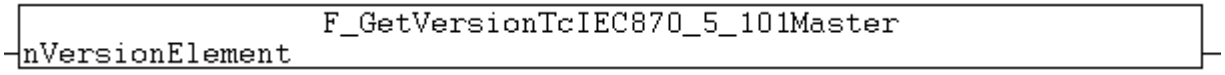
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.2 F_GetVersionTcIEC870_5_101Master

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_101Master: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

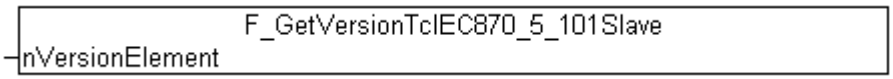
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.3 F_GetVersionTcIEC870_5_101Slave

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_101Slave: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

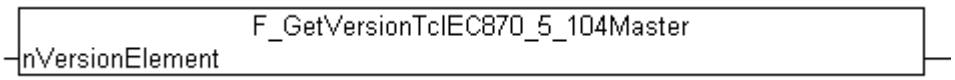
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.4 F_GetVersionTcIEC870_5_104Master

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_104Master: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

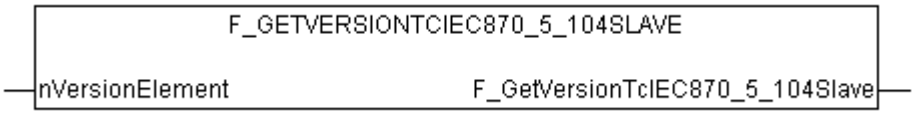
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.5 F_GetVersionTcIEC870_5_104Slave

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_104Slave: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

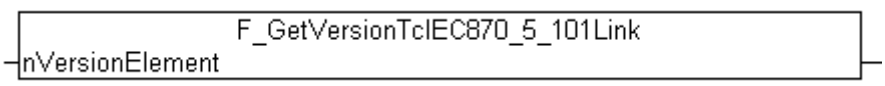
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.6 F_GetVersionTcIEC870_5_101Link

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_101Link: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element that is to be read. Possible parameters:

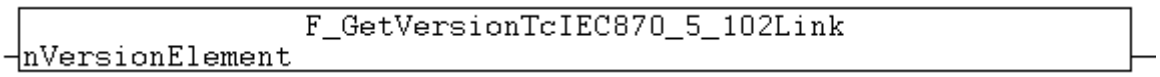
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.7 F_GetVersionTcIEC870_5_102Link

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_102Link: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

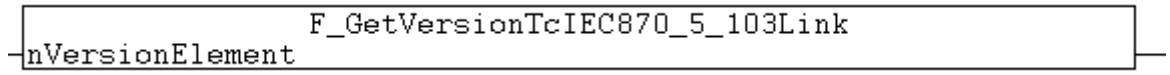
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.8 F_GetVersionTcIEC870_5_103Link

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_103Link: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

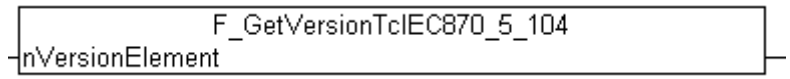
- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.2.58.9 F_GetVersionTcIEC870_5_104

This function is an outdated function, which is only included in the library for compatibility reasons! The current version information of the library is contained in a global constant [▶ 661].



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcIEC870_5_104: UINT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element, that is to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3 Data types

5.3.1 ST_IEC870_5_101ExSystemInterface

Syntax

```
TYPE ST_IEC870_5_101ExSystemInterface :
STRUCT
    device      : ST_IEC870_5_101DeviceInterface;
    service     : ST_IEC870_5_101SystemServices;
    hSOTable    : T_HAODBTABLE;
END_STRUCT
END_TYPE
```

device: Internal communication interface of the IEC device [▶ 591].

service: IEC device services [▶ 583];

hSOTable: System object database handle [▶ 633];

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.2 ST_IEC870_5_101SystemInterface

Syntax

```
TYPE ST_IEC870_5_101SystemInterface :
STRUCT
    device  : ST_IEC870_5_101DeviceInterface;
    service : ST_IEC870_5_101SlaveServices;
END_STRUCT
END_TYPE
```

device: Communication interface [▶ 591] of the IEC device. This variable is used by other PLC functions or function blocks as communication interface with the IEC device (here: substation).

service: IEC device services;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.3 ST_IEC870_5_101SystemServices

This structured variable is only used for internal communication by the station.

Syntax

```
TYPE ST_IEC870_5_101SystemServices
STRUCT
(**)
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.4 ST_IEC870_5_101TBuffer

This data structure (TX/RX data buffer) is used for the data exchange via the IEC60870-5-104/101 low-level interface.

Syntax

```
TYPE ST_IEC870_5_101TBuffer :
STRUCT
    eDbg        : E_IEC870_5_101FifoDbgFlags :=eIEC870_FIFO_DBG_OFF;
    asduFmt     : ST_IEC870_5_101AsduFmtParams;
    asduSize    : BYTE := 0;
    mode        : DWORD := 0;
    dataLink    : ST_IEC870_5_101DataLink;
    bOverwrite  : BOOL := FALSE;
END_STRUCT
END_TYPE
```

eDbg: Configures the debugging output [▶ 616] of the ASDUs.

asduFmt: ASDU format parameters [▶ 591].

asduSize: Maximum byte length of the ASDU.

mode: Reserved; currently not used. This value should be null.

dataLink: The elements of this data structure should not be accessed directly, but only with an instance of the FB_IEC870_5_101TBufferCtrl [▶ 506] function block.

Internally, the TX/RX data buffer uses two transmit FIFOs and one receive FIFO:

1. Class 1 transmit FIFO with (high-priority) data;
2. Class 2 transmit FIFO with (low-priority) data;
3. Receive FIFO (for Class 1 and Class 2 data);

The lower transport function of the library first empty the Class 1 FIFO, then the Class 2 FIFO. The Class 2 data can only be sent if the Class 1 FIFO contains no data to be sent.

Each of the internal FIFOs has a fixed size of 200000 bytes. This should be adequate for most applications, since the number of frames, which can be sent once is limited through the iK and iW protocol parameters. Experience shows that each FIFO can hold approx. 2000 ASDUs with one information element (object) or approx. 600 ASDUs with a sequence of 100 information elements (objects).

If a larger number of transmit or receive frames is to be cached (e.g. >2000), external buffers/FIFOs specified by the PLC programmer can be used. At runtime, the PLC application can then top up the TwinCAT transmit FIFO with its own FIFO entries or empty the TwinCAT receive FIFO, if many frames are received. Another possibility is to use two buffers, for example, and alternately fill/read them and transfer the content to the communication block.

bOverwrite: Activates/deactivates the overwriting of the oldest messages in the send buffer if the max. permissible buffer size is exceeded. This parameter should only be enabled in offline mode (i.e. if the connection was broken) and if offline data storage was additionally enabled via the **bRetainBuffer** parameter in the protocol parameters (ST_IEC870_5_10xProtPara). In online mode (i.e. once the data transfer has started), no older messages should be overwritten (otherwise some intermediate values may be missing).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.5 ST_IEC870_5_101AODBEntry

An IEC application object database entry. The IEC application object database is declared as an array variable of type ST_IEC870_5_101AODBEntry. The member variables of this structured type are not modified directly, but only using the available functions or function blocks. F_iecIniAOEntry [▶ 537] is part of such a function, for example.

Syntax

```
TYPE ST_IEC870_5_101AODBEntry :
STRUCT
    aObj : ST_IEC870_5_101AOEntry;
    ctrl : FB_IEC870_5_101AOCtrl;
END_STRUCT
END_TYPE
```

Sample for a declaration of an application database with 2001 objects:

Sample

```
VAR_GLOBAL
    slave1AODB : ARRAY[0..2000] OF ST_IEC870_5_101AODBEntry;
ENd_VAR
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.6 ST_IEC870_5_101AOGen

Variables of this type represent an ASDU object.

Syntax

```
TYPE ST_IEC870_5_101AOGen:
STRUCT
    head : ST_IEC870_5_101FifoHead :=( source := ( link := 0, addr := 0 ),
                                       target := ( link := 0, addr := 0 ),
                                       ctrl := 0 );

    ident: ST_IEC870_5_101DataUnit_Ident := ( eType      := ASDU_TYPEUNDEF,
                          nObj    := 0,
                          bSQ     := FALSE,
                          bT      := FALSE,
                          bPN     := FALSE,
                          nORG    := 0,
                          asduAddr:= 0,
                          eCOT    := eIEC870_COT_UNUSED,
                          eClass  := eIEC870_Class_None );
 info    : ST_IEC870_5_101AOInfoObj := ( objAddr := 0, stream := ( length := 0 ) );
END_STRUCT
END_TYPE
```

head: Header (reserved).

ident: ID fields of the data unit [▶ 586] (ASDU).

info: Information object/information element data field [▶ 587].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.7 ST_IEC870_5_101AOCfg

ASDU object configuration parameters. These parameters are set during configuration of the data points (F_iecInitAOEntry or F_iecAddTableEntry) and should not be written directly from the PLC application.

Syntax

```
TYPE ST_IEC870_5_101AOCfg:
STRUCT
    group       : DWORD := IEC870_GRP_INROGEN;
    multiplier  : BYTE := 0;
    ioMapType   : E_IEC870_5_101IOMappingType := MAP_AREA_NONE;
```

```
    byteOffs    : UXINT := 0;
    bitOffs     : UDINT := 0;
    hUser       : UXINT := 0;
END_STRUCT
END_TYPE
```

group: Object group configuration flags. A description of all group flags can be found here [▶ 655]. The flags can be combined with an OR operation. Not all combinations are meaningful!

multiplier: Basic cycle time multiplier for cyclic/periodic data transfer. 0 = deactivated. The basic cycle time can be configured via the *tPerCyclicBase* parameter in the system parameters [▶ 588].

ioMapType: TwinCAT PLC process data area [▶ 608]. This parameter defines the mapping for the TwinCAT PLC and IEC application object process data.

byteOffs: TwinCAT PLC process data byte offset (x86 platform => max. 32 bit, x64 platform => max. 64 bit).

bitOffs: TwinCAT PLC process data bit offset.

hUser: Optional user handle. Freely definable value, also pointer (x86 platform => 32 bit, x64 platform => 64 bit).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.8 ST_IEC870_5_101DataUnit_Ident

ID field of the data unit (ASDU).

Syntax

```
TYPE ST_IEC870_5_101DataUnit_Ident:
STRUCT
    eType    : E_IEC870_5_101TcTypeID;
    bSQ      : BOOL;
    nObj     : BYTE;
    bT       : BOOL;
    bPN      : BOOL;
    eCOT     : E_IEC870_5_101COTType;
    nORG     : BYTE;
    asduAddr: DWORD;
    eClass   : E_IEC870_5_101ClassType;
END_STRUCT
END_TYPE
```

eType: IEC 60870-5-101/104 ASDU type ID [▶ 601] := <0..255>.

bSQ: Sequence-flag := <FALSE = single, TRUE = sequence>.

nObj: Number of information objects or information elements := <0..127>.

bT: Test flag (rarely used).

bPN: Positive/negative confirmation := <FALSE = positive, TRUE = negative>.

eCOT: IEC 60870-5-101/104 cause of transfer [▶ 611] (COT) := <0..63>.

nORG: Source address.

asduAddr: Common ASDU address (CAA).

eClass: Priority class [▶ 611] (TX data).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.9 ST_IEC870_5_101AOInfoObj

Information object description.

Syntax

```
TYPE ST_IEC870_5_101AOInfoObj :
STRUCT
    objAddr : DWORD;
    stream  : ST_IEC870_5_101Stream;
END_STRUCT
END_TYPE
```

objAddr : Information object address (IOA).

stream: Information element data buffer [▶ 587].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.10 ST_IEC870_5_101Stream

Variables of this type are used as data buffer (stream). "Raw" data bytes cached in the stream in the order in which they are to be sent later or in which they were received. The null data byte is always sent or received first.

Syntax

```
TYPE ST_IEC870_5_101Stream :
STRUCT
    length  : DWORD := 0;
    data    : ARRAY[0..IEC870_MAX_ASDU_DATA_BYTE] OF BYTE;
END_STRUCT
END_TYPE
```

length: Current number of data bytes in the stream.

data: Data buffer (byte array).

Memory representation of a stream variable with a cached DWORD variables with the value: **16#BECF1234**. Note the interchanged data bytes in the Intel format!

length	data											
4	16#34	16#12	16#CF	16#BE	...	...	...	...	...	...	...	IEC870_MAX_ASDU_DATA_BYTE

Use the functions listed in the table to change the memory content of a stream variable:

Function	Description
F_iecResetStream [▶ 553]	Stream initialization/reset.
F_iecCopyBufferToStream [▶ 544]	Copies data bytes from an external buffer variable to the stream. The memory content of the stream variable is increased.
F_iecCopyBufferToStreamByOffset [▶ 545]	Copies data bytes from an external buffer variable to the stream from a byte offset. The memory content of the stream variable is increased.
F_iecCopyStreamToBuffer [▶ 546]	Copies data bytes from the stream to an external buffer variable. The memory content of the stream variable remains unchanged.
F_iecCopyStreamToBufferByOffset [▶ 548]	Copies data bytes from stream to an external buffer variable from a byte offset. The memory content of the stream variable remains unchanged.
F_iecCopyStreamToStream [▶ 549]	Copies data bytes from one stream to another stream. The memory content of the source variable remains unchanged. The memory content of the target variable is increased.
F_iecMoveStreamToBuffer [▶ 551]	Copies data bytes from the stream to an external buffer variable and then deletes the copied data bytes from the stream. The memory content of the stream variable is reduced.
F_iecMoveStreamToStream [▶ 552]	Copies data bytes from the source stream to the target stream and then deletes the copied data bytes from the source stream. The memory content of the source variable is reduced, the content of the target variable is increased.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.11 ST_IEC870_5_101SystemParams

System parameters of the station.

Syntax

```
TYPE ST_IEC870_5_101SystemParams :
STRUCT
    asduAddr        : DWORD := 11;
    asduAddrRev     : DWORD := 22;

    asduFmt         : ST_IEC870_5_101AsduFmtParams := ( eCOTSize := eIEC870_COT_TwoOctets,
```

```
                                                    eAsduAddrSize := eIEC870_AsduAddr_TwoOctets,
                                                    eObjAddrSize := eIEC870_ObjAddr_ThreeOctets
);
    bEndOfInit      : BOOL := TRUE;

    bSyncTime       : BOOL := TRUE;
    bSyncPCTime     : BOOL := FALSE;

    bUsePCTime      : BOOL := TRUE;
    tSyncToPCTime   : TIME := T#0s;
    sPCTimeNetID    : T_AmsNetID := '';

    bTimeCOT3       : BOOL := FALSE;
    tSyncTimeut     : TIME := T#0m;

    bACTCONRes      : BOOL := TRUE;
    bACTTERMRes     : BOOL := TRUE;

    bPerCyclic      : BOOL := FALSE;
    tPerCyclicBase  : TIME := T#60s;

    bBackScan       : BOOL := FALSE;
    tBackScanCycle  : TIME := T#60s;

    bPerFRZ         : BOOL := FALSE;
    tPerFRZCycle    : TIME := T#60s;
    bSyncPerFRZtoHour : BOOL := FALSE;

    tSelExecTimeout : TIME := T#10s;
    tActConTimeout  : TIME := T#5s;
    tActTermTimeout : TIME := T#30s;
    tDeactConTimeout: TIME := T#5s;
    tReadResTimeout : TIME := T#5s;

    dbgMode         : DWORD := 0;
    orgAddr         : BYTE := 1;

    bOverwrite      : BOOL := FALSE;
    bCBOActTerm     : BOOL := FALSE;
    bAdoptSuccTSC   : BOOL := FALSE;
END_STRUCT
END_TYPE
```

asduAddr: Common ASDU address (CAA) in standard direction. Default: 11.

asduAddrRev: Common ASDU address (CAA) in reverse direction. Default: 22.

asduFmt: ASDU format parameter [▶ 591] (e.g. octet length of the cause of transfer, ASDU address and object address);

bEndOfInit: If TRUE, send M_EI_NA_1 (end of init) after station initialization is completed. A station initialization is performed during a system startup or after a process reset. Default: TRUE.

bSyncTime: If TRUE, activate IEC system time synchronization on receiving the C_CS_NA_1 command. Default: TRUE.

bSyncPCTime: If TRUE, synchronize the IEC system time and the system time of the TwinCAT PC (the Windows system time in the taskbar). Default: FALSE.

bUsePCTime: If TRUE, synchronize the IEC system time with the system time of the PC. Following the initialization of the substation, the IEC system time is first synchronized with the system time of the TwinCAT PC (Windows system time). Default: TRUE.

tSyncToPCTime : controls the cyclic synchronization of the IEC system time with the system time of the TwinCAT PC (in online and offline mode). Implemented in IEC870-5-104 slave library v3.0.3 and higher.

The time is determined by the cycle time in which the synchronization is executed. A value of T#0s deactivates the cyclic synchronization. Default: T#0s.

sPCTimeNetID: TwinCAT network address of the PC whose system time is to be used for the synchronization. If it is to be run on the local computer, an empty string can be entered. Default: Empty string = local PC.

bTimeCOT3: If TRUE, send the system time to the central station with the cause of transfer <*3*> *spontaneous* on the hour. Default: FALSE.

tSyncTimeout: Time synchronization interval timeout monitoring. Implemented in IEC870-5-101/104 slave library v2.0.0 and higher. This parameter is not used in older versions. The timestamps have an IV quality flag (invalid). If the value is <> T#0m (e.g. T#60m), the IV quality flag for all subsequent timestamps is set to invalid, if no time synchronization was performed within one hour. The master can use the IV quality flag of the timestamp as indicator for the quality of the timestamp. Monitoring is not active if tSyncTimeout = T#0m.

bACTCONRes: If TRUE, send ACTCON response.

bACTTERMRes: If TRUE, send ACTTERM response.

bPerCyclic: If TRUE, activate cyclic/periodic data transfer.

tPerCyclicBase: Base time for the cyclic/periodic data transfer.

bBackScan: If TRUE, activate background scan.

tBackScanCycle: Background scan cycle time.

bPerFRZ: If TRUE, activate local freezing/resetting of the counter values.

tPerFRZCycle: Cycle time for local freezing/resetting.

bSyncPerFRZtoHour: Synchronize periodic/cyclic freezing of the counters on the hour. Default: FALSE (deactivated).

tSelExecTimeout: Max. timeout time between the selection command and the execution command (central station only).

tActConTimeout: Max. timeout time for receiving the command activation confirmation (central station only).

tActTermTimeout: Max. timeout time for receiving the command termination (central station only).

tDeactConTimeout: Max. timeout time for receiving the command abortion confirmation (central station only).

tReadResTimeout: Max. timeout time for execution of the read command (C_RD_NA_1) (central station only).

dbgMode: Debug flags:

Value	Description
IEC870_DEBUGMODE_DISABLED	Logging deactivated
IEC870_DEBUGMODE_ASDU	The ASDUs are logged as hexadecimal output in the application log.
IEC870_DEBUGMODE_DEVSTATE	State changes of the station are logged in the application log.
IEC870_DEBUGMODE_LINKLAYER	Link layer frames are logged in the application log IEC870-5-101 slave library: available for all versions; IEC870-5-104 slave library: available from v2.0.0 and higher;
IEC870_DEBUGMODE_LINKERROR	Link layer error messages are logged in the application log.

The flags can be OR-linked in the required combination.

orgAddr: Source address (currently not used).

bOverwrite: Activates/deactivates the overwriting of the oldest messages in the send buffer if the max. permissible buffer size is exceeded. This parameter is active only in the offline mode, (i.e. if the connection has been interrupted) and if offline data storage has additionally been activated via the **bRetainBuffer** parameter in the protocol parameters (ST_IEC870_5_104ProtPara). In online mode (i.e. if data transfer has been started) no older messages are overwritten (otherwise some intermediate values would possibly be missing).

bCBOActTerm : Only for: C_BO_NA_1, C_BO_TA_1, C_BO_TB_1. Used/expected causes of transfer: actTerm, deact, deactCon bit string command. Default: FALSE (do not use).

bAdoptSuccTSC : Only for: C_TS_TA_1: Accept/synchronize received counter value.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.12 ST_IEC870_5_101DeviceInterface

Syntax

```
TYPE ST_IEC870_5_101DeviceInterface :
STRUCT
    asduAddr    : DWORD;
    asduAddrRev : DWORD;
    dataLink    : ST_IEC870_5_101DataLink;
    clock       : ST_IEC870_5_101SystemRTC;
    comp        : ST_IEC870_5_101DeviceCompatibility;
    errors      : FB_IEC870_5_101ErrorFifo;
    status      : DWORD;
END_STRUCT
END_TYPE
```

asduAddr: Common ASDU address in standard direction;

asduAddrRev: Common ASDU address in reverse direction.

dataLink: Internal ASDU transmit/receive FIFOs;

clock: IEC system time object;

comp: Compatibility list with the assignment of the ASDU types to the causes of transfer;

errors: Device error FIFO [▶ 522];

status: Global device status. 1=OK, 0=Not OK;

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.13 ST_IEC870_5_101AsduFmtParams

ASDU frame format parameter.

Syntax

```
TYPE ST_IEC870_5_101AsduFmtParams :
STRUCT
    eCOTSize      : E_IEC870_5_101COTSize      := eIEC870_COT_TwoOctets;
    eAsduAddrSize : E_IEC870_5_101AsduAddrSize := eIEC870_AsduAddr_TwoOctets;
    eObjAddrSize  : E_IEC870_5_101ObjAddrSize  := eIEC870_ObjAddr_ThreeOctets;
END_STRUCT
END_TYPE
```

eCOTSize : Octet size of the cause of transfer [▶ 609] (COT).

eAsduAddrSize : Octet size of the common ASDU address [▶ 608] (CAA).

eObjAddrSize: Octet size of the information object address [▶ 610] (IOA).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.14 ST_IEC870_5_101ErrorFifoEntry

TwinCAT IEC 60870-5-10x device error.

Syntax

```
TYPE ST_IEC870_5_101ErrorFifoEntry :
STRUCT
    nErrId : UDINT;
    eSrcId : E_IEC870_5_101ErrorSourceID;
END_STRUCT
END_TYPE
```

nErrID: Error code (depending on protocol):

- TwinCAT IEC 60870-5-101 error codes [▶ 784];
- TwinCAT IEC 60870-5-102 error codes [▶ 793];
- TwinCAT IEC 60870-5-103 error codes [▶ 799];
- TwinCAT IEC 60870-5-104 error codes [▶ 805];

eSrcID: Source of error [▶ 610]. The source of error provides further information about the cause of the error, the component or the service reporting the error.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.15 ST_IEC870_5_101AcquisitionParams

Configuration parameters for cyclic data acquisition. In most cases, an initialization sequence should be executed once by the central station. This sequence contains some commands which are executed after the communication link was established and/or after receipt of ASDU: M_EI_NA_1 (end of initialization). Such commands include, for example, the clock synchronization command. Then the central station should, for example, continuously send cyclic interrogation commands to the substation. On an IEC 60870-5-104 central station, data exchange is activated after the communication link is established and "STARTDT" (Start Data Transfer) is received. In the case of an IEC 60870-5-101 central station after the Reset Of Link.

From Tc2_IEC60870_5_10x.compiled-library v3.0.7.0, the time of processing the initialization sequence and the cyclic interrogation commands can be configured. The application can choose whether the initialization sequence is to be executed after the STARTDT or after receiving M_EI_NA_1.

The structure component *arrInitSeq* is used to configure the initialization sequence. There you can preconfigure some different commands.

The structural component *eMode* determines when these commands are to be executed. The initialization sequence can be executed, for example, after the connection has been established (after STARTDT or Reset Of Link) or after receipt of ASDU: M_EI_NA_1 (end of initialization).

The cyclic interrogation commands are configured via the structure components *testCmd*, *clockSync*, *arrGenro, arrCoro* or *genCmd.*

Syntax

```
TYPE ST_IEC870_5_101AcquisitionParams :
STRUCT
    eMode : E_IEC870_5_101AcquisitionMode := E_IEC870_5_101AcquisitionMode.InitSeq_CyclicPoll;

    arrInitSeq : ARRAY[IEC870_MIN_ISTEP..IEC870_MAX_ISTEP] OF E_IEC870_5_101InitSeqStep :=
        [eIEC870_ISTEP_TEST,
         eIEC870_ISTEP_CLOCK,
         eIEC870_ISTEP_GENRO,
         eIEC870_ISTEP_CORO,
         eIEC870_ISTEP_UNUSED,
         eIEC870_ISTEP_UNUSED,
         eIEC870_ISTEP_UNUSED];

    testCmd : ST_IEC870_5_101TestPollParams :=
            (asduAddr := IEC870_ASDUADDR_SYSPARA, tPollCycle := T#60S, bEnable := TRUE);

    clockSync : ST_IEC870_5_101ClockPollParams :=
              (asduAddr := IEC870_ASDUADDR_SYSPARA, tPollCycle := T#60S, bEnable := TRUE);

    arrGenro : ARRAY[0..16] OF ST_IEC870_5_101GenroPollParams :=
        [(eQOI := eIEC870_QOI_INROGEN, tPollCycle := T#60S, bEnable := TRUE),
         (eQOI := eIEC870_QOI_INRO1),
         (eQOI := eIEC870_QOI_INRO2),
         (eQOI := eIEC870_QOI_INRO3),
         (eQOI := eIEC870_QOI_INRO4),
         (eQOI := eIEC870_QOI_INRO5),
         (eQOI := eIEC870_QOI_INRO6),
         (eQOI := eIEC870_QOI_INRO7),
         (eQOI := eIEC870_QOI_INRO8),
         (eQOI := eIEC870_QOI_INRO9),
         (eQOI := eIEC870_QOI_INRO10),
         (eQOI := eIEC870_QOI_INRO11),
         (eQOI := eIEC870_QOI_INRO12),
         (eQOI := eIEC870_QOI_INRO13),
         (eQOI := eIEC870_QOI_INRO14),
         (eQOI := eIEC870_QOI_INRO15),
         (eQOI := eIEC870_QOI_INRO16)];

    arrCoro : ARRAY[0..4] OF ST_IEC870_5_101CoroPollParams :=
        [(eRQT := eIEC870_RQT_REQCOGEN, eFRZ := eIEC870_FRZ_READ, tPollCycle := T#60S, bEnable := TR
UE),
         (eRQT := eIEC870_RQT_REQCO1),
         (eRQT := eIEC870_RQT_REQCO2),
         (eRQT := eIEC870_RQT_REQCO3),
         (eRQT := eIEC870_RQT_REQCO4)];

    genCmd : ST_IEC870_5_101GenCmdPollParams :=
        (tPollCycle := T#1H, bEnable := FALSE);
```

```
    delayCmd : ST_IEC870_5_101DelayPollParams := (tDelay := T#5S);

    eAODBType : E_IEC870_5_101AODBType := eIEC870_AODB_STATIC;
END_STRUCT
END_TYPE
```

eMode: Configuration of the trigger [▶ 600] for processing the initialization sequence (arrInitSeq) and the cyclic interrogation commands. Since not every server always sends the ASDU: M_EI_NA_1 (end of initialization), the following standard setting was selected: E_IEC870_5_101AcquisitionMode.InitSeq_CyclicPoll.

In the standard setting, after the start of the data transfer (IEC 60870-5-104, STARTDT) or the reset of the connection (IEC 60870-5-101), first all commands of the initialization sequence are executed once, then the cyclic interrogation commands are executed continuously. The ASDU: M_EI_NA_1 is not evaluated and ignored.

If the initialization sequence is to be executed when M_EI_NA_1 is received, another setting must be selected, e.g: E_IEC870_5_101AcquisitionMode.CyclicPoll_OnEndOfInit_InitSeq_CyclicPoll or E_IEC870_5_101AcquisitionMode.OnEndOfInit_InitSeq_CyclicPoll.

arrInitSeq: Initialization sequence [▶ 617]. The standard configuration of the initialization sequence executes the following commands:

- Test command (eIEC870_ISTEP_TEST);
- Clock synchronization command (eIEC870_ISTEP_CLOCK);
- General interrogation (eIEC870_ISTEP_GENRO, General group);
- Counter interrogation (eIEC870_ISTEP_CORO, Counter group General, Read counter);

testCmd: Parameter for cyclic test commands [▶ 595]. Default: Active; test command every 60 seconds.

clockSync: Parameter for cyclic clock synchronization commands [▶ 596]. Default: Active; clock synchronization every 60 seconds.

arrGenro: Parameter for cyclic station interrogation commands [▶ 596]. Station interrogations for up to 17 data point groups may be configured. Default: Active; station interrogation for group 'General' every 60 seconds.

arrCoro: Parameter for cyclic counter interrogation commands [▶ 597]. Counter interrogations for up to 5 counter groups may be configured. Default: Active; counter interrogation of the group General every 60 seconds.

genCmd: Parameter for cyclic transfer of the data points in control direction [▶ 597] (single command, double command, setpoints etc.). Default: Not active; the commands are transferred every 60 minutes.

delayCmd: Parameter for delaying the next initialization step [▶ 598] during execution of the initialization sequence. Default: The next initialization step is delayed by 5 seconds.

eAODBType: Application database type [▶ 617]. This parameter determines how the data points are stored in the application database.

Sample in ST

In the following program section cyclic data acquisition is configured as follows: All initialization steps are deactivated. The cyclic test command and time synchronization command are also deactivated. In addition to the standard station interrogation a further station interrogation from group 1 is configured (every 100 s). In addition to the standard counter interrogation, a counter interrogation (freezing) from counter group 1 is configured (every 200 s).

```
PROGRAM P_AcquisitionConfig
VAR_IN_OUT
    acqPara : ST_IEC870_5_101AcquisitionParams;
END_VAR
acqPara.arrInitSeq[0]          := eIEC870_ISTEP_UNUSED;
acqPara.arrInitSeq[1]          := eIEC870_ISTEP_UNUSED;
acqPara.arrInitSeq[2]          := eIEC870_ISTEP_UNUSED;
acqPara.arrInitSeq[3]          := eIEC870_ISTEP_UNUSED;
```

```
acqPara.arrInitSeq[4]          := eIEC870_ISTEP_UNUSED;

acqPara.testCmd.bEnable        := FALSE;

acqPara.clockSync.bEnable      := FALSE;

acqPara.arrGenro[1].asduAddr   := IEC870_ASDUADDR_SYSPARA;
acqPara.arrGenro[1].eQOI       := eIEC870_QOI_INRO1;
acqPara.arrGenro[1].tPollCycle:= T#100s;
acqPara.arrGenro[1].bEnable    := TRUE;

acqPara.arrCoro[1].asduAddr    := IEC870_ASDUADDR_SYSPARA;
acqPara.arrCoro[1].eFRZ        := eIEC870_FRZ_FREEZE;
acqPara.arrCoro[1].eRQT        := eIEC870_RQT_REQCO1;
acqpara.arrCoro[1].tPollCycle  := T#200s;
acqPara.arrCoro[1].bEnable     := TRUE;
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.16 ST_IEC870_5_101TestPollParams

Configuration parameters for the cyclic test command.

Syntax

```
TYPE ST_IEC870_5_101TestPollParams:
STRUCT
    asduAddr        : DWORD := IEC870_ASDUADDR_SYSPARA;
    tPollCycle      : TIME := T#60s;
    bEnable         : BOOL := FALSE;
END_STRUCT
END_TYPE
```

asduAddr: Destination address of the test command (common ASDU address);

tPollCycle: Cycle time of the test command.

bEnable: Enables (TRUE) or disables (FALSE) the cyclic test commands.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.17 ST_IEC870_5_101ClockPollParams

Configuration parameters for the cyclic clock synchronization command. The station sending the clock synchronization commands has its own internal software clock. During station initialization, this clock is synchronized with the local Windows system time (the time shown in the Windows taskbar). During operation, the destination stations are synchronized with the time of the internal software clock via the clock synchronization commands.

Syntax

```
TYPE ST_IEC870_5_101ClockPollParams:
STRUCT
    asduAddr        : DWORD := IEC870_ASDUADDR_SYSPARA;
    tPollCycle      : TIME := T#60s;
    bEnable         : BOOL := FALSE;
END_STRUCT
END_TYPE
```

asduAddr: Destination address of the time synchronization command (common ASDU address).

tPollCycle: Cycle time of the time synchronization command.

bEnable: Activates/deactivates the cyclic clock synchronization commands.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.18 ST_IEC870_5_101GenroPollParams

Configuration parameters for the cyclic station interrogation commands.

Syntax

```
TYPE ST_IEC870_5_101GenroPollParams:
STRUCT
    asduAddr        : DWORD := IEC870_ASDUADDR_SYSPARA;
    tPollCycle      : TIME := T#60s;
    eQOI            : E_IEC870_5_101QOI := eIEC870_QOI_INROGEN;
    bEnable         : BOOL := FALSE;
END_STRUCT
END_TYPE
```

asduAddr: Destination address for the station interrogation command (CAA).

tPollCycle: Station interrogation cycle time.

eQOI: Interrogation command parameter/ID. [▶ 623]

bEnable: Enables (TRUE) or disables (FALSE) the station interrogation commands.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.19 ST_IEC870_5_101CoroPollParams

Configuration parameters for the cyclic counter interrogation command.

Syntax

```
TYPE ST_IEC870_5_101CoroPollParams:
STRUCT
    asduAddr    : DWORD := IEC870_ASDUADDR_SYSPARA;
    tPollCycle  : TIME := T#60s;
    eRQT        : E_IEC870_5_101RQT := eIEC870_RQT_REQCOGEN;
    eFRZ        : E_IEC870_5_101FRZ := eIEC870_FRZ_READ;
    bEnable     : BOOL := FALSE;
END_STRUCT
END_TYPE
```

asduAddr: Destination address of the counter interrogation command (common ASDU address).

tPollCycle: Cycle time of the counter interrogation command.

eRQT: Counter interrogation ID [▶ 625].

eFRZ: FREEZE/RESET ID [▶ 624].

bEnable: Activates/deactivates the cyclic counter interrogation commands.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.20 ST_IEC870_5_101GenCmdPollParams

Parameter for cyclic transfer of the data points in control direction (single command, double command, setpoints etc.).

The (high-level) central station function blocks FB_IEC870_5_101Master and FB_IEC870_5_104Master implement a simple/automatic command execution. In its implementation, a command is activated/executed as soon as the process value of a control data point has changed (example: the value of a single command (C_SC_NA_1) changes from OFF to ON). In some cases, however, it may be necessary to resend the values of all control data points to the substation, especially after the loss and restoration of the communication link.

In this case, the cyclic transfer of the data points can be useful. Please note that there may be concurrent access to command execution if, for example, the data points should be cyclically transmitted in the control direction (command execution) and a control data point is already executing a command (by spontaneous value changes at the data point in the control direction). The active command must first be executed to the end. Only then can a new command be activated. Because multiple primitives (activation, confirmation of activation and termination) are involved in command execution, a command execution time of a few milliseconds to seconds/minutes can occur, depending on the system. This should be taken into account when implementing a central station application. Filtering should then be carried out for rapidly changing PLC setpoints. E.g. with analog setpoints, the least significant bits can be masked out. In this way, the number of command activations can be reduced.

In the case of concurrent access to command execution, commands can be activated (repeated) at a later point in time. However, the repeated command is activated with the setpoint available at a later time.

Syntax

```
TYPE ST_IEC870_5_101GenCmdPollParams:
STRUCT
    tPollCycle           : TIME := T#1h;
    bEnable              : BOOL := FALSE;
    options              : DWORD := 0;
    bQueueConcurrentCmds : BOOL := FALSE;
END_STRUCT
END_TYPE
```

tPollCycle: Transfer cycle time.

bEnable: Activates/deactivates the cyclic transfer of the data points.

options: Additional parameters.

bQueueConcurrentCmds: Controls the behavior of command execution if a command is already executed at the time of cyclic command activation (e.g. triggered by a spontaneous value change in the PLC process data):

- FALSE (default value). For data points that are currently executing a command, no repeated commands are activated at a later time.
- TRUE: For data points that are currently executing a command, the command is activated later (after the currently active command execution is complete).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.21 ST_IEC870_5_101DelayPollParams

Configuration parameter for delaying the next initialization step.

Syntax

```
TYPE ST_IEC870_5_101DelayPollParams:
STRUCT
    tDelay : TIME := T#5s;
END_STRUCT
END_TYPE
```

tDelay: Delay time.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.22 ST_IEC870_5_101HashTableKey

Application object database lookup key. The key can be used for locating and modifying the hash table entries.

Syntax

```
TYPE ST_IEC870_5_101HashTableKey :
STRUCT
    eType       : E_IEC870_5_101TcTypeID := ASDU_TYPEUNDEF;
    asduAddr    : DWORD := 0;
    objAddr     : DWORD := 0;
    group       : DWORD := 0;
    lookup      : DWORD := IEC870_LOOKUP_KEY_ALL_ON;
END_STRUCT
END_TYPE
```

eType: IEC 60870-5-101/104 ASDU type ID [▶ 601] (e.g.: M_SP_NA_1 for single-point or M_DP_NA_1 for double point etc.).

asduAddr: Common ASDU address (CAA).

objAddr: Information object address (IOA).

group: Object group configuration flags [▶ 655]. The flags can be combined with an OR operation. If this parameter is null the group parameters are ignored.

lookup: Additional lookup key parameters. The available parameters are declared as constants (see table below). They can be combined with an OR operation.

Constant	Value	Description
IEC870_LOOKUP_KEY_ALL_ON	0	All parameters are considered in the search (*eType, asduAddr, objAddr, group*).
IEC870_LOOKUP_KEY_TYPE_OFF	1	The *eType* parameter is ignored in the search.
IEC870_LOOKUP_KEY_ASDUADDR_OFF	2	The *asduAddr* parameter is ignored in the search.
IEC870_LOOKUP_KEY_GROUP_OFF	4	The *group* parameter is ignored in the search.
IEC870_LOOKUP_KEY_OBJADDR_OFF	8	The *objAddr* parameter is ignored in the search. This parameter is not recommended because all data points must be identifiable via a unique object address.

Sample in ST:

See function description: F_iecLookupTableEntry [▶ 560], F_iecRemoveTableEntry [▶ 562], F_iecGetPosOfTableEntry [▶ 558].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.23 E_IEC870_5_101AcquisitionMode

From Tc2_IEC60870_5_10x.compiled-library v3.0.7.0, the trigger of the commands in the initialization sequence and/or the cyclic interrogation commands can be configured. The commands of the initialization sequence and the cyclic interrogation commands are executed after the communication link "STARTDT" (IEC 60870-5-104) has been established or after "reset of link" (IEC 60870-5-101) and/or after receipt of the ASDU: M_EI_NA_1 (end of initialization). The detailed configuration of the cyclic data acquisition takes place via the components of the structured type: ST_IEC870_5_101AcquisitionParams [▶ 593].

The initialization sequence includes commands configured in the ST_IEC870_5_101AcquisitionParams.arrInitSeq [▶ 594] component.

The cyclic interrogation commands are commands that have been configured via the following structure components:

- ST_IEC870_5_101AcquisitionParams.testCmd [▶ 594]
- ST_IEC870_5_101AcquisitionParams.clockSync [▶ 594]
- ST_IEC870_5_101AcquisitionParams.arrGenro [▶ 594]
- ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594]
- ST_IEC870_5_101AcquisitionParams.genCmd [▶ 594]

Syntax

```
TYPE E_IEC870_5_101AcquisitionMode :
(
    CyclicPoll                                      := 0,
    InitSeq_CyclicPoll                              := 1,
    CyclicPoll_OnEndOfInit_InitSeq_CyclicPoll       := 2,
    InitSeq_CyclicPoll_OnEndOfInit_InitSeq_CyclicPoll := 3,
    OnEndOfInit_InitSeq_CyclicPoll                  := 4
);
END_TYPE
```

Values

Name	Description
CyclicPoll	After receiving STARTDT/ResetOfLink, the processing of the cyclic interrogation commands is started and continued continuously. Commands in the initialization sequence (even if configured) are not executed. The ASDU: M_EI_NA_1 is not evaluated and ignored.
InitSeq_CyclicPoll	After receipt of STARTDT/ResetOfLink, the execution of the commands in the initialization sequence is started first. When all commands in the initialization sequence have been executed, the execution of the cyclic interrogation commands is started and continued continuously. The ASDU: M_EI_NA_1 is not evaluated and ignored.
CyclicPoll_OnEndOfInit_InitSeq_CyclicPoll	After receipt of STARTDT/ResetOfLink, the processing of the cyclic interrogation commands is started. However, the cyclic interrogation commands are only processed until receipt of ASDU: M_EI_NA_1. When ASDU: M_EI_NA_1 is received, the execution of the cyclic interrogation commands is interrupted (but the last active command is still completed) and the execution of the commands in the initialization sequence is started. After the execution of all commands in the initialization sequence, the execution of the cyclic interrogation commands is started again and continued continuously.
InitSeq_CyclicPoll_OnEndOfInit_InitSeq_CyclicPoll	After receipt of STARTDT/ResetOfLink and until receiving ASDU: M_EI_NA_1 the processing of the commands in the initialization sequence is started. Then (if all could be executed before receiving M_EI_NA_1) the processing of the cyclic interrogation commands is started. When ASDU: M_EI_NA_1 is received, the execution of the commands in the initialization sequence or the cyclic interrogation commands is interrupted (but the last active command is still completed) and the execution of the commands in the initialization sequence is restarted. After the execution of all commands in the initialization sequence, the execution of the cyclic interrogation commands is started again and continued continuously.
OnEndOfInit_InitSeq_CyclicPoll	The system waits for the receipt of the ASDU: M_EI_NA_1. When M_EI_NA_1 is received, the processing of the commands in the initialization sequence is started first. After the execution of all commands in the initialization sequence, the execution of the cyclic interrogation commands is started and continued continuously.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022.28 and higher	PC or CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x v3.0.7.0 and higher (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.24 E_IEC870_5_101TcTypeID

Data point type ID according to IEC 60870-5-101/104.

Syntax

```
TYPE E_IEC870_5_101TcTypeID :
(
    ASDU_TYPEUNDEF := 0, (* (000, 0x00) not allowed *)
    (* reserved standard asdu types *)
    M_SP_NA_1, (* (001, 0x01) Single-point information *)
    M_SP_TA_1, (* (002, 0x02) Single-point information with time tag *)
    M_DP_NA_1, (* (003, 0x03) Double-point information *)
    M_DP_TA_1, (* (004, 0x04) Double-point information with time tag *)
    M_ST_NA_1, (* (005, 0x05) Step position information *)
    M_ST_TA_1, (* (006, 0x06) Step position information with time tag *)
```

```
    M_BO_NA_1, (* (007, 0x07) Bitstring of 32 bit *)
    M_BO_TA_1, (* (008, 0x08) Bitstring of 32 bit with time tag *)
    M_ME_NA_1, (* (009, 0x09) Measured value, normalised value *)
    M_ME_TA_1, (* (010, 0x0A) Measured value, normalized value with time tag *)
    M_ME_NB_1, (* (011, 0x0B) Measured value, scaled value *)
    M_ME_TB_1, (* (012, 0x0C) Measured value, scaled value wit time tag *)
    M_ME_NC_1, (* (013, 0x0D) Measured value, short floating point number *)
    M_ME_TC_1, (* (014, 0x0E) Measured value, short floating point number with time tag *)
    M_IT_NA_1, (* (015, 0x0F) Integrated totals *)
    M_IT_TA_1, (* (016, 0x10) Integrated totals with time tag *)
    M_EP_TA_1, (* (017, 0x11) Event of protection equipment with time tag *)
    M_EP_TB_1, (* (018, 0x12) Packed start events of protection equipment with time tag *)
    M_EP_TC_1, (* (019, 0x13) Packed output circuit information of protection equipment with time ta
g *)
    M_PS_NA_1, (* (020, 0x14) Packed single point information with status change detection *)
    M_ME_ND_1, (* (021, 0x15) Measured value, normalized value without quality descriptor *)
    ASDU_TYPE_22,
    ASDU_TYPE_23,
    ASDU_TYPE_24,
    ASDU_TYPE_25,
    ASDU_TYPE_26,
    ASDU_TYPE_27,
    ASDU_TYPE_28,
    ASDU_TYPE_29,
    M_SP_TB_1, (* (030, 0x1E) Single-point information with time tag CP56Time2a *)
    M_DP_TB_1, (* (031, 0x1F) Double-point information with time tag CP56Time2a *)
    M_ST_TB_1, (* (032, 0x20) Step position information with time tag CP56Time2a *)
    M_BO_TB_1, (* (033, 0x21) Bitstring of 32 bit with time tag CP56Time2a *)
    M_ME_TD_1, (* (034, 0x22) Measured value, normalised value with time tag CP56Time2a *)
    M_ME_TE_1, (* (035, 0x23) Measured value, scaled value with time tag CP56Time2a *)
    M_ME_TF_1, (* (036, 0x24) Measured value, short floating point number with time tag CP56Time2a *
)
    M_IT_TB_1, (* (037, 0x25) Integrated totals with time tag CP56Time2a *)
    M_EP_TD_1, (* (038, 0x26) Event of protection equipment with time tag CP56Time2a *)
    M_EP_TE_1, (* (039, 0x27) Packed start events of protection equipment with time tag CP56Time2a *
)
    M_EP_TF_1, (* (040, 0x28) Packed output circuit information of protection equipment with time ta
g CP56Time2a *)
    ASDU_TYPE_41,
    ASDU_TYPE_42,
    ASDU_TYPE_43,
    ASDU_TYPE_44,
    C_SC_NA_1, (* (045, 0x2D) Single command *)
    C_DC_NA_1, (* (046, 0x2E) Double command *)
    C_RC_NA_1, (* (047, 0x2F) Regulating step command *)
    C_SE_NA_1, (* (048, 0x30) Set-point Command, normalised value *)
    C_SE_NB_1, (* (049, 0x31) Set-point Command, scaled value *)
    C_SE_NC_1, (* (050, 0x32) Set-point Command, short floating point number *)
    C_BO_NA_1, (* (051, 0x33) Bitstring 32 bit command *)
    ASDU_TYPE_52,
    ASDU_TYPE_53,
    ASDU_TYPE_54,
    ASDU_TYPE_55,
    ASDU_TYPE_56,
    ASDU_TYPE_57,
    C_SC_TA_1, (* (058, 0x3A) Single command with time tag CP56Time2a *)
    C_DC_TA_1, (* (059, 0x3B) Double command with time tag CP56Time2a *)
    C_RC_TA_1, (* (060, 0x3C) Regulating step command with time tag CP56Time2a *)
    C_SE_TA_1, (* (061, 0x3D) Measured value, normalised value command with time tag CP56Time2a *)
    C_SE_TB_1, (* (062, 0x3E) Measured value, scaled value command with time tag CP56Time2a *)
    C_SE_TC_1, (* (063, 0x3F) Measured value, short floating point number command with time tag CP56
Time2a *)
    C_BO_TA_1, (* (064, 0x40) Bitstring of 32 bit command with time tag CP56Time2a *)
    ASDU_TYPE_65,
    ASDU_TYPE_66,
    ASDU_TYPE_67,
    ASDU_TYPE_68,
    ASDU_TYPE_69,
    M_EI_NA_1, (* (070, 0x46) End of Initialisation *)
    ASDU_TYPE_71,
    ASDU_TYPE_72,
    ASDU_TYPE_73,
    ASDU_TYPE_74,
    ASDU_TYPE_75,
    ASDU_TYPE_76,
    ASDU_TYPE_77,
    ASDU_TYPE_78,
    ASDU_TYPE_79,
    ASDU_TYPE_80,
    ASDU_TYPE_81,
```

```
    ASDU_TYPE_82,
    ASDU_TYPE_83,
    ASDU_TYPE_84,
    ASDU_TYPE_85,
    ASDU_TYPE_86,
    ASDU_TYPE_87,
    ASDU_TYPE_88,
    ASDU_TYPE_89,
    ASDU_TYPE_90,
    ASDU_TYPE_91,
    ASDU_TYPE_92,
    ASDU_TYPE_93,
    ASDU_TYPE_94,
    ASDU_TYPE_95,
    ASDU_TYPE_96,
    ASDU_TYPE_97,
    ASDU_TYPE_98,
    ASDU_TYPE_99,
    C_IC_NA_1, (* (100, 0x64) Interrogation command *)
    C_CI_NA_1, (* (101, 0x65) Counter interrogation command *)
    C_RD_NA_1, (* (102, 0x66) Read Command*)
    C_CS_NA_1, (* (103, 0x67) Clock synchronisation command *)
    C_TS_NA_1, (* (104, 0x68) Test command *)
    C_RP_NA_1, (* (105, 0x69) Reset process command *)
    C_CD_NA_1, (* (106, 0x6A) C_CD_NA_1 Delay acquisition command *)
    C_TS_TA_1, (* (107, 0x6B) Test command with time tag CP56Time2a *)
    ASDU_TYPE_108,
    ASDU_TYPE_109,
    P_ME_NA_1, (* (110, 0x6E) Parameter of measured values, normalized value *)
    P_ME_NB_1, (* (111, 0x6F) Parameter of measured values, scaled value *)
    P_ME_NC_1, (* (112, 0x70) Parameter of measured values, short floating point number *)
    P_AC_NA_1, (* (113, 0x71) Parameter activation *)
    ASDU_TYPE_114,
    ASDU_TYPE_115,
    ASDU_TYPE_116,
    ASDU_TYPE_117,
    ASDU_TYPE_118,
    ASDU_TYPE_119,
    F_FR_NA_1, (* (120, 0x78) File ready *)
    F_SR_NA_1, (* (121, 0x79) Section ready *)
    F_SC_NA_1, (* (122, 0x7A) Call directory, select file, call file, call section *)
    F_LS_NA_1, (* (123, 0x7B) Last section, last segment *)
    F_FA_NA_1, (* (124, 0x7C) ACK file, ACK section *)
    F_SG_NA_1, (* (125, 0x7D) Segment *)
    F_DR_TA_1, (* (126, 0x7E) Directory *)
    ASDU_TYPE_127,
    (* reserved user asdu types *)
    ASDU_TYPE_128,
    ASDU_TYPE_129,
    ASDU_TYPE_130,
    ASDU_TYPE_131,
    ASDU_TYPE_132,
    ASDU_TYPE_133,
    ASDU_TYPE_134,
    ASDU_TYPE_135,
    ASDU_TYPE_136,
    ASDU_TYPE_137,
    ASDU_TYPE_138,
    ASDU_TYPE_139,
    ASDU_TYPE_140,
    ASDU_TYPE_141,
    ASDU_TYPE_142,
    ASDU_TYPE_143,
    ASDU_TYPE_144,
    ASDU_TYPE_145,
    ASDU_TYPE_146,
    ASDU_TYPE_147,
    ASDU_TYPE_148,
    ASDU_TYPE_149,
    ASDU_TYPE_150,
    ASDU_TYPE_151,
    ASDU_TYPE_152,
    ASDU_TYPE_153,
    ASDU_TYPE_154,
    ASDU_TYPE_155,
    ASDU_TYPE_156,
    ASDU_TYPE_157,
    ASDU_TYPE_158,
    ASDU_TYPE_159,
    ASDU_TYPE_160,
```

```
    ASDU_TYPE_161,
    ASDU_TYPE_162,
    ASDU_TYPE_163,
    ASDU_TYPE_164,
    ASDU_TYPE_165,
    ASDU_TYPE_166,
    ASDU_TYPE_167,
    ASDU_TYPE_168,
    ASDU_TYPE_169,
    ASDU_TYPE_170,
    ASDU_TYPE_171,
    ASDU_TYPE_172,
    ASDU_TYPE_173,
    ASDU_TYPE_174,
    ASDU_TYPE_175,
    ASDU_TYPE_176,
    ASDU_TYPE_177,
    ASDU_TYPE_178,
    ASDU_TYPE_179,
    ASDU_TYPE_180,
    ASDU_TYPE_181,
    ASDU_TYPE_182,
    ASDU_TYPE_183,
    ASDU_TYPE_184,
    ASDU_TYPE_185,
    ASDU_TYPE_186,
    ASDU_TYPE_187,
    ASDU_TYPE_188,
    ASDU_TYPE_189,
    ASDU_TYPE_190,
    ASDU_TYPE_191,
    ASDU_TYPE_192,
    ASDU_TYPE_193,
    ASDU_TYPE_194,
    ASDU_TYPE_195,
    ASDU_TYPE_196,
    ASDU_TYPE_197,
    ASDU_TYPE_198,
    ASDU_TYPE_199,
    ASDU_TYPE_200,
    ASDU_TYPE_201,
    ASDU_TYPE_202,
    ASDU_TYPE_203,
    ASDU_TYPE_204,
    ASDU_TYPE_205,
    ASDU_TYPE_206,
    ASDU_TYPE_207,
    ASDU_TYPE_208,
    ASDU_TYPE_209,
    ASDU_TYPE_210,
    ASDU_TYPE_211,
    ASDU_TYPE_212,
    ASDU_TYPE_213,
    ASDU_TYPE_214,
    ASDU_TYPE_215,
    ASDU_TYPE_216,
    ASDU_TYPE_217,
    ASDU_TYPE_218,
    ASDU_TYPE_219,
    ASDU_TYPE_220,
    ASDU_TYPE_221,
    ASDU_TYPE_222,
    ASDU_TYPE_223,
    ASDU_TYPE_224,
    ASDU_TYPE_225,
    ASDU_TYPE_226,
    ASDU_TYPE_227,
    ASDU_TYPE_228,
    ASDU_TYPE_229,
    ASDU_TYPE_230,
    ASDU_TYPE_231,
    ASDU_TYPE_232,
    ASDU_TYPE_233,
    ASDU_TYPE_234,
    ASDU_TYPE_235,
    ASDU_TYPE_236,
    ASDU_TYPE_237,
    ASDU_TYPE_238,
    ASDU_TYPE_239,
    ASDU_TYPE_240,
```

```
    ASDU_TYPE_241,
    ASDU_TYPE_242,
    ASDU_TYPE_243,
    ASDU_TYPE_244,
    ASDU_TYPE_245,
    ASDU_TYPE_246,
    ASDU_TYPE_247,
    ASDU_TYPE_248,
    ASDU_TYPE_249,
    ASDU_TYPE_250,
    ASDU_TYPE_251,
    ASDU_TYPE_252,
    ASDU_TYPE_253,
    ASDU_TYPE_254,
    ASDU_TYPE_255,
    ASDU_TYPEMAX (* not used *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.25 E_IEC870_5_102TypeID

ASDU type IDs according to IEC 60870-5-102.

Syntax

```
TYPE E_IEC870_5_102TypeID:
(
    ASDU_TYPEUNDEF_2 := 0, (* (000) not allowed *)(* Process inormation in monitor direction *)
    M_SP_TA_2 := 1, (* (001) Single-point information with time tag *)
    M_IT_TA_2 := 2, (* (002) Accounting integrated totals, 4 octets each *)
    M_IT_TB_2 := 3, (* (003) Accounting integrated totals, 3 octets each *)
    M_IT_TC_2 := 4, (* (004) Accounting integrated totals, 2 octets each *)
    M_IT_TD_2 := 5, (* (005) Periodical reset accounting integrated totals, 4 octets each *)
    M_IT_TE_2 := 6, (* (006) Periodical reset accounting integrated totals, 3 octets each *)
    M_IT_TF_2 := 7, (* (007) Periodical reset accounting integrated totals, 2 octets each *)
    M_IT_TG_2 := 8, (* (008) Operational integrated totals, 4 octets each *)
    M_IT_TH_2 := 9, (* (009) Operational integrated totals, 3 octets each *)
    M_IT_TI_2 := 10, (* (010) Operational integrated totals, 2 octets each *)
    M_IT_TK_2 := 11, (* (011) Periodical reset operational integrated totals, 4 octets each *)
    M_IT_TL_2 := 12, (* (012) Periodical reset operational integrated totals, 3 octets each *)
    M_IT_TM_2 := 13, (* (013) Periodical reset operational integrated totals, 2 octets each *)
    (* System information in monitor direction *)
    M_EI_NA_2 := 70, (* (070) End of initialization *)
    P_MP_NA_2 := 71, (* (071) Manufacturer and product specification of integrated total DTE *)
    M_TI_TA_2 := 72, (* (072) Current system time of integrated total DTE *)
    (* System information in control direction *)
    C_RD_NA_2 := 100, (* (100) Read manufacturer and product specification *)
    C_SP_NA_2 := 101, (* (101) Read record of single-point information with time tag *)
    C_SP_NB_2 := 102, (* (102) Read record of single-
point information with time tag of a selected time range *)
    C_TI_NA_2 := 103, (* (103) Read current system time of integrated total DTE *)
    C_CI_NA_2 := 104, (* (104) Read accounting integrated totals of the oldest integration period *)
    C_CI_NB_2 := 105, (* (105) Read accounting integrated totals of the oldest integration period an
d of a selected range of addresses *)
    C_CI_NC_2 := 106, (* (106) Read accounting integrated totals of a specific past integration peri
od *)
    C_CI_ND_2 := 107, (* (107) Read accounting integrated totals of a specific past integration peri
od and of a selected range of addresses *)
    C_CI_NE_2 := 108, (* (108) Read periodical reset accounting integrated totals of the oldest inte
gration period *)
    C_CI_NF_2 := 109, (* (109) Read periodical reset accounting integrated totals of the oldest inte
gration period and of a selected range of addresses *)
```

```
    C_CI_NG_2 := 110, (* (110) Read periodical reset accounting integrated totals of a specific past
 integration period *)
    C_CI_NH_2 := 111, (* (111) Read periodical reset accounting integrated totals of a specific past
 integration period and of a selected range of addresses *)
    C_CI_NI_2 := 112, (* (112) Read operational integrated totals of the oldest integration period *
)
    C_CI_NK_2 := 113, (* (113) Read operational integrated totals of the oldest integration period a
nd of a selected range of addresses *)
    C_CI_NL_2 := 114, (* (114) Read operational integrated totals of a specific past integration per
iod *)
    C_CI_NM_2 := 115, (* (115) Read operational integrated totals of a specific past integration per
iod and of a selected range of addresses *)
    C_CI_NN_2 := 116, (* (116) Read periodical reset operational integrated totals of the oldest int
egration period *)
    C_CI_NO_2 := 117, (* (117) Read periodical reset operational integrated totals of the oldest int
egration period and of a selected range of addresses *)
    C_CI_NP_2 := 118, (* (118) Read periodical reset operational integrated totals of a specific pas
t integration period *)
    C_CI_NQ_2 := 119, (* (119) Read periodical reset operational integrated totals of a specific pas
t integration period and of a selected range of addresses *)
    C_CI_NR_2 := 120, (* (120) Read accounting integrated totals of a specific past integration peri
od of a selected time range and of a selected range of addresses *)
    C_CI_NS_2 := 121, (* (121) Read periodical reset accounting integrated totals of a specific past
 integration period of a selected time range and of a selected range of addresses *)
    C_CI_NT_2 := 122, (* (122) Read operational integrated totals of a specific past integration per
iod of a selected time range and of a selected range of addresses *)
    C_CI_NU_2 := 123, (* (123) Read periodical reset operational integrated totals of a specific pas
t integration period of a selected time range and of a selected range of addresses *)

    M_DS_TA_2 := 128, (* (128) *)
    P_ME_NA_2 := 129, (* Parameters of the measuring point *)
    M_DS_TB_2 := 130, (**)
    M_CH_TA_2 := 131, (**)
    C_PK_2 := 132, (* Load private key *)
    C_TA_VC_2 := 133, (* Read tariff information ( current values ) *)
    C_TA_VM_2 := 134, (* Read tariff information ( stored values ) *)
    M_TA_VC_2 := 135, (* Tariff information ( current values ) *)
    M_TA_VM_2 := 136, (* Tariff information ( stored values ) *)
    C_TA_CP_2 := 137, (* Close accounting period *)
    M_IB_TG_2 := 139, (* Block of operational integrated totals ( absolute values ) *)
    M_IB_TK_2 := 140, (* Block of periodical reset operational integrated totals ( increment values
) *)
    C_RM_NA_2 := 141, (* Read configuration data of the meter device *)
    M_RM_NA_2 := 142, (* Configuration of the meter device *)
    C_MR_NA_2 := 143, (* Change configuration data of the meter device *)
    C_PC_NA_2 := 144, (* (144) *)
    M_PC_NA_2 := 145, (* (145) *)
    C_MC_NA_2 := 146, (* (146) *)
    C_DF_NA_2 := 147, (* (147) *)
    M_DF_NA_2 := 148, (* (148) *)
    C_MF_NA_2 := 149, (* (149) *)

    C_DS_TA_2 := 180, (* (180) *)
    C_CS_TA_2 := 181, (* (181) Change date and time ( Time syncronization ) *)
    C_PI_NA_2 := 182, (* (182) Read parameters of the measuring point *)
    C_AC_NA_2 := 183, (* (183) Start session and send access key *)

    C_DS_TB_2 := 184, (* (184) *)
    C_CH_TA_2 := 185, (* (185) *)
    C_MH_TA_2 := 186, (* (186) *)
    C_FS_NA_2 := 187, (* (187) Finish session *)
    C_MP_NA_2 := 188, (* (188) *)
    C_CB_NT_2 := 189, (* (189) Read a block of operational integrated totals of a time period and a
selected address *)
    C_CB_UN_2 := 190 (* (190) Read a block of periodical reset operational integrated totals of a ti
me period and a selected address *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.26 E_IEC870_5_103MTypeID

ASDU type IDs in monitoring direction (slave -> master) according to IEC 60870-5-103.

Syntax

```
TYPE E_IEC870_5_103MTypeID:
(
    M_TYPEUNDEF_3 := 0, (* Not used *)
    M_TTM_TA_3 := 1, (* Time-tagged message *)
    M_TMR_TA_3 := 2, (* Time-tagged message with relative time *)
    M_MEI_NA_3 := 3, (* Measurands I *)
    M_TME_TA_3 := 4, (* Time-tagged measurands with relative time *)
    M_IRC_NA_3 := 5, (* Identification *)
    M_SYN_TA_3 := 6, (* Time synchronisation *)
    M_TGI_NA_3 := 8, (* General interrogation *)
    M_MEII_NA_3 := 9, (* Measurands II *)
    M_GD_XA_3 := 10,(* Generic data *)
    M_GI_XA_3 := 11,(* Generic identification *)
    M_LRD_TA_3 := 23,(* List of recorded disturbances *)
    M_RTD_TA_3 := 26,(* Ready for transmission of disturbance data *)
    M_RTC_NA_3 := 27,(* Ready for transmission of channel *)
    M_RTT_NA_3 := 28,(* Ready for transmission of tags *)
    M_TOT_NA_3 := 29,(* Transmission of tags *)
    M_TOV_NA_3 := 30,(* Transmission of disturbance values *)
    M_EOT_NA_3 := 31 (* End of transmission *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.27 E_IEC870_5_103CTypeID

ASDU type IDs in control direction (master -> slave) according to IEC 60870-5-103.

Syntax

```
TYPE E_IEC870_5_103CTypeID:
(
    C_TYPEUNDEF_3 := 0, (* Not used *)
    C_SYN_TA_3 := 6, (* Time synchronisation *)
    C_IGI_NA_3 := 7, (* General interrogation *)
    C_GD_NA_3 := 10,(* Generic data *)
    C_GRC_NA_3 := 20,(* General command *)
    C_GC_NA_3 := 21,(* Generic command *)
    C_ODT_NA_3 := 24,(* Order for disturbance data transmission *)
```

```
    C_ADT_NA_3 := 25 (* Acknowledgement for disturbance data transmission *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.28 E_IEC870_5_101IOMappingType

TwinCAT PLC process data area (inputs, outputs, memory, data) into/from which the IEC process data are to be mapped (copied).

Syntax

```
TYPE E_IEC870_5_101IOMappingType :
(
    MAP_AREA_NONE   := 0,
    MAP_AREA_MEMORY := 1,
    MAP_AREA_INPUT  := 2,
    MAP_AREA_OUTPUT := 4,
    MAP_AREA_DATA   := 8
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.29 E_IEC870_5_101AsduAddrSize

Octet size (byte size) of the common ASDU address.

Syntax

```
TYPE E_IEC870_5_101AsduAddrSize :
(
    eIEC870_AsduAddr_OneOctet  := 1,(* Ein Byte *)
    eIEC870_AsduAddr_TwoOctets := 2 (* Zwei Byte *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.30 E_IEC870_5_101COTSize

Octet size (byte size) of the cause of transfer.

Syntax

```
TYPE E_IEC870_5_101COTSize :
(
    eIEC870_COT_OneOctet  := 1,(* Ein Byte *)
    eIEC870_COT_TwoOctets := 2 (* Zwei Byte (inklusive der Quelladresse) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.31 E_IEC870_5_101LinkAddrSize

Octet size (byte size) of the link address.

Syntax

```
TYPE E_IEC870_5_101LinkAddrSize :
(
    eIEC870_LinkAddr_None      := 0,(* Keine (wird nicht benutzt) *)
    eIEC870_LinkAddr_OneOctet  := 1,(* Ein Byte *)
    eIEC870_LinkAddr_TwoOctets := 2 (* Zwei Byte *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.32 E_IEC870_5_101ObjAddrSize

Octet size (byte size) of the information object address (IOA).

Syntax

```
TYPE E_IEC870_5_101ObjAddrSize :
(
    eIEC870_ObjAddr_OneOctet    := 1,(* Ein Byte *)
    eIEC870_ObjAddr_TwoOctets   := 2,(* Zwei Byte *)
    eIEC870_ObjAddr_ThreeOctets := 3 (* Drei Byte *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.33 E_IEC870_5_101ErrorSourceID

Source of error.

Syntax

```
TYPE E_IEC870_5_101ErrorSourceID :
(
    eIEC870_ESRC_NONE,              (* Fehlerquelle ist unbekannt *)
    eIEC870_ESRC_COUNTER_INTERROGATION,(* Fehler während der Ausführung der Zählwertabfrage *)
    eIEC870_ESRC_SYNC_CLOCK_CTRL,   (* Fehler während der Ausführung der Uhrzeitsynchronisation *)
    eIEC870_ESRC_CLOCK_EVENT,       (* Fehler während der spontanen Übertragung der Uhrzeit *)
    eIEC870_ESRC_GETPCTIME,         (* Fehler während der Synchronisierung der Geräte-
Uhrzeit mit der Uhrzeit des PC's *)
    eIEC870_ESRC_SETPCTIME,         (* Fehler während der Synchronisierung der PC-
Uhrzeit mit der Uhrzeit des Gerätes *)
    eIEC870_ESRC_STATION_INTERROGATION,(* Fehler während der Generalabfrage *)
    eIEC870_ESRC_READ_DATA_CMD,     (* Fehler beim Lesekommando *)
    eIEC870_ESRC_RESET_PROCESS,     (* Fehler beim Process-Reset-Kommando *)
    eIEC870_ESRC_TEST_CMD,          (* Fehler beim Testkommando *)
    eIEC870_ESRC_ENDOFINIT,         (* Fehler bei M_EI_NA_1 (Ende der Initialisierung) *)
    eIEC870_ESRC_BACKGROUND_SCAN,   (* Fehler bei der Ausführung der Hintegrundabfrage *)
    eIEC870_ESRC_COMMAND_CTRL,      (* Fehler bei der Initialisierung des Befehls (single-
command, double-command, setpoint-command ...) *)
    eIEC870_ESRC_COMMAND_EXEC,      (* Fehler bei der Ausführung des Befehls *)
    eIEC870_ESRC_LOCAL_FREEZE_RESET,(* Fehler beim Umspeichern / Zurücksetzen *)
    eIEC870_ESRC_PERIODIC_CYCLIC,   (* Fehler bei der periodischen/zyklischen Datenübertragung *)
    eIEC870_ESRC_USERAPP_OBJECT,    (* Fehler im Applikationsobjekt (fehlerhafte Konfiguration/
Wert) *)
    eIEC870_ESRC_USERAPP_SETQUALITY,(* Fehler bei der Quality-Flag-Bearbeitung *)
    eIEC870_ESRC_IEC60870_5_104LINK,(* Fehler im IEC60870-5-104 Link Layer *)
    eIEC870_ESRC_IEC60870_5_101LINK,(* Fehler im IEC60870-5-101 Link Layer *)
    eIEC870_ESRC_COMLIB,            (* Fehler in der unterlagerten Kommunikation über die seriellen
Schnittstellen *)
    eIEC870_ESRC_POLLING_SEVICE     (* Fehler in der Acquisition-State-
Machine (Abarbeitung der zyklische Pollanfragen)*)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.34 E_IEC870_5_101ClassType

Priority class of the ASDU. High-priority data are assigned to class 1, low-priority data to class 2.

Syntax

```
TYPE E_IEC870_5_101ClassType :
(
    eIEC870_Class_None,
    eIEC870_Class_1,
    eIEC870_Class_2
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.35 E_IEC870_5_101COTType

IEC 60870-5-101/104 cause of transfer.

Syntax

```
TYPE E_IEC870_5_101COTType:
(
    eIEC870_COT_UNUSED := 0,
    eIEC870_COT_CYCLIC := 1,
    eIEC870_COT_BACKGROUND := 2,
    eIEC870_COT_SPONTAN := 3,
    eIEC870_COT_INIT := 4,
    eIEC870_COT_REQ := 5,
    eIEC870_COT_ACT := 6,
    eIEC870_COT_ACT_CON := 7,
    eIEC870_COT_DEACT := 8,
    eIEC870_COT_DEACT_CON := 9,
    eIEC870_COT_ACT_TERM := 10,
    eIEC870_COT_RETREM := 11,
    eIEC870_COT_RETLOC := 12,
    eIEC870_COT_FILE := 13,
    eIEC870_COT_14 := 14,
    eIEC870_COT_15 := 15,
    eIEC870_COT_16 := 16,
    eIEC870_COT_17 := 17,
    eIEC870_COT_18 := 18,
    eIEC870_COT_19 := 19,
    eIEC870_COT_INROGEN := 20,
    eIEC870_COT_INRO1 := 21,
```

```
    eIEC870_COT_INRO2 := 22,
    eIEC870_COT_INRO3 := 23,
    eIEC870_COT_INRO4 := 24,
    eIEC870_COT_INRO5 := 25,
    eIEC870_COT_INRO6 := 26,
    eIEC870_COT_INRO7 := 27,
    eIEC870_COT_INRO8 := 28,
    eIEC870_COT_INRO9 := 29,
    eIEC870_COT_INRO10 := 30,
    eIEC870_COT_INRO11 := 31,
    eIEC870_COT_INRO12 := 32,
    eIEC870_COT_INRO13 := 33,
    eIEC870_COT_INRO14 := 34,
    eIEC870_COT_INRO15 := 35,
    eIEC870_COT_INRO16 := 36,
    eIEC870_COT_REQCOGEN := 37,
    eIEC870_COT_REQCO1 := 38,
    eIEC870_COT_REQCO2 := 39,
    eIEC870_COT_REQCO3 := 40,
    eIEC870_COT_REQCO4 := 41,
    eIEC870_COT_42 := 42,
    eIEC870_COT_43 := 43,
    eIEC870_COT_UNKNOWN_TYPE := 44,
    eIEC870_COT_UNKNOWN_CAUSE := 45,
    eIEC870_COT_UNKNOWN_ASDU_ADDRESS := 46,
    eIEC870_COT_UNKNOWN_OBJECT_ADDRESS := 47,
    eIEC870_COT_48 := 48,
    eIEC870_COT_49 := 49,
    eIEC870_COT_50 := 50,
    eIEC870_COT_51 := 51,
    eIEC870_COT_52 := 52,
    eIEC870_COT_53 := 53,
    eIEC870_COT_54 := 54,
    eIEC870_COT_55 := 55,
    eIEC870_COT_56 := 56,
    eIEC870_COT_57 := 57,
    eIEC870_COT_58 := 58,
    eIEC870_COT_59 := 59,
    eIEC870_COT_60 := 60,
    eIEC870_COT_61 := 61,
    eIEC870_COT_62 := 62,
    eIEC870_COT_63 := 63
);
END_TYPE
```

Value	Description
eIEC870_COT_UNUSED	Is not used
eIEC870_COT_CYCLIC	Cyclic data
eIEC870_COT_BACKGROUND	Background scan
eIEC870_COT_SPONTAN	Spontaneous data
eIEC870_COT_INIT	End of initialization
eIEC870_COT_REQ	Read request
eIEC870_COT_ACT	Command activation
eIEC870_COT_ACT_CON	Confirmation of command activation
eIEC870_COT_DEACT	Command abortion
eIEC870_COT_DEACT_CON	Confirmation of command abortion
eIEC870_COT_ACT_TERM	Termination of command activation
eIEC870_COT_RETREM	Response due to remote command
eIEC870_COT_RETLOC	Response due to local command
eIEC870_COT_FILE	File access
eIEC870_COT_INROGEN	Station interrogation (general)
eIEC870_COT_INRO1	Station interrogation for group 1
eIEC870_COT_INRO2	Station interrogation for group 2
eIEC870_COT_INRO3	Station interrogation for group 3
eIEC870_COT_INRO4	Station interrogation for group 4
eIEC870_COT_INRO5	Station interrogation for group 5
eIEC870_COT_INRO6	Station interrogation for group 6
eIEC870_COT_INRO7	Station interrogation for group 7
eIEC870_COT_INRO8	Station interrogation for group 8
eIEC870_COT_INRO9	Station interrogation for group 9
eIEC870_COT_INRO10	Station interrogation for group 10
eIEC870_COT_INRO11	Station interrogation for group 11
eIEC870_COT_INRO12	Station interrogation for group 12
eIEC870_COT_INRO13	Station interrogation for group 13
eIEC870_COT_INRO14	Station interrogation for group 14
eIEC870_COT_INRO15	Station interrogation for group 15
eIEC870_COT_INRO16	Station interrogation for group 16
eIEC870_COT_REQCOGEN	Counter interrogation (general)
eIEC870_COT_REQCO1	Counter interrogation for group 1
eIEC870_COT_REQCO2	Counter interrogation for group 2
eIEC870_COT_REQCO3	Counter interrogation for group 3
eIEC870_COT_REQCO4	Counter interrogation for group 4
eIEC870_COT_UNKNOWN_TYPE	Unknown type
eIEC870_COT_UNKNOWN_CAUSE	Unknown cause of transfer
eIEC870_COT_UNKNOWN_ASDU_ADDRESS	Unknown common ASDU address
eIEC870_COT_UNKNOWN_OBJECT_ADDRESS	Unknown object address
eIEC870_COT_14..eIEC870_COT_19 eIEC870_COT_42..eIEC870_COT_43 eIEC870_COT_48..eIEC870_COT_63	Reserved/unused range.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.36 E_IEC870_5_102COTType

Causes of transfer (monitoring direction and/or control direction) according to IEC 60870-5-102.

Syntax

```
TYPE E_IEC870_5_102COTType:
(
     eIEC870_2COT_0          := 0,
     eIEC870_2COT_1          := 1,
     eIEC870_2COT_2          := 2,
     eIEC870_2COT_SPONTAN    := 3, (* spontanous *)
     eIEC870_2COT_INIT       := 4, (* initialization *)
     eIEC870_2COT_REQ        := 5, (* request or requested *)
     eIEC870_2COT_ACT        := 6, (* command activation *)
     eIEC870_2COT_ACT_CON    := 7, (* command activation confirmation *)
     eIEC870_2COT_DEACT      := 8, (* command deactivation *)
     eIEC870_2COT_DEACT_CON  := 9, (* command deactivation confirmation *)
     eIEC870_2COT_ACT_TERM   := 10, (* command activation termination *)
     eIEC870_2COT_11         := 11,
     eIEC870_2COT_12         := 12,
     eIEC870_2COT_RECORD_NOT_FOUND := 13, (* requested record list is not available *)
     eIEC870_2COT_UNKNOWN_ASDU_TYPE := 14, (* unknown ASDU type idetifier *)
     eIEC870_2COT_UNKNOWN_RECORD_NUMBER := 15, (* uknown record number *)
     eIEC870_2COT_UNKNOWN_RECORD_ADDRESS := 16, (* unknown record address *)
     eIEC870_2COT_OBJECT_NOT_FOUND := 17, (* information object not available *)
     eIEC870_2COT_PERIOD_NOT_AVAILABLE := 18, (* requested measurement period not available *)
     eIEC870_2COT_19 := 19,
     eIEC870_2COT_20 := 20,
     eIEC870_2COT_21 := 21,
     eIEC870_2COT_22 := 22,
     eIEC870_2COT_23 := 23,
     eIEC870_2COT_24 := 24,
     eIEC870_2COT_25 := 25,
     eIEC870_2COT_26 := 26,
     eIEC870_2COT_27 := 27,
     eIEC870_2COT_28 := 28,
     eIEC870_2COT_29 := 29,
     eIEC870_2COT_30 := 30,
     eIEC870_2COT_31 := 31,
     eIEC870_2COT_32 := 32,
     eIEC870_2COT_33 := 33,
     eIEC870_2COT_34 := 34,
     eIEC870_2COT_35 := 35,
     eIEC870_2COT_36 := 36,
     eIEC870_2COT_37 := 37,
     eIEC870_2COT_38 := 38,
     eIEC870_2COT_39 := 39,
     eIEC870_2COT_40 := 40,
     eIEC870_2COT_41 := 41,
     eIEC870_2COT_42 := 42,
     eIEC870_2COT_43 := 43,
     eIEC870_2COT_44 := 44,
     eIEC870_2COT_45 := 45,
     eIEC870_2COT_46 := 46,
     eIEC870_2COT_47 := 47,
     eIEC870_2COT_48 := 48,
     eIEC870_2COT_49 := 49,
     eIEC870_2COT_50 := 50,
     eIEC870_2COT_51 := 51,
     eIEC870_2COT_52 := 52,
     eIEC870_2COT_53 := 53,
```

```
    eIEC870_2COT_54 := 54,
    eIEC870_2COT_55 := 55,
    eIEC870_2COT_56 := 56,
    eIEC870_2COT_57 := 57,
    eIEC870_2COT_58 := 58,
    eIEC870_2COT_59 := 59,
    eIEC870_2COT_60 := 60,
    eIEC870_2COT_61 := 61,
    eIEC870_2COT_62 := 62,
    eIEC870_2COT_63 := 63
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.37 E_IEC870_5_103MCOT

Causes of transfer in monitoring direction (slave -> master) according to IEC 60870-5-103.

Syntax

```
TYPE E_IEC870_5_103MCOT:
(
    eIEC870_MCOT_UNUSED := 0, (* Not used *)
    eIEC870_MCOT_SPONTAN := 1, (* Spontanous data *)
    eIEC870_MCOT_CYCLIC := 2, (* Cyclic data *)
    eIEC870_MCOT_FCB := 3, (* Reset FCB bit *)
    eIEC870_MCOT_CU := 4, (* Reset communication unit *)
    eIEC870_MCOT_SR := 5, (* Start/Restart *)
    eIEC870_MCOT_ON := 6, (* Power on *)
    eIEC870_MCOT_TST := 7, (* Test mode *)
    eIEC870_MCOT_SYN := 8, (* Time synchronisation *)
    eIEC870_MCOT_GI := 9, (* General interrogation *)
    eIEC870_MCOT_TGI := 10,(* Termination of general interrogation *)
    eIEC870_MCOT_LO := 11,(* Local operation *)
    eIEC870_MCOT_RO := 12,(* Remote operation *)
    eIEC870_MCOT_CP := 20,(* Positive ack of command *)
    eIEC870_MCOT_CN := 21,(* Negative ack of command *)
    eIEC870_MCOT_TOV := 31,(* Transmission of disturbance values *)
    eIEC870_MCOT_WP := 40,(* Positive ack of generic write command *)
    eIEC870_MCOT_WN := 41,(* Negative ack of generic write command *)
    eIEC870_MCOT_RP := 42,(* Valid data response to genric read command *)
    eIEC870_MCOT_RN := 43,(* Invalid data response to generic read command *)
    eIEC870_MCOT_CWC := 44 (* Confirmation of generic write *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.38 E_IEC870_5_103CCOT

Causes of transfer in control direction (master -> slave) according to IEC 60870-5-103.

Syntax

```
TYPE E_IEC870_5_103CCOT:
(
    eIEC870_CCOT_UNUSED := 0, (* Not used *)
    eIEC870_CCOT_SYN := 8, (* Time synchronisation *)
    eIEC870_CCOT_IGI := 9, (* Initialisation of general interrogation *)
    eIEC870_CCOT_GRC := 20, (* General command *)
    eIEC870_CCOT_TOV := 31, (* Transmission of disturbance values *)
    eIEC870_CCOT_WC := 40, (* Generic write command *)
    eIEC870_CCOT_RC := 42 (* Generic read command *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.39 E_IEC870_5_101FifoDbgFlags

Debugging output flags. If debugging output is activated, the ASDU hex bytes are logged in the TwinCAT XAE "Error List" window.

Syntax

```
TYPE E_IEC870_5_101FifoDbgFlags:
(
    eIEC870_FIFO_DBG_OFF := 0,
    eIEC870_FIFO_DBG_PUT := 1,
    eIEC870_FIFO_DBG_GET := 2,
    eIEC870_FIFO_DBG_ALL := 3
);
END_TYPE
```



Please note that the debugging output results in higher system load.

Value	Description
eIEC870_FIFO_DBG_OFF	No debugging output
eIEC870_FIFO_DBG_PUT	Debugging output when FIFO elements are added
eIEC870_FIFO_DBG_GET	Debugging output when FIFO elements are removed
eIEC870_FIFO_DBG_ALL	Debugging output when FIFO elements are added or removed

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.40 E_IEC870_5_101AODBType

Application database type. Currently only the type eIEC870_AODB_STATIC is supported. I.e. all application objects (data points) have to be configured by the PLC application.

Syntax

```
TYPE E_IEC870_5_101AODBType :
(
    eIEC870_AODB_STATIC := 0,
    eIEC870_AODB_DYNAMIC := 1
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.41 E_IEC870_5_101InitSeqStep

Selection of commands that a central station (only high-level) should send to the substation after establishing the communication link and/or after receiving the ASDU M_EI_NA_1 (end of initialization). Several of these commands can be configured to form an initialization sequence (successive initialization steps). The entire configuration of the cyclic data acquisition (including the initialization sequence) is carried out via the components of the structured type: ST_IEC870_5_101AcquisitionParams [▶ 593].

The initialization sequence is mainly configured in the ST_IEC870_5_101AcquisitionParams.arrInitSeq [▶ 594] component. However, some of the commands require additional configuration parameters. These parameters are then located in other ST_IEC870_5_101AcquisitionParams [▶ 593] components.
Samples:

- eIEC870_ISTEP_GENRO command uses by default the parameter value: ST_IEC870_5_101AcquisitionParams.arrGenro [▶ 594][0].QOI := eIEC870_QOI_INROGEN.
- eIEC870_ISTEP_GENRO10 command uses by default the parameter value ST_IEC870_5_101AcquisitionParams.arrGenro [▶ 594][10].QOI := eIEC870_QOI_INRO10.
- eIEC870_ISTEP_CORO command uses by default the parameter values: ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594][0].eRQT := eIEC870_RQT_REQCOGEN and ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594][0].eFRZ := eIEC870_FRZ_READ.
- eIEC870_ISTEP_CORO1 command uses by default the parameter values: ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594][1].eRQT := eIEC870_RQT_REQCO1 and ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594][1].eFRZ := eIEC870_FRZ_READ.

- eIEC870_ISTEP_DELAY command uses by default the parameter value: ST_IEC870_5_101AcquisitionParams.delayCmd [▶ 594].tDelay := T#5S.

From Tc2_IEC60870_5_10x.compiled-library v3.0.7.0, the trigger/time of processing the initialization sequence can be configured via the ST_IEC870_5_101AcquisitionParams.eMode [▶ 594] component. By default, processing of the initialization sequence is started after "STARTDT" (IEC 60870-5-104) or "Reset of link" (IEC 60870-5-101).

The central station function blocks (high-level) already define an input variable: FB_IEC870_5_101Master [▶ 488].acqPara or FB_IEC870_5_104Master [▶ 492].acqPara of type: ST_IEC870_5_101AcquisitionParams. The low-level function blocks offer more control over cyclic data acquisition. However, the low-level function blocks require more programming effort.

Syntax

```
TYPE E_IEC870_5_101InitSeqStep :
(
    eIEC870_ISTEP_UNUSED  := 0,
    eIEC870_ISTEP_CLOCK   := 1,
    eIEC870_ISTEP_TEST    := 2,
    eIEC870_ISTEP_GENRO   := 3,
    eIEC870_ISTEP_CORO    := 4,
    eIEC870_ISTEP_COMMAND := 5,
    eIEC870_ISTEP_DELAY   := 6,

    eIEC870_ISTEP_CORO1   := 7,
    eIEC870_ISTEP_CORO2   := 8,
    eIEC870_ISTEP_CORO3   := 9,
    eIEC870_ISTEP_CORO4   := 10,

    eIEC870_ISTEP_GENRO1  := 11,
    eIEC870_ISTEP_GENRO2  := 12,
    eIEC870_ISTEP_GENRO3  := 13,
    eIEC870_ISTEP_GENRO4  := 14,
    eIEC870_ISTEP_GENRO5  := 15,
    eIEC870_ISTEP_GENRO6  := 16,
    eIEC870_ISTEP_GENRO7  := 17,
    eIEC870_ISTEP_GENRO8  := 18,
    eIEC870_ISTEP_GENRO9  := 19,
    eIEC870_ISTEP_GENRO10 := 20,
    eIEC870_ISTEP_GENRO11 := 21,
    eIEC870_ISTEP_GENRO12 := 22,
    eIEC870_ISTEP_GENRO13 := 23,
    eIEC870_ISTEP_GENRO14 := 24,
    eIEC870_ISTEP_GENRO15 := 25,
    eIEC870_ISTEP_GENRO16 := 26
);
END_TYPE
```

Values

Name	Meaning
eIEC870_ISTEP_UNUSED	The initialization step is not used.
eIEC870_ISTEP_CLOCK	The central station should send a clock synchronization command to the substation.
eIEC870_ISTEP_TEST	The central station should send a test command to the substation.
eIEC870_ISTEP_GENRO	The central station should send a general interrogation command (General group) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro [▶ 594][0].
eIEC870_ISTEP_CORO	The central station should send a counter interrogation command (General group) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrCoro [▶ 594][0].
eIEC870_ISTEP_COMMAND	The central station should send all commands to the substation once (data points in control direction: single commands, double commands, setpoints, etc.).
eIEC870_ISTEP_DELAY	The central station is to wait before executing the next initialization step. The wait time parameter for this command is located in: ST_IEC870_5_101AcquisitionParams.delayCmd [▶ 594].
eIEC870_ISTEP_CORO1	The central station should send a counter interrogation command (group 1) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrCoro[1].
eIEC870_ISTEP_CORO2	The central station should send a counter interrogation command (group 2) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrCoro[2].
eIEC870_ISTEP_CORO3	The central station should send a counter interrogation command (group 3) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrCoro[3].
eIEC870_ISTEP_CORO4	The central station should send a counter interrogation command (group 4) to the substation. Further configuration parameters for this command can be found in:ST_IEC870_5_101AcquisitionParams.arrCoro[4].
eIEC870_ISTEP_GENRO1	The central station should send a general interrogation command (group 1) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[1].
eIEC870_ISTEP_GENRO2	The central station should send a general interrogation command (group 2) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[2].
eIEC870_ISTEP_GENRO3	The central station should send a general interrogation command (group 3) to the substation. Further configuration parameters for this command can be found in:ST_IEC870_5_101AcquisitionParams.arrCoro[3].
eIEC870_ISTEP_GENRO4	The central station should send a general interrogation command (group 4) to the substation. Further configuration parameters for this command can be found in:ST_IEC870_5_101AcquisitionParams.arrGenro[4].
eIEC870_ISTEP_GENRO5	The central station should send a general interrogation command (group 5) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[5].

Name	Meaning
eIEC870_ISTEP_GENRO6	The central station should send a general interrogation command (group 6) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[6].
eIEC870_ISTEP_GENRO7	The central station should send a general interrogation command (group 7) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[7].
eIEC870_ISTEP_GENRO8	The central station should send a general interrogation command (group 8) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[8].
eIEC870_ISTEP_GENRO9	The central station should send a general interrogation command (group 9) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[9].
eIEC870_ISTEP_GENRO10	The central station should send a general interrogation command (group 10) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[10].
eIEC870_ISTEP_GENRO11	The central station should send a general interrogation command (group 11) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[11].
eIEC870_ISTEP_GENRO12	The central station should send a general interrogation command (group 12) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[12].
eIEC870_ISTEP_GENRO13	The central station should send a general interrogation command (group 13) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[13].
eIEC870_ISTEP_GENRO14	The central station should send a general interrogation command (group 14) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[14].
eIEC870_ISTEP_GENRO15	The central station should send a general interrogation command (group 15) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[15].
eIEC870_ISTEP_GENRO16	The central station should send a general interrogation command (group 16) to the substation. Further configuration parameters for this command can be found in : ST_IEC870_5_101AcquisitionParams.arrGenro[16].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.42 E_IEC870_5_101FBufferState

Current offline ASDU file buffer status.

Syntax

```
TYPE E_IEC870_5_101QU :
(
    eIEC870_FBUFFER_IDLE := 0,
    eIEC870_FBUFFER_SAVING := 1,
    eIEC870_FBUFFER_LOADING := 2,
    eIEC870_FBUFFER_ERROR := 3
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.43 E_IEC870_5_101SCS

Single command state.

Syntax

```
TYPE E_IEC870_5_101SCS :
(
    eIEC870_SCS_OFF := 0,
    eIEC870_SCS_ON  := 1
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.44 E_IEC870_5_101DCS

Double command state.

Syntax

```
TYPE E_IEC870_5_101DCS :
(
    eIEC870_DCS_INDETERMINATE0  := 0,
    eIEC870_DCS_OFF             := 1,
    eIEC870_DCS_ON              := 2,
    eIEC870_DCS_INDETERMINATE3  := 3
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.45 E_IEC870_5_101COI

Cause of initialization.

Syntax

```
TYPE E_IEC870_5_101COI :
(
    eIEC870_COI_LOCAL_POWER_ON      := 0,(* Local power ON *)
    eIEC870_COI_LOCAL_MANUAL_RESET  := 1,(* Local manual reset *)
    eIEC870_COI_REMOTE_RESET        := 2 (* Remote reset *)
    (* <3..31> := Reserved for future norm definitions *)
    (* <32..127> := Reserved for user definitions (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.46 E_IEC870_5_101QOI

Qualifier of interrogation command.

Syntax

```
TYPE E_IEC870_5_101QOI :
(
    eIEC870_QOI_UNUSED      := 0, (* Not used*)
    (* <1..19> := Reserved for standard definitions of this companion standard *)
    eIEC870_QOI_INROGEN     := 20, (* Global station interrogation *)
    eIEC870_QOI_INRO1       := 21, (* Group 1 station interrogation *)
    eIEC870_QOI_INRO2       := 22, (* Group 2 station interrogation...*)
    eIEC870_QOI_INRO3       := 23,
    eIEC870_QOI_INRO4       := 24,
    eIEC870_QOI_INRO5       := 25,
    eIEC870_QOI_INRO6       := 26,
    eIEC870_QOI_INRO7       := 27,
    eIEC870_QOI_INRO8       := 28,
    eIEC870_QOI_INRO9       := 29,
    eIEC870_QOI_INRO10      := 30,
    eIEC870_QOI_INRO11      := 31,
    eIEC870_QOI_INRO12      := 32,
    eIEC870_QOI_INRO13      := 33,
    eIEC870_QOI_INRO14      := 34,
    eIEC870_QOI_INRO15      := 35,
    eIEC870_QOI_INRO16      := 36 (* Group 16 station interrogation*)
    (* <37..63> := Reserved for future norm definitions of this companion standard (compatible range
```

```
) *)
    (* <64..255> := Reserved for user definitions (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.47 E_IEC870_5_101QL

Qualifier of set-point command.

Syntax

```
TYPE E_IEC870_5_101QL :
(
    eIEC870_QL_DEFAULT := 0
   (* <1..63> := Reserved for standard definitions (compatible range) *)
   (* <64..127> := Reserved for special use (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.48 E_IEC870_5_101FRZ

Freeze/reset qualifier of counter interrogation command.

Syntax

```
TYPE E_IEC870_5_101FRZ :
(
    eIEC870_FRZ_READ        := 0, (* Read only (no freeze or reset) *)
    eIEC870_FRZ_FREEZE      := 1, (* Counter freeze without reset (value frozen represents integrate
d total) *)
    eIEC870_FRZ_FREEZE_AND_RESET := 2, (* Counter freeze with reset (value frozen represents increme
ntal information) *)
    eIEC870_FRZ_RESET       := 3 (* Counter reset *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.49 E_IEC870_5_101RQT

Request qualifier of counter interrogation command.

Syntax

```
TYPE E_IEC870_5_101RQT :
(
    eIEC870_RQT_NONE        := 0,(* No counter read *)
    eIEC870_RQT_REQCO1      := 1,(* Group 1 counter interrogation *)
    eIEC870_RQT_REQCO2      := 2,(* Group 2 counter interrogation *)
    eIEC870_RQT_REQCO3      := 3,(* Group 3 counter interrogation *)
    eIEC870_RQT_REQCO4      := 4,(* Group 4 counter interrogation *)
    eIEC870_RQT_REQCOGEN    := 5 (* General counter interrogation *)
    (* <6..31> := Reserved for future norm definitions *)
    (* <32..63> := Reserved for user definitions (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.50 E_IEC870_5_101QRP

Qualifier for the process reset command.

Syntax

```
TYPE E_IEC870_5_101QRP :
(
    eIEC870_QRP_UNUSED      := 0,(* Not used *)
    eIEC870_QRP_GENERAL     := 1,(* General process reset *)
    eIEC870_QRP_TTEVENTS    := 2 (* Reset pending events with time tag *)
    (* <3..127> := Reserved for future norm definitions *)
    (* <128..255> := Reserved for user definitions (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.51 E_IEC870_5_101QU

Qualifier for the commands.

Syntax

```
TYPE E_IEC870_5_101QU :
(
    eIEC870_QU_UNSPECIFIED  := 0,
    eIEC870_QU_SHORTPULSE   := 1,
    eIEC870_QU_LONGPULSE    := 2,
    eIEC870_QU_PERSISTENT   := 3
    (* <4..8> := Reserved for standrad definitions of companion standard (compatible range) *)
    (* <9..15> := Reserved for the selection of other predefined functions *)
    (* <16..31> := Reserved for special use (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.52 E_IEC870_5_101ES

Event state (single event of protection equipment).

Syntax

```
TYPE E_IEC870_5_101ES:
(
    eIEC870_ES_INDETERMINATE0 := 0,
    eIEC870_ES_OFF            := 1,
    eIEC870_ES_ON             := 2,
    eIEC870_ES_INDETERMINATE3 := 3
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.53 E_IEC870_5_101KPA

Kind of parameter.

Syntax

```
TYPE E_IEC870_5_101KPA:
(
    eIEC870_KPA_UNUSED := 0,
    eIEC870_KPA_THRESH := 1,
    eIEC870_KPA_FILTER := 2,
    eIEC870_KPA_LOLIMIT := 3,
    eIEC870_KPA_HILIMIT := 4
   (* <5..31> := Reserved for standard definitions (compatible range) *)
   (* <32..63> := Reserved for special use (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.54 E_IEC870_5_101QPA

Qualifier of the parameter activation.

Syntax

```
TYPE E_IEC870_5_101QPA:
(
    eIEC870_QPA_UNUSED := 0,
    eIEC870_QPA_GENERAL := 1,
    eIEC870_QPA_OBJECT := 2,
    eIEC870_QPA_TRANSMISSION := 3
    (* <4..127> := Reserved for standard definitions (compatible range) *)
    (* <128..255> := Reserved for special use (private range) *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.55 E_IEC870_5_101RCS

Status information of the step command.

Syntax

```
TYPE E_IEC870_5_101RCS :
(
    eIEC870_RCS_NOTALLOWED0 := 0,
    eIEC870_RCS_DECREMENT   := 1,
    eIEC870_RCS_INCREMENT   := 2,
    eIEC870_RCS_NOTALLOWED3 := 3
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.56 E_IEC870_5_101SPI

Single-point information.

Syntax

```
TYPE E_IEC870_5_101SPI :
(
    eIEC870_SPI_OFF := 0,
    eIEC870_SPI_ON  := 1
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.57 E_IEC870_5_101DPI

Double-point information.

Syntax

```
TYPE E_IEC870_5_101DPI :
(
    eIEC870_DPI_INDETERMINATE0  := 0,
    eIEC870_DPI_OFF             := 1,
    eIEC870_DPI_ON              := 2,
    eIEC870_DPI_INDETERMINATE3  := 3
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.58 E_IEC870_5_101FrameType

IEC 60870-5-10x Frame Type. Currently only FT1.2 is supported.

Syntax

```
TYPE E_IEC870_5_101FrameType :
(
    eIEC870_FrameType_FT1_2
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.59 E_IEC870_5_101LinkMode

Configures the data transfer mode.

Syntax

```
TYPE E_IEC870_5_101LinkMode :
(
    eIEC870_LinkMode_Unbalanced,
    eIEC870_LinkMode_Balanced
);
END_TYPE
```

eIEC870_LinkMode_Unbalanced: Asymmetric data transfer. In this transfer mode the control station (master) controls the data exchange with the substations (slaves). The data exchange is always initiated by the control station. In this way, the control station sequentially queries all substations.

eIEC870_LinkMode_Balanced: Symmetric data transfer. In this transfer mode each station can initiate the data exchange. It can only be used in "point-to-point" and "multiple-point-to-point" configurations.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.60 E_IEC870_5_101SerialLinkState

IEC 60870-5-101/102/103 connection status.

Syntax

```
TYPE E_IEC870_5_101SerialLinkState :
(
    eSERIALLINK_DISCONNECTED,
    eSERIALLINK_ESTABLISHED,
    eSERIALLINK_SUSPENDED
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.61 E_IEC870_5_101PartyLineMode

Party line activation mode. See also description of the function block: FB_IEC870_PartyLineCtrl [▶ 515].

Syntax

```
TYPE E_IEC870_5_101PartyLineMode :
(
    eIEC870_PartylineMode_Off := 0, (* Deaktiviert (wird nicht verwendet) *)
    eIEC870_PartylineMode_On := 1,(* Aktiviert *)
    eIEC870_PartylineMode_Ext_On := 2 (* Aktiviert, verwendet "Extended Ctrl/Status" Prozessdaten *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.62 E_IEC870_5_101LinkReset

Reset type during initialization of the link layer.

Syntax

```
TYPE E_IEC870_5_101LinkReset :
(
    eIEC870_LinkReset_None  := 0,(* Disabled *)
    eIEC870_LinkReset_CU    := 1,(* Reset communication unit *)
    eIEC870_LinkReset_UP    := 2,(* Reset user process *)
    eIEC870_LinkReset_FCB   := 3 (* Reset FCB bit *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.63 E_IEC870_DEVICE_TYPE

Protocol or station type.

Syntax

```
TYPE E_IEC870_DEVICE_TYPE :
(
    eIEC870_101_SLAVE := 0, (* Secondary (responding) station *)
    eIEC870_101_MASTER := 1,(* Primary (initiating) station *)

    eIEC870_102_SLAVE := 2, (* Secondary (responding) station *)
    eIEC870_102_MASTER := 3,(* Primary (initiating) station *)

    eIEC870_103_SLAVE := 4, (* Secondary (responding) station *)
    eIEC870_103_MASTER := 5 (* Primary (initiating) station *)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.64 E_IEC870_5_104DataTransferState

Status of the IEC 60870-5-104 data transfer.

Syntax

```
TYPE E_IEC870_5_104DataTransferState :
(
    eIEC870_STOPDT, (* data exchange deactivated *)
    eIEC870_STARTDT, (* data exchange activated *)
    eIEC870_STOPDT_PENDING, (* waiting for STOPDT confirmation (master only)*)
    eIEC870_STARTDT_PENDING (* waiting for STARTDT confirmation (master only)*)
);
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.65 T_HSERIALCTRL

A variable of this type represents a connection handle of the serial interface. The structural elements should not be written to or modified directly. Variables of this type are used for exchanging the data to be sent or received between the function block FB_IEC870_SerialLineCtrl [▶ 519] and other IEC 60870-5-101 PLC blocks.

Syntax

```
TYPE T_HSERIALCTRL
STRUCT
(*…*)
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.66 T_HAODBTable

Application object database handle (hash table handle). The table handle must be initialized once with the function F_iecCreateTableHnd [▶ 554] before it can be used.

Syntax

```
TYPE T_HAODBTable :
STRUCT
(*...*)
END_STRUCT
END_TYPE
```

The variables of this structured type are not accessed directly, but only using the available functions or function blocks. Use the functions listed in the table:

Function	Description
F_iecCreateTableHnd [▶ 554]	Initializes the hash table handle
F_iecAddTableEntry [▶ 555]	Configures and adds a new hash table entry
F_iecRemoveTableEntry [▶ 562]	Removes a hash table entry
F_iecLookupTableEntry [▶ 560]	Checks whether a certain hash table entry exists
F_iecGetPosOfTableEntry [▶ 558]	Determines the linear position of a hash table entry

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.67 T_CP56Time2a

Binary (7 byte) time format (type a).

Syntax

```
TYPE T_CP56Time2a :
STRUCT
    Milliseconds    : WORD;
    IVResMinute     : BYTE;
    SURes2Hour      : BYTE;
    DOWDay          : BYTE;
    Res3Month       : BYTE;
    Res4Year        : BYTE;
END_STRUCT
END_TYPE
```

Milliseconds: Milliseconds := <0..59999>.

IVResMinute: Bit 7 = IV := <0=valid, 1=invalid time>, bit 6 = res (reserved) := <0>, bit 0..5 = minute := <0..59>.

SURes2Hour: Bit 7 = SU := <0=standard time, 1=summer time>, bits 5..6 = res2 (reserved) := <0>, bits 0..4 = hour := <0..23>.

DOWDay : Bits 5..7 = DOW (day of the week) := <0=not used, 1=Monday..7=Sunday>, bits 0..4 = day (day of the month) := <1..31>.

Res3Month: Bits 4..7 = res3 (reserved) := <0>, bits 0..3 = month := <1..12>.

Res4Year: Bit 7 = res4 (reserved) := <0>, bits 0..6 = year := <0..99>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.68 T_CP56Time2b

Binary (7 byte) time format (type b).

Syntax

```
TYPE T_CP56Time2b :
STRUCT
    MsSecond    : WORD := 16#00;
    IVTisMinute : BYTE := 16#00;
    SURes1Hour  : BYTE := 16#00;
    DOWDay      : BYTE := 16#01;
    PtiEtiMonth : BYTE := 16#01;
    Res2Year    : BYTE := IEC870_DEFAULT_CP56TIME2A_YEAR;
END_STRUCT
END_TYPE
```

MsSecond: Bits 11..15 = second := <0..59>, bits 0..9 = ms (millisecond) := <0..999>.

IVTisMinute: Bit 7 = IV := <0=valid, 1=invalid time>, bit 6 = Tif (tariff information) := <0..1>, bits 0..5 = minute := <0..59>.

SURes1Hour: Bit 7 = SU := <0=standard time, 1=summer time>, bits 5..6 = res1 (reserved) := <0>, bits 0..4 = hour := <0..23>.

DOWDay : Bits 5..7 = DOW (day of the week) := <0=not used, 1=Monday..7=Sunday>, bits 0..4 = day (day of the month) := <1..31>.

PtiEtiMonth: Bits 6..7 = Pti (power tariff information) := <0..3>, bits 4..5 = Eti (energy tariff information) := <0..3>, bits 0..3 = month := <1..12>.

Res2Year: Bit 7 = res2 (reserved) := <0>, bits 0..6 = year := <0..99>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.69 T_CP40Time2a

Binary (5 byte) time format.

Syntax

```
TYPE T_CP40Time2a :
STRUCT
    IVTisMinute : BYTE := 16#00;
    SURes1Hour  : BYTE := 16#00;
    DOWDay      : BYTE := 16#01;
    PtiEtiMonth : BYTE := 16#01;
    Res2Year    : BYTE := IEC870_DEFAULT_CP56TIME2A_YEAR;
END_STRUCT
END_TYPE
```

IVTisMinute: Bit 7 = IV := <0=valid, 1=invalid time>, bit 6 = Tif (tariff information) := <0..1>, bits 0..5 = minute := <0..59>.

SURes1Hour: Bit 7 = SU := <0=standard time, 1=summer time>, bits 5..6 = res1 (reserved) := <0>, bits 0..4 = hour := <0..23>.

DOWDay : Bits 5..7 = DOW (day of the week) := <0=not used, 1=Monday..7=Sunday>, bits 0..4 = day (day of the month) := <1..31>.

PtiEtiMonth: Bits 6..7 = Pti (power tariff information) := <0..3>, bits 4..5 = Eti (energy tariff information) := <0..3>, bits 0..3 = month := <1..12>.

Res2Year: Bit 7 = res2 (reserved) := <0>, bits 0..6 = year := <0..99>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.70 T_CP32Time2a

Binary (4 byte) time format.

Syntax

```
TYPE T_CP32Time2a :
STRUCT
    Milliseconds    : WORD;
    IVResMinute     : BYTE;
```

```
    SURes2Hour      : BYTE;
END_STRUCT
END_TYPE
```

Milliseconds: Milliseconds := <0..59999>.

IVResMinute: Bit 7 = IV := <0=valid, 1=invalid time>, bit 6 = res (reserved) := <0>, bit 0..5 = minute := <0..59>.

SURes2Hour: Bit 7 = SU := <0=standard time, 1=summer time>, bits 5..6 = res2 (reserved) := <0>, bits 0..4 = hour := <0..23>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.71 T_CP24Time2a

Binary (3 byte) time format.

Syntax

```
TYPE T_CP24Time2a :
STRUCT
    Milliseconds    : WORD;
    IVResMinute     : BYTE;
END_STRUCT
END_TYPE
```

Milliseconds: Milliseconds := <0..59999>.

IVResMinute: Bit 7 = IV := <0=valid, 1=invalid time>, bit 6 = res (reserved) := <0>, bit 0..5 = minute := <0..59>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.72 T_CP16Time2a

Binary (2 byte) time format.

Syntax

```
TYPE T_CP16Time2a :
STRUCT
    Milliseconds    : WORD;
END_STRUCT
END_TYPE
```

Milliseconds: Milliseconds := <0..59999>.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.73 T_CP24IOA

Structured TwinCAT object address.

Syntax

```
TYPE T_CP24IOA:
STRUCT
    ioMap      : E_IEC870_5_101IOMappingType := MAP_AREA_NONE;
    byteOffs   : WORD := 0;
    bitOffs    : BYTE(0..7) := 0;
END_STRUCT
END_TYPE
```

ioMapType: TwinCAT PLC process data area [▶ 608]. This parameter defines the mapping for the TwinCAT PLC and IEC application object process data.

byteOffs: TwinCAT PLC process data byte offset. Value range: 0..65535.

bitOffs: TwinCAT PLC process data bit offset. Value range: 0..7.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.74 T_IEC870_5_101COTBits

Cause of transfer bit mask.

Syntax

```
TYPE T_IEC870_5_101COTBits :  ARRAY[0..7] OF BYTE;
END_TYPE
```

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.75 ST_IEC870_5_101FBufferCfg

Configuration settings for the offline ASDU file buffer. This structure is used by the FB_IEC870_5_101FBufferCtrl [▶ 502] function block.

Syntax

```
TYPE ST_IEC870_5_101FBufferCfg :
STRUCT
    sNetID          : T_AmsNetID := '';
    sPathName       : T_MaxString := 'c:\Temp\data.dat';
    ePath           : E_OpenPath := PATH_GENERIC;
    cbBuffer        : UDINT := 16#100000;
    bOverwrite      : BOOL := TRUE;
    bFilter         : BOOL := FALSE;
    cotFilter       : T_IEC870_5_101COTBits := 8(0);
    tTimeout        : TIME := DEFAULT_ADS_TIMEOUT;
    bContinueAfterOverflow    : BOOL := FALSE;
    bResetTxBufferOnOverwrite : BOOL := FALSE;
END_STRUCT
END_TYPE
```

sNetID: The network address of TwinCAT computer, on which the buffer file is to be created, can be specified here. If it is to be run on the local computer, an empty string can be entered.

sPathName: Contains the path and file name of the file to be opened (max. 255 characters). The path can only point to the local file system of the computer. This means that network paths cannot be used here.

ePath: This input can be used to select a TwinCAT system file path on the target device for opening the file.

cbBuffer: Max. byte size of the buffer file. An error is returned if bOverwrite = FALSE and the maximum size was exceeded. Default setting: 16#100000 = 1MB.

bOverwrite: When the maximum size is reached, the oldest entries are overwritten, if this variable was set to TRUE.

bFilter: Currently not implemented. Activates/deactivates a COT filter. Only ASDUs with certain causes of transfer are cached in the file.

cotFilter: Currently not implemented. This variable can be used to configure causes of transfer (COT filter [▶ 637]), which are to be cached in the file.

tTimeout: Maximum timeout time, which should not be exceeded during file access.

bContinueAfterOverflow: TRUE => ignore file overrun errors when maximum file size is reached (only if bOverwrite = FALSE and TF6500 v3.0.6.0 or higher).

bResetTxBufferOnOverwrite: TRUE => reset the unsent Tx data in the memory buffer, once the first entry in the file buffer was overwritten (only if bOverwrite = TRUE and TF6500 v3.0.6.0 or higher).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.76 ST_IEC870_5_101FBufferStatus

Status information for the offline ASDU file buffer. This structure is used by the FB_IEC870_5_101FBufferCtrl [▶ 502] function block.

Syntax

```
TYPE ST_IEC870_5_101FBufferStatus:
STRUCT
    eState      : E_IEC870_5_101FBufferState := eIEC870_FBUFFER_IDLE;
    nErrID      : UDINT;
    bCorrupted  : BOOL;
    nCount      : UDINT;
    bOverwriting: BOOL;
    nOverwrites : UDINT;
    bOverflowing: BOOL;
END_STRUCT
END_TYPE
```

eState: Returns the file buffer status [▶ 622] (the file is written, read or closed, or a write/read error has occurred).

nErrID: Returns the ADS error number in the event of a file access error.

bCorrupted: If TRUE, the last opened file was identified as corrupt and replaced with new, empty file. File corruption may occur, if the maximum file buffer size was changed or the file was not closed properly, for example.

nCount: Current number of cached entries in the file buffer. The file buffer must first be opened, before the number of entries in an existing file can be counted. I.e. the connection must briefly switch to offline mode.

bOverwriting: TRUE => The oldest file entry is about to be overwritten (only for TF6500 v3.0.6.0 or higher).

nOverwrites: Number of file entries, which were overwritten (only for TF6500 v3.0.6.0 or higher).

bOverflowing: TRUE => Maximum file size was reached, or the last entry could not be written (only for TF6500 v3.0.6.0 or higher).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.77 ST_IEC870_5_101ProtocolParams

Protocol parameters of the link layer. The meaning of the parameters differs, depending on the protocol or station type.

Syntax

```
TYPE ST_IEC870_5_101ProtocolParams :
STRUCT
    eType         : E_IEC870_DEVICE_TYPE       := eIEC870_101_SLAVE;
    eLinkReset    : E_IEC870_5_101LinkReset    := eIEC870_LinkReset_CU;
    linkAddr      : DWORD                      := 1;
    eLinkAddrSize : E_IEC870_5_101LinkAddrSize := eIEC870_LinkAddr_TwoOctets;
    eLinkMode     : E_IEC870_5_101LinkMode     := eIEC870_LinkMode_Unbalanced;
    eFrameType    : E_IEC870_5_101FrameType    := eIEC870_FrameType_FT1_2;
    tRxTimeout    : TIME := T#5s;
    tTxTimeout    : TIME := T#5s;
    bForceC1Res   : BOOL := TRUE;
    bForceC2Res   : BOOL := TRUE;
    tClass1Poll   : TIME := T#200ms;
    tClass2Poll   : TIME := T#200ms;
    nRetries      : BYTE := 3;
    tRetry        : TIME := T#100ms;
    tResponse     : TIME := T#1s;
    tTestLink     : TIME := T#5s;
    tPollDFC      : TIME := T#1s;
    FRAMELength   : BYTE(MIN_IEC870_5_101Link_FRAMELEN..MAX_IEC870_5_101Link_FRAMELEN) := MAX_IEC870
_5_101Link_FRAMELEN;
    bRetainBuffer : BOOL := FALSE;
    tMaxPollDelay : TIME := T#0s;
    tLinkPollCycle: TIME := T#10s;
END_STRUCT
END_TYPE
```

The table below contains notes regarding the configuration. It the values are fixed, the parameter must match the value from the table (not configurable).

Key:

- **X** The parameter is used and is configurable;
- **N/A** The parameter is not used and is not configurable;

Parameter name	Initialization value	IEC 60870-5-101 controlled station (slave, unbalanced mode)	IEC 60870-5-101 control station (master, unbalanced mode)	IEC 60870-5-103 control station (master, unbalanced mode)	IEC 60870-5-102 control station (master, unbalanced mode)	Description
eType [▶ 631]	eIEC870_101_SLAVE	fixed, eIEC870_101_SLAVE	fixed, eIEC870_101_MASTER	fixed, eIEC870_103_MASTER	fixed, eIEC870_102_MASTER	Configures the protocol and station type.
eLinkReset [▶ 631]	eIEC870_LinkReset_CU	N/A	X	X	fixed, eIEC870_LinkReset_CU	Reset type during initialization of the link layer.
linkAddr	1	Own address	Address of the remote device	Address of the remote device	Address of the remote device	Connection address.
eLinkAddrSize [▶ 609]	eIEC870_LinkAddr_TwoOctets	X	X	X	X	Octet size of the connection address.
eLinkMode [▶ 629]	eIEC870_LinkMode_Unbalanced	fixed, eIEC870_LinkMode_Unbalanced	fixed, eIEC870_LinkMode_Unbalanced	fixed, eIEC870_LinkMode_Unbalanced	fixed, eIEC870_LinkMode_Unbalanced	Connection mode (balanced/ unbalanced).
eFrameType [▶ 629]	eIEC870_FrameType_FT1_2	fixed, eIEC870_FrameType_FT1_2	fixed, eIEC870_FrameType_FT1_2	fixed, eIEC870_FrameType_FT1_2	fixed, eIEC870_FrameType_FT1_2	Telegram frame format. Currently only the format F1.2 is supported.
tRxTimeout	T#5s	X	X	X	X	Max. frame receive timeout time.
tTxTimeout	T#5s	X	X	X	X	Max. frame send timeout time.

Parameter name	Initialization value	IEC 60870-5-101 controlled station (slave, unbalanced mode)	IEC 60870-5-101 control station (master, unbalanced mode)	IEC 60870-5-103 control station (master, unbalanced mode)	IEC 60870-5-102 control station (master, unbalanced mode)	Description
bForceC1Res	TRUE	X	N/A	N/A	N/A	If this parameter is set (TRUE), Class 1 data are sent as response to a Class 2 request, if Class 1 data are available (optimized, higher data throughput for Class 1). If this parameter is not set (FALSE), a negative response is sent to a Class 2 request, if Class 1 data are present (lower data throughput, empty telegrams).

Parameter name	Initialization value	IEC 60870-5-101 controlled station (slave, unbalanced mode)	IEC 60870-5-101 control station (master, unbalanced mode)	IEC 60870-5-103 control station (master, unbalanced mode)	IEC 60870-5-102 control station (master, unbalanced mode)	Description
bForceC2Res	TRUE	X	N/A	N/A	N/A	If this parameter is set (TRUE), Class 2 data are sent as response to a Class 2 request, even if Class 1 data are available (optimized, higher data throughput for Class 2). If this parameter is not set (FALSE), a negative response is sent to a Class 2 request, if Class 1 data are present (lower data throughput, empty telegrams)
tClass1Poll	T#200ms	N/A	X	X	X	Cycle time in which the Class 1 data are to be queried (unbalanced mode only).
tClass2Poll	T#200ms	N/A	X	X	X	Cycle time in which the Class 2 data are to be queried (unbalanced mode only).
nRetries	3	Not used up to v3.0.6 (error on first disturbed frame), configurable from v3.0.8 or higher	X	X	X	Maximum number of telegram retries when the data exchange fails.

Parameter name	Initialization value	IEC 60870-5-101 controlled station (slave, unbalanced mode)	IEC 60870-5-101 control station (master, unbalanced mode)	IEC 60870-5-103 control station (master, unbalanced mode)	IEC 60870-5-102 control station (master, unbalanced mode)	Description
tRetry	T#100ms	N/A	X	X	X	Delay time for telegram retries when the data exchange fails.
tResponse	T#1s	N/A	X	X	X	This parameter determines the maximum delay time for a telegram acknowledge ment. If this time is exceeded, the telegram is repeated up to *nRetry* times.
tTestLink	T#5s	N/A	N/A	N/A	N/A	Cycle time in which the test telegrams are to be sent (balanced mode only)
tPollDFC	T#1s	N/A	X	X	X	Cycle time in which the connection status is to be queried, if the DFC bit is set. The DFC bit is set by the remote device if further telegrams could cause an overflow.
FRAMELength	MAX_IEC870_5_101Link_FRAMELEN	X	X	X	X	Telegram length, max. length L.
bRetainBuffer	FALSE	from product version v3.0.2 or higher	X	X	X	If this parameter is set, the internal TX/RX buffers are not reset (deleted) on initialization of the link layer.

Parameter name	Initialization value	IEC 60870-5-101 controlled station (slave, unbalanced mode)	IEC 60870-5-101 control station (master, unbalanced mode)	IEC 60870-5-103 control station (master, unbalanced mode)	IEC 60870-5-102 control station (master, unbalanced mode)	Description
tMaxPollDelay	T#0s (= 0: deactivated, <> 0: active)	from product version v3.0.2 or higher	N/A	N/A	N/A	The connection status is declared OFFLINE, when this time is exceeded in the absence of poll queries.
tLinkPollCycle	T#10s	N/A	X	X	X	Cycle time in which link status telegrams are sent to the substation if there is no connection.

Additional information for **bForceC1Res** and **bForceC2Res** parameters:

Irrespective whether you set both parameters, one of them or none of them, the ACD bit is set accordingly in order to notify the control station whether Class 1 or Class 2 data should be queried next. These parameters only influence the behavior of the substation for Class 2 interrogations.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.78 ST_IEC870_5_102TBuffer

This data structure (TX/RX data buffer) is used for the data exchange via the IEC 60870-5-102 low-level interface.

Syntax

```
TYPE ST_IEC870_5_102TBuffer :
STRUCT
    eDbg          : E_IEC870_5_101FifoDbgFlags :=eIEC870_FIFO_DBG_OFF;
    eAsduAddrSize: E_IEC870_5_101AsduAddrSize;
    asduSize      : BYTE := 253;
    mode          : DWORD := 0;
    dataLink      : ST_IEC870_5_101DataLink;
END_STRUCT
END_TYPE
```

eDbg: Debugging output parameter [▶ 616].

eAsduAddrSize: Octet length of the common ASDU address [▶ 608] (CAA) := <1..2>.

asduSize: Maximum byte length of the ASDU := <50..253>.

mode: Reserved; currently not used. This value should be null.

dataLink: The elements of this data structure should not be accessed directly, but only with an instance of the FB_IEC870_5_102TBufferCtrl [▶ 508] function block.

Internally, the TX/RX data buffer uses two transmit FIFOs and one receive FIFO:

1. Class 1 transmit FIFO with (high-priority) data;
2. Class 2 transmit FIFO with (low-priority) data;
3. Receive FIFO (for Class 1 and Class 2 data);

The lower transport function of the library first empty the Class 1 FIFO, then the Class 2 FIFO. The Class 2 data can only be sent if the Class 1 FIFO contains no data to be sent.

Each of the internal FIFOs has a fixed size of 200000 bytes. Experience shows that each FIFO can hold approx. 2000 ASDUs with one information element (object) with the maximum size or approx. 600 ASDUs with a sequence of 100 information elements (objects).

If a larger number of transmit or receive frames is to be cached (e.g. >2000), external buffers/FIFOs specified by the PLC programmer can be used. At runtime, the PLC application can then top up the TwinCAT transmit FIFO with its own FIFO entries or empty the TwinCAT receive FIFO, if many frames are received. Another possibility is to use two buffers, for example, and alternately fill/read them and transfer the content to the communication block.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.79 ST_IEC870_5_102AOGen

Variables of this type represent an ASDU object.

Syntax

```
TYPE ST_IEC870_5_102AOGen:
STRUCT
    head : ST_IEC870_5_101FifoHead :=( source := ( link := 0, addr := 0 ),
                    target := ( link := 0, addr := 0 ),
                    ctrl := 0 );

    ident   : ST_IEC870_5_102DataUnit_Ident := ( eType := ASDU_TYPEUNDEF_2,
                         nObj := 0,
                         bSQ := FALSE,
                         bT := FALSE,
                         bPN := FALSE,
                         eCOT := 0,
                         asduAddr := 0,
                         eClass := eIEC870_Class_None );
 info    : ST_IEC870_5_102AOInfoObj  := ( rcdAddr := 0,
                        n  := 0,
                        stream  := ( length := 0 ) );
END_STRUCT
END_TYPE
```

head: Header (reserved).

ident: ID fields of the data unit [▶ 647] (ASDU).

info: IEC 60870-5-102 information object description [▶ 647].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.80 ST_IEC870_5_102AOInfoObj

IEC 60870-5-102 information object description.

Syntax

```
TYPE ST_IEC870_5_102AOInfoObj :
STRUCT
    rcdAddr : BYTE;
    stream  : ST_IEC870_5_101Stream;
END_STRUCT
END_TYPE
```

rcdAddr: List address of a measuring period or list address [▶ 329] of a single point.

stream: Information elements or information objects [▶ 587] (byte array buffer).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.81 ST_IEC870_5_102DataUnit_Ident

ID field of the data unit (ASDU).

Syntax

```
TYPE ST_IEC870_5_102DataUnit_Ident:
STRUCT
    eType   : E_IEC870_5_102TypeID;
    bSQ     : BOOL;
    nObj    : BYTE(0..127);
    bT      : BOOL;
    bPN     : BOOL;
    eCOT    : E_IEC870_5_102COTType;
    asduAddr: DWORD;
    eClass  : E_IEC870_5_101ClassType;
END_STRUCT
END_TYPE
```

eType: IEC 60870-5-102 ASDU type ID [▶ 605] := <0..255>.

bSQ: Sequence flag := <FALSE = individual, TRUE = sequence>.

nObj: Number of information objects or information elements. Permitted range: <0..127>.

bT: Test bit (rarely used).

bPN: Positive/negative confirmation := <FALSE = positive, TRUE = negative>.

eCOT: IEC 60870-5-102 cause of transfer [▶ 614] (COT). Permitted range: <0..63>.

asduAddr: Common ASDU address (CAA).

eClass: Priority class [▶ 611].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.82 ST_IEC870_5_103TBuffer

This data structure (TX/RX data buffer) is used for the data exchange via the IEC 60870-5-103 low-level interface.

Syntax

```
TYPE ST_IEC870_5_103TBuffer :
STRUCT
    eDbg    : E_IEC870_5_101FifoDbgFlags :=eIEC870_FIFO_DBG_OFF;
    asduSize: BYTE := 249;
    mode    : DWORD := 0;
    dataLink: ST_IEC870_5_101DataLink;
END_STRUCT
END_TYPE
```

eDbg: Debugging output parameter [▶ 616].

asduSize: Maximum byte length of the ASDU.

mode: Reserved; currently not used. This value should be null.

dataLink: The elements of this data structure should not be accessed directly, but only with an instance of the FB_IEC870_5_103TBufferCtrl [▶ 509] function block.

Internally, the TX/RX data buffer uses two transmit FIFOs and one receive FIFO:

1. Class 1 transmit FIFO with (high-priority) data;
2. Class 2 transmit FIFO with (low-priority) data;
3. Receive FIFO (for Class 1 and Class 2 data);

The lower transport function of the library first empty the Class 1 FIFO, then the Class 2 FIFO. The Class 2 data can only be sent if the Class 1 FIFO contains no data to be sent.

Each of the internal FIFOs has a fixed size of 200000 bytes. Experience shows that each FIFO can hold approx. 2000 ASDUs with one information element (object) with the maximum size or approx. 600 ASDUs with a sequence of 100 information elements (objects).

If a larger number of transmit or receive frames is to be cached (e.g. >2000), external buffers/FIFOs specified by the PLC programmer can be used. At runtime, the PLC application can then top up the TwinCAT transmit FIFO with its own FIFO entries or empty the TwinCAT receive FIFO, if many frames are received. Another possibility is to use two buffers, for example, and alternately fill/read them and transfer the content to the communication block.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.83 ST_IEC870_5_103AOGen

Variables of this type represent an ASDU object.

Syntax

```
TYPE ST_IEC870_5_103AOGen:
STRUCT
    head : ST_IEC870_5_101FifoHead :( source := ( link := 0, addr := 0 ),
                  target := ( link := 0, addr := 0 ),
                  ctrl   := 0 );

    ident   : ST_IEC870_5_103DataUnit_Ident := ( e := 0,
                        nObj    := 0,
                        bSQ     := FALSE,
                        eCOT    := 0,
                        eClass  := eIEC870_Class_None,
                        asduAddr:= 0 );
    info    : ST_IEC870_5_103AOInfoObj  := ( fc := 0,
                                             n := 0,
                                             stream  := ( length := 0 ) );
END_STRUCT
END_TYPE
```

head: Header (reserved).

ident: ID fields of the data unit [▶ 650] (ASDU).

info: Information object/information element data field [▶ 649].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.84 ST_IEC870_5_103AOInfoObj

IEC 60870-5-103 information object description.

Syntax

```
TYPE ST_IEC870_5_103AOInfoObj :
STRUCT
    fc      : BYTE;
    n       : BYTE;
    stream  : ST_IEC870_5_101Stream;
END_STRUCT
END_TYPE
```

fc: Function number/function code [▶ 388].

n: Information number [▶ 388].

stream : Information element [▶ 587] of max. length.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.85 ST_IEC870_5_103DataUnit_Ident

ID field of the data unit (ASDU).

Syntax

```
TYPE ST_IEC870_5_103DataUnit_Ident:
STRUCT
    eType    : INT(0..255);
    bSQ      : BOOL;
    nObj     : BYTE(0..127);
    eCOT     : INT(0..255);
    asduAddr: BYTE;
    eClass   : E_IEC870_5_101ClassType;
END_STRUCT
END_TYPE
```

eType: IEC 60870-5-103 ASDU type ID in control direction [▶ 607] or ASDU type ID in monitoring direction [▶ 607] := < 0..255>.

bSQ: Sequence flag := <FALSE = single, TRUE = sequence>.

nObj: Number of information objects or information elements := <0..127>.

eCOT: IEC 60870-5-103 cause of transfer in control direction [▶ 616] or cause of transfer in monitoring direction [▶ 615] := <0..255>.

asduAddr: Common ASDU address (CAA, one byte).

eClass: Priority class [▶ 611].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.86 ST_IEC870_5_104ProtocolParams

Syntax

```
TYPE ST_IEC870_5_104ProtocolParams :
STRUCT
    t0            : TIME := T#30s;
    t1            : TIME := T#15s;
    t2            : TIME := T#10s;
    t3            : TIME := T#20s;
    iK            : INT := 12;
    iW            : INT := 8;

    bSFrameACK    : BOOL := TRUE;
    bTESTFRAct    : BOOL := TRUE;
    bDTControlled: BOOL := TRUE;
    bControlDT    : BOOL := FALSE;

    bSTARTDTCon   : BOOL := TRUE;
    bSTOPDTCon    : BOOL := TRUE;
    bTESTFRCon    : BOOL := TRUE;

    eAcceptMode   : E_SocketAcceptMode := eACCEPT_ALL;
    sRemoteHost   : STRING(15) := '';
    nRemotePort   : UDINT := 0;

    APDULength    : BYTE(MIN_IEC870_5_104_APDULEN..MAX_IEC870_5_104_APDULEN) := MAX_IEC870_5_104_APDU
LEN;
    bThrottleMode: BOOL := FALSE;
    bPackFrames   : BOOL := FALSE;
    bRetainBuffer: BOOL := FALSE;
    bCOTFilter    : BOOL := TRUE;
    cotFilter     : T_IEC870_5_101COTBits := 2#10001111, 7(16#FF);
    eSwitchMode   : E_IEC870_5_104GrpSwitchMode := eIEC870_SWITCHMODE_OFF;
END_STRUCT
END_TYPE
```

t0: Cycle time for connection attempts (control station only).

t1: Max. timeout time for a response to a STARTDTAct, STOPDTAct or TESTFRAct frame.

t2: An S frame is sent after this time, at the latest.

t3: A test frame is sent after this time, at the latest.

iK: The connection is closed at the latest after this APDU was sent in I format but not acknowledged.

iW: The receiver acknowledges the reception at the latest after receipt of w APDUs in I format.

bSFrameACK: Send S frames.

bTESTFRAct: Send test frames.

bDTControlled : Wait for STARTDT, STOPDT frame from master.

bControlDT : Only for the master configuration: Send STARTDT to the slave.

bSTARTDTCon: Send STARTDT confirmation

bSTOPDTCon: Send STOPDT confirmation.

bTESTFRCon: Send TESTFR confirmation.

eAcceptMode : Determines whether connections are permitted to all remote clients, or only to clients with certain host and port addresses. Default: Accept all incoming connections. On the substation side this only works for a logical connection.

sRemoteHost: Host address of the remote client. If eAcceptMode = eACCEPT_ALL this parameter is ignored.

nRemotePort : Port address of the remote client. If eAcceptMode = eACCEPT_ALL this parameter is ignored.

APDULength: Maximum length of the APDU. Default values:

Max. length of APDU = 255 bytes - 1 start octet - 1 length octet = 253 octets;
Max. length of ASDU = 253 - 4 control octets = 249 octets;

bThrottleMode: The TCP/IP sockets are polled from the PLC. This parameter can be used to reduce the number of polling read access operations and therefore the system load, particularly if data are received only rarely (e.g. general interrogation or clock synchronization commands).

- FALSE: Each read access is followed by the next one, even if no new data were received (higher system load);
- TRUE: Each read access, which does not return new data, is followed by a delay. The next read access is delayed (lower system load). Each read access, which returns new data, is followed by the next one without delay. The maximum delay time is <= 2 seconds.

bPackFrames: By default a TCP/IP send call only sends a single APDU. The send performance can be increased substantially (thereby reducing send buffer overflows) by setting this parameter to TRUE.

- FALSE: For an APDU a TCP/IP send call is required. A maximum of 1 APDU can be sent about every 3 PLC cycles;
- TRUE: Several APDUs are consolidated to form a larger TCP/IP block (maximum iK frames) and sent with a TCP/IP send call;

bRetainBuffer: In the standard setting (FALSE) the internal Tc/Rx buffers are deleted when the connection is severed. If this flag is set to TRUE, the ASDUs in the internal send buffer that have not yet been sent are not deleted. Hence, offline buffering of approx. 100-200 measured values (dependent on the ASDU size) in the RAM memory is possible. The station always removes the ASDUs from the send buffer when their receipt has been confirmed. Please note that the ASDUs that have already been sent but not yet confirmed also remain in the buffer and are sent again the next time. The other station may then receive the values twice. You can configure this behavior via two further parameters: **bCOTFilter** and **cotFilter**.

bCOTFilter: Activates/deactivates the filter mask with the causes of transfer. This parameter is only valid when the *bRetainBuffer* parameter is also set to TRUE. A SessionID is incremented internally with each establishment of a new connection. This SessionID is always attached to the received and sent ASDUs. Hence, the ASDUs that have not yet been sent and remain in the offline buffer are assigned to the old connection. These ASDUs can then be filtered and discarded with the aid of the COT mask (**cotFilter**) (COT = Cause Of Transfer). This is sometimes necessary if the other communication partner does not accept the repeated ASDUs.

cotFilter: Filter mask with causes of transfer (COT = Cause Of Transfer). This parameter is only valid when the *bRetainBuffer* and *bCOTFilter* parameters have also been set. The cause of transfer of the ASDUs to be sent is only checked if their SessionID does not correlate to the current SessionID (i.e. the ASDUs originating from the previous connection). Each bit corresponds to a cause of transfer. The cause of transfer is only checked if the appropriate bit has been set.

The causes of transfer are coded in the following way in the bits:

cotFilter[0].7 = eIEC870_COT_UNUSED
cotFilter[0].6 = eIEC870_COT_CYCLIC
cotFilter[0].5 = eIEC870_COT_BACKGROUND
cotFilter[0].4 = eIEC870_COT_SPONTAN
cotFilter[0].3 = eIEC870_COT_INIT
cotFilter[0].2 = eIEC870_COT_REQ
cotFilter[0].1 = eIEC870_COT_ACT

cotFilter[0].0 = eIEC870_COT_ACT_CON
cotFilter[1].7 = eIEC870_COT_DEACT
cotFilter[1].6 = eIEC870_COT_DEACT_CON
cotFilter[1].5 = eIEC870_COT_ACT_TERM

... etc.

Default: Filtering of COT: eIEC870_COT_CYCLIC or eIEC870_COT_BACKGROUND or eIEC870_COT_SPONTAN enabled.

eSwitchMode : Redundancy group switching mode (reserved).

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.87 ST_IEC870_5_104ExSystemInterface

Syntax

```
TYPE ST_IEC870_5_104ExSystemInterface :
STRUCT
    device      : ST_IEC870_5_101DeviceInterface;
    service     : ST_IEC870_5_101SystemServices;
    hSOTable    : T_HAODBTABLE;
END_STRUCT
END_TYPE
```

device: Internal communication interface of the IEC device [▶ 591].

service: IEC device services [▶ 583].

hSOTable: System object database handle [▶ 633].

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.88 ST_IEC870_5_104ServerConnection

An IEC 60870-5-104 substation acts as TCP/IP server. Theoretically, a TCP/IP server can establish connections with several clients. A variable of this type represents a communication link between the IEC 870-5-104 substation and the control station.

Sample of a declaration for a triple connection:

Code

```
connections : ARRAY[0..2] OF ST_IEC870_5_104ServerConnection;
```

Sample of a declaration for a single connection:

Code

```
connections : ARRAY[0..0] OF ST_IEC870_5_104ServerConnection;
```

NOTICE

The structural elements should not be written to or modified directly.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.89 ST_IEC870_5_104GrpStatus

Status of an IEC870-5-104 slave group.

Syntax

```
TYPE ST_IEC870_5_104GrpStatus:
STRUCT
    nConnected : DWORD := 0;
    nSuspended : DWORD := 0;
    nDTStarted : DWORD := 0;
END_STRUCT
END_TYPE
```

nConnected: Number of established TCP/IP connections (ESTABLISHED).

nSuspended: Number of connections whose state is in the process of changing (CONNECTED->DISCONNECTED or DISCONNECTED->CONNECTED).

nDTStarted: Number of connections with active data transfer.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.3.90 ST_IEC870_5_104SystemInterface

Syntax

```
TYPE ST_IEC870_5_104SystemInterface :
STRUCT
    device  : ST_IEC870_5_101DeviceInterface;
```

```
    service : ST_IEC870_5_104SlaveServices;
END_STRUCT
END_TYPE
```

device: Communication interface [▶ 591] of the IEC device.

service: IEC device service.

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.4 Global constants

5.4.1 Group configuration flags

The group flags are used to assign application objects (data points) to certain groups. The group flags for station query and count transfer are defined as follows:

- Station interrogation: All application objects that were assigned to group 1 to 16 automatically also belong to the global group: IEC870_GRP_INROGEN;
- Count transfer: All application objects that were assigned to group 1 to 4 automatically also belong to the global group: IEC870_GRP_REQCOGEN;

Automatic assignment of the application objects to the global group can be prevented by masking the relevant bits during configuration.

Examples:

IEC870_GRP_INRO3 AND NOT IEC870_GRP_INROGEN

or

IEC870_GRP_REQCO1 AND NOT IEC870_GRP_REQCOGEN

Constant	Value	Description
Station interrogation		
IEC870_GRP_INROGEN	16#00000001	Queried through general query (station or global)
IEC870_GRP_INRO1	16#00000003	Queried through station query of group 1
IEC870_GRP_INRO2	16#00000005	Queried through station query of group 2
IEC870_GRP_INRO3	16#00000009	Queried through station query of group 3
IEC870_GRP_INRO4	16#00000011	Queried through station query of group 4
IEC870_GRP_INRO5	16#00000021	Queried through station query of group 5
IEC870_GRP_INRO6	16#00000041	Queried through station query of group 6
IEC870_GRP_INRO7	16#00000081	Queried through station query of group 7
IEC870_GRP_INRO8	16#00000101	Queried through station query of group 8
IEC870_GRP_INRO9	16#00000201	Queried through station query of group 9
IEC870_GRP_INRO10	16#00000401	Queried through station query of group 10
IEC870_GRP_INRO11	16#00000801	Queried through station query of group 11
IEC870_GRP_INRO12	16#00001001	Queried through station query of group 12
IEC870_GRP_INRO13	16#00002001	Queried through station query of group 13
IEC870_GRP_INRO14	16#00004001	Queried through station query of group 14
IEC870_GRP_INRO15	16#00008001	Queried through station query of group 15
IEC870_GRP_INRO16	16#00010001	Queried through station query of group 16
Count transfer		
IEC870_GRP_REQCOGEN	16#00020000	Queried through general counter query
IEC870_GRP_REQCO1	16#00060000	Queried through counter query of group 1
IEC870_GRP_REQCO2	16#000A0000	Queried through counter query of group 2
IEC870_GRP_REQCO3	16#00120000	Queried through counter query of group 3
IEC870_GRP_REQCO4	16#00220000	Queried through counter query of group 4
IEC870_GRP_LOCFREEZE	16#00400000	Enables cyclic local relocation of the counter values
IEC870_GRP_LOCRESET	16#00800000	Enables cyclic local resetting of the counter values
Others		
IEC870_GRP_IV_OFF	16#01000000	Disables setting of the quality flag: Invalid (IV)
IEC870_GRP_REVERSE	16#02000000	Object is used in reverse direction (reserved)
IEC870_GRP_SELECTCMD	16#04000000	Force command selection/execution for this data point (reserved)
not defined	16#08000000	not used (reserved)
IEC870_GRP_USERCMD	16#10000000	User-defined command (reserved)
IEC870_GRP_BACKGROUND	16#20000000	Enables background query for this data point
IEC870_GRP_PERCYC	16#40000000	Enables periodic/cyclic data transfer for this data point
IEC870_GRP_SPONTOFF	16#80000000	Disables spontaneous data transfer for this data point

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.4.2 Quality flags

Constant	Value	Description
IECQ_BL_ON	16#0001	Blocked
IECQ_BL_OFF	16#0002	Not blocked
IECQ_SB_ON	16#0004	Replaced
IECQ_SB_OFF	16#0008	Not replaced
IECQ_NT_ON	16#0010	Not current
IECQ_NT_OFF	16#0020	Current
IECQ_IV_ON	16#0040	Invalid
IECQ_IV_OFF	16#0080	Valid
IECQ_OV_ON	16#0100	Overflow
IECQ_OV_OFF	16#0200	No overflow
IECQ_EI_ON	16#0400	Elapsed time is invalid
IECQ_EI_OFF	16#0800	Elapsed time is valid
IECQ_CY_ON	16#1000	Carry
IECQ_CY_OFF	16#2000	No carry
IECQ_CA_ON	16#4000	Counter was set
IECQ_CA_OFF	16#8000	Counter was not set

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.4.3 IEC 60870-5-10x error codes

Error code (Hex)	Error code (Dec)	Symbolic constant	Description
0x00008101	33025	IEC870_COMMERR_INVALIDSTARTBYTE	Invalid frame start character <> 0x68
0x00008102	33026	IEC870_COMMERR_RXBUFFOVERFLOW	Receive buffer overflow
0x00008103	33027	IEC870_COMMERR_TXBUFFOVERFLOW	Send buffer overflow
0x00008104	33028	IEC870_COMMERR_INVALIDUFFMT	Invalid U-Frame format, more than one function (STARTDT, STOPDT, TESTFR) activated
0x00008105	33029	IEC870_COMMERR_INVALIDSFFMT	Invalid S-Frame format, invalid length parameter
0x00008106	33030	IEC870_COMMERR_T1RESPONSE	t1 (response timeout) expired
0x00008107	33031	IEC870_COMMERR_SENDSEQ	Send sequence error
0x00008108	33032	IEC870_COMMERR_KOVERFLOW	k reached
0x00008109	33033	IEC870_COMMERR_FATALERR	Fatal internal error
0x0000810A	33034	IEC870_COMMERR_INVALIDSTATE	Device is in invalid state (disconnected?)
0x0000810B	33035	IEC870_COMMERR_INVALIDSIZE	Invalid parameter size
0x0000810C	33036	IEC870_COMMERR_INVALIDVALUE	Invalid parameter value
0x0000810D	33037	IEC870_COMMERR_INVALIDTYPE	Invalid asdu (object) type
0x0000810F	33039	IEC870_COMMERR_TIMEOUT	Communication timeout
0x00008110	33040	IEC870_COMMERR_LENGTH1	Invalid length field value
0x00008111	33041	IEC870_COMMERR_LENGTH2	Length field and length field copy differs
0x00008112	33042	IEC870_COMMERR_STARTCHAR2	Invalid second stat character
0x00008113	33043	IEC870_COMMERR_CHECKSUM	Invalid checksum
0x00008114	33044	IEC870_COMMERR_ENDCHAR	Invalid end character
0x00008115	33045	IEC870_COMMERR_LINKADDR	Invalid link address size
0x00008116	33046	IEC870_COMMERR_SRVFUNCODE	Invalid link service function code
0x00008117	33047	IEC870_COMMERR_FRAMETYPE	Invalid frame type
0x00008118	33048	IEC870_COMMERR_UNSUPMODE	Unsupported communication mode (balanced mode)
0x00008119	33049	IEC870_COMMERR_T2KOVERFLOW	k reached and t2 (response timeout) expired
0x0000811A	33050	IEC870_COMMERR_INVALIDCONFIG	Invalid object configuration/initialization
0x0000811B	33051	IEC870_COMMERR_UNKNOWNTYPE	Unknown asdu type
0x0000811C	33052	IEC870_COMMERR_UNKNOWNCOT	Unknown cause of transfer

Error code (Hex)	Error code (Dec)	Symbolic constant	Description
0x0000811D	33053	IEC870_COMMERR_UNKNOWNASDUADDR	Unknown asdu address
0x0000811E	33054	IEC870_COMMERR_UNKNOWNOBJADDR	Unknown object address
0x0000811F	33055	IEC870_COMMERR_NEGACTCON	Negative activation confirmation
0x00008120	33056	IEC870_COMMERR_NEGACTTERM	Negative activation termination
0x00008121	33057	IEC870_COMMERR_NEGDEACTCON	Negative deactivation confirmation
0x00008122	33058	IEC870_COMMERR_BUSY	Already in busy state
0x00008123	33059	IEC870_COMMERR_AODBOVERFLOW	Application object database overflow
0x00008124	33060	IEC870_COMMERR_AODBNOTFOUND	Application object not in database
0x00008125	33061	IEC870_COMMERR_ACTCONTIMEOUT	Activation confirmation timeout error
0x00008126	33062	IEC870_COMMERR_ACTTERMTIMEOUT	Activation termination timeout error
0x00008127	33063	IEC870_COMMERR_DEACTCONTIMEOUT	Deactivation termination timeout error
0x00008128	33064	IEC870_COMMERR_SELEXECTIMEOUT	Command select/execute timeout error
0x00008129	33065	IEC870_COMMERR_READRESTIMEOUT	Read command response timeout error
0x00008130	33072	IEC870_COMMERR_LIBNOTCOMPAT	Product libraries are incompatible
0x00008131	33073	IEC870_COMMERR_DIR	Invalid DIR bit value
0x00008132	33074	IEC870_COMMERR_PRM	Invalid PRM bit value
0x00008133	33075	IEC870_COMMERR_FCV	Invalid FCV bit value
0x00008134	33076	IEC870_COMMERR_SCANTIMEOUT	Station scan cycle timeout error

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.4.4 Parameter list

Constant	Value	Description
IEC870_MAX_RXTX_FIFO_SIZE	200000	Maximum Tx/Rx FIFO size (in bytes)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4018.26 TF6500 v3.0.5.0	PC or CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

5.4.5 Library version

All libraries have a specific version. This version information is displayed in the PLC library repository. The function F_CmpLibVersion (defined in the Tc2_System library) can be used to compare the available version information with the required information.
A global constant in the library includes the version information:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_IEC60870_5_10x : ST_LibVersion;
END_VAR
```



The options familiar from TwinCAT 2.x for querying the version information are outdated

Requirements

Development environment	Target system type	PLC libraries to be linked (category group)
TwinCAT v3.1.4012.0	PC oder CX (x86, x64, Arm®)	Tc2_IEC60870_5_10x (Communication->IEC60870) Tc2_SerialCom (Communication->Serial) Tc2_TcpIp (Communication->TcpIp) Tc2_Utilities (System)

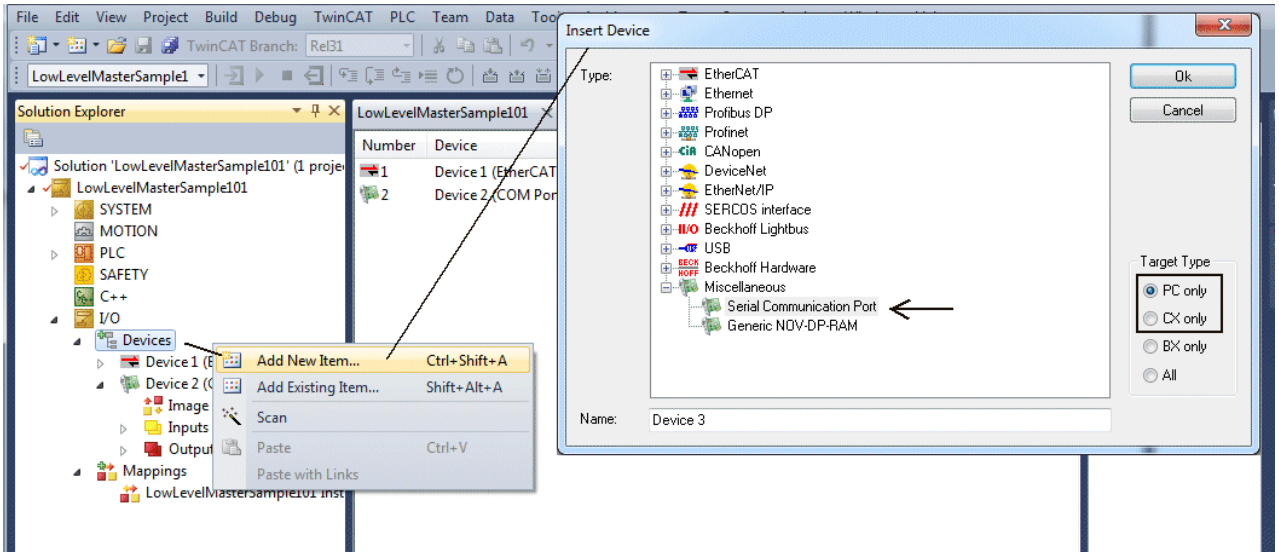
6 Samples

6.1 Configuration of serial interfaces

- Standard PC COMx port [▶ 662]
- Serial KL6xxx bus terminal [▶ 663]
- Serial EL6xxx bus terminal [▶ 664]

1. Standard PC COMx port

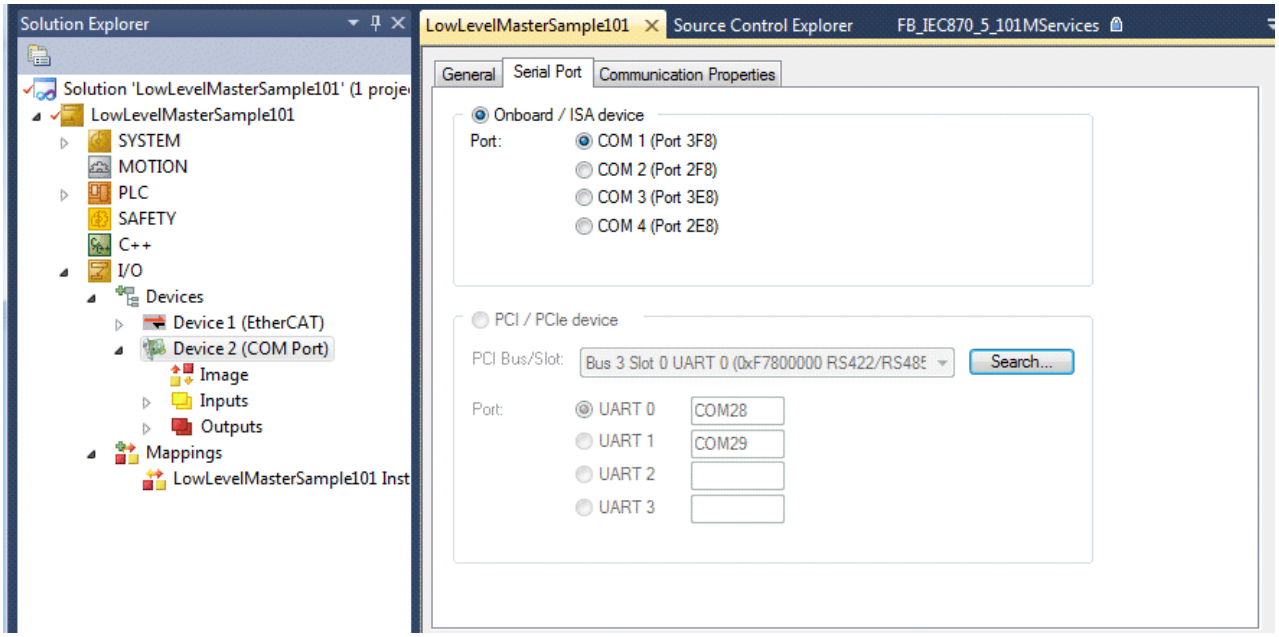
The serial interface can only be added in PC/CX systems. Right-click on "I/O->Devices" node. Select "Add new Item..." from context menue. Select "Serial Communication Port" under "Miscellaneous".



Then set the following settings.

1.1. Serial Port tab settings

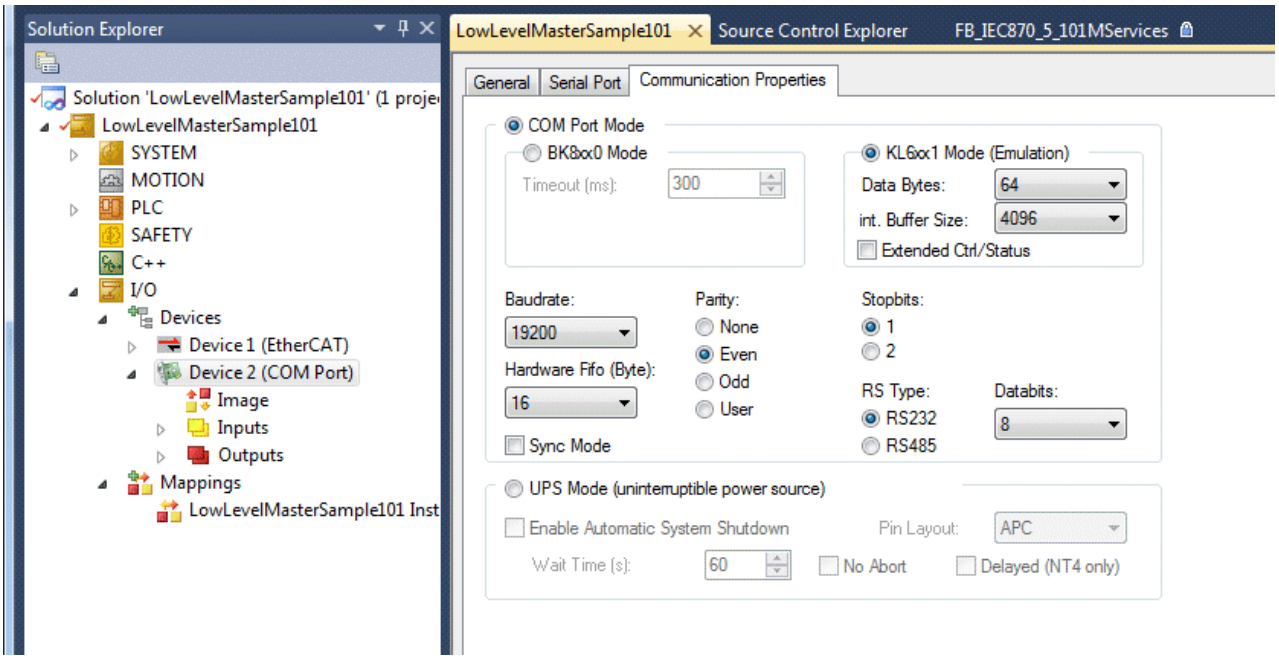
Select serial interface (integrated COM1..COM4 or one existing PCI/PCIe device).



1.2. Communication Properties tab settings

Select the option KL6xxx1 Mode (Emulation) and set the communication parameter:

Baud rate, here 19200 Baud, 8 data bits, Parity = even, 1 stop bit



2. Serial KL6xxx bus terminal

The interface is configured within the TwinCAT PLC project by the instance of the FB_IEC870_SerialLineCtrlfunction block. Communication parameters like baud rate, parity etc. have to be set via this function block.

```
(* High speed task. Serial background communication. *)
PROGRAM P_SerialComm_HighSpeed
VAR
    serial_In        AT%I*   : PcComInData;
    serial_Out       AT%Q*   : PcComOutData;
    KL6_In           AT%I*   : KL6inData5B;
    KL6_Out          AT%Q*   : KL6outData5B;
    serial_ExtStatus AT%I*   : WORD;
    serial_ExtCtrl   AT%Q*   : WORD;

    hSerial          : T_HSERIALCTRL;(* Serial connection control handle *)
fbSerialLineCtrl : FB_IEC870_SerialLineCtrl := ( Mode := SERIALLINEMODE_KL6_5B_STANDARD(*SERIALLINEMODE_PC_COM_PORT*),
                                                 Baudrate := 19200,
                                                 NoDatabits := 8,
                                                 Parity := PARITY_EVEN,
                                                 Stopbits := 1,
                                                 Handshake := HANDSHAKE_NONE,
                                                 ContinousMode := FALSE );
```

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged by means of half duplex transfer.The transmit and receive lines are connected to each other in RS485 operating mode. As a result, the terminal receives not only data from other devices, but also its own transmitted data. The own data may disturb the communication. For half-duplex mode, the *handshake* input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Serial EL6xxx bus terminal

The settings for the serial interface can be found in the CoE Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To avoid this, the modified data can be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

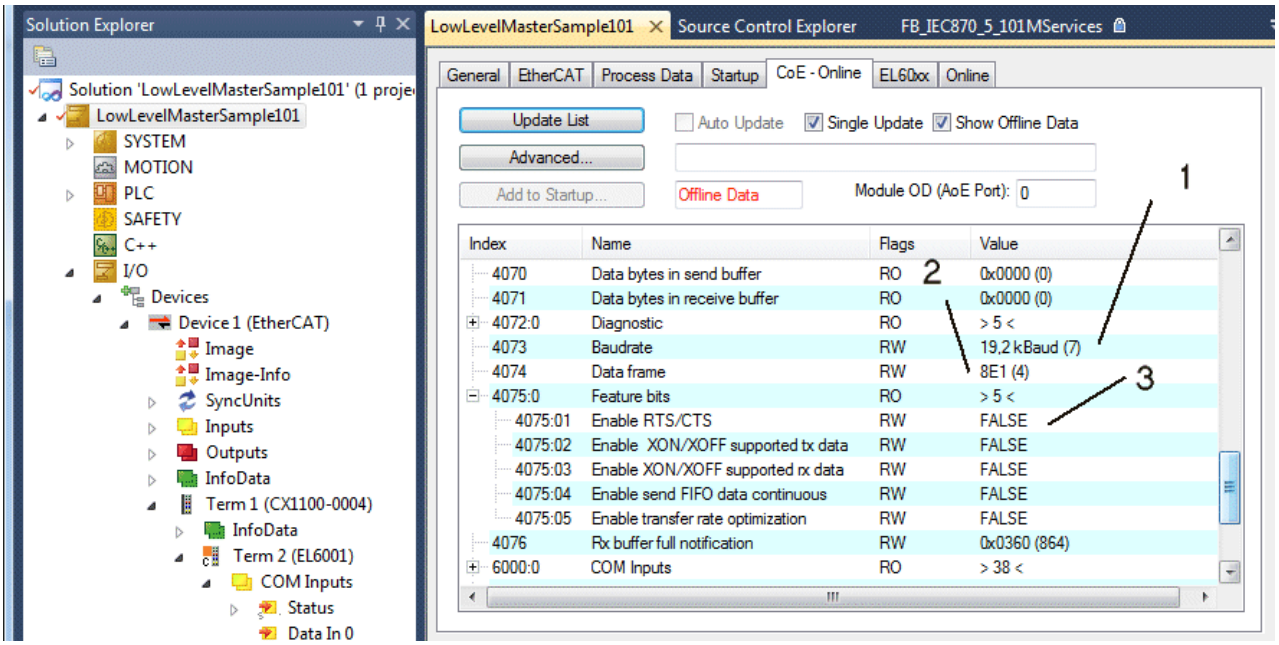
3.1. CoE tab settings

(1) Baud rate, here 19200 Baud

(2) Date frame, 8E1 corresponds to 8 data bits, Parity = even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In the RS485 mode, the data is exchanged by means of half duplex transfer. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own transmitted data. The own data may disturb the communication. This can be suppressed with the option "Enable half duplex" in the "COM Settings" object (index 8000:06).

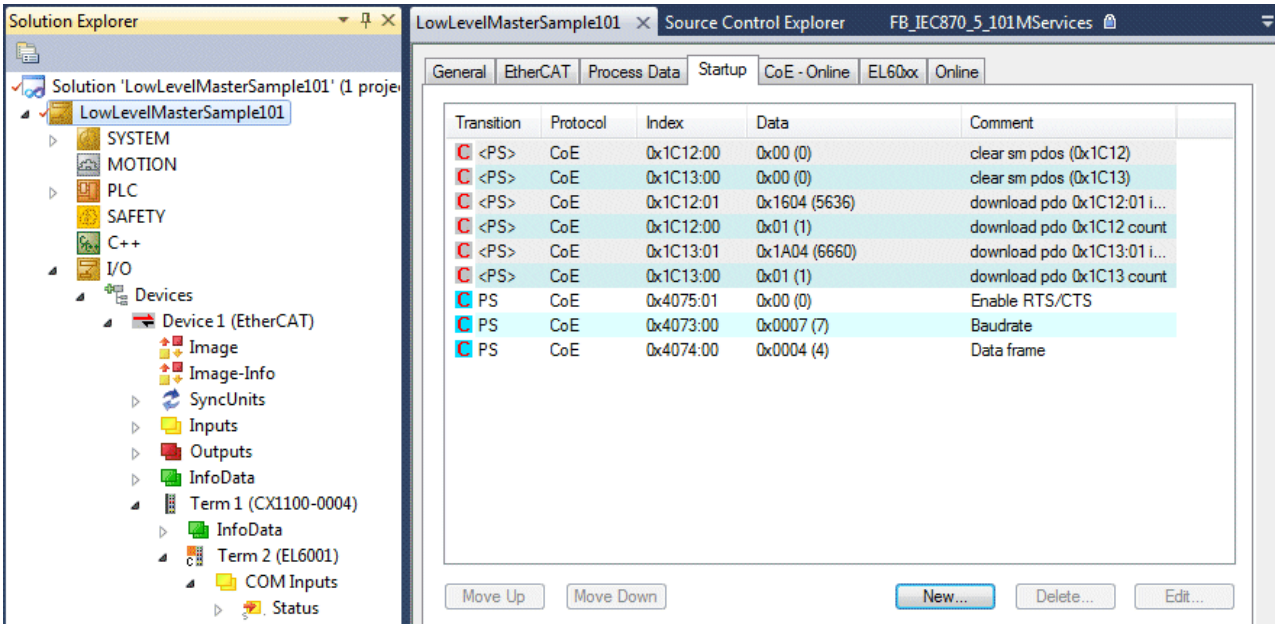
3.2. Startup tab settings

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 Baud

Date frame, 8E1 corresponds to 8 data bits, Parity = even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.2 IEC 60870-5-101

6.2.1 Control station

6.2.1.1 High-Level Interface

6.2.1.1.1 Quick start

Simple project with complete sources can be found here: IEC60870-5-101 control station [▶ 666].

Interoperability check list can be found here: Interoperability check list [▶ 13]

Communication- and/or device error code overview can be found here: Error codes [▶ 784]

Short quide

Application object database

The application object database of the central station must be configured as hash table with the function F_iecCreateTableHnd [▶ 554]. The individual array elements are linked with each other in the form of a hash table. This enables faster access to the individual data points, but also has certain disadvantages that have to be taken into account:

- The size of the application database (array size) must not modified at runtime (e.g. through online change). The central station stops execution immediately and reports an error. The reason: The hash table links no longer match. When the program is modified it is best to load the complete project into the runtime system.
- The individual array elements must not be accessed via the index but via the special functions (e.g. F_iecAddTableEntry [▶ 555] etc.).
- With indexed table element access the internal configuration parameters must not be overwritten or modified. If the type, the ASDU address or the object address is changed the data point can no longer be found. To reconfigure a data point it should first be removed from the table via function call F_iecRemoveTableEntry [▶ 562]. The new data point can then be added.

An implementation in the form of linear table would mean that for each received ASDU (data unit) the central station would have to search the complete array for a suitable element. With a large number of data points this would lead to long execution times.

Protocol parameters

Most protocol parameters are preconfigured with default values and do not have to be set explicitly.

System parameters

The system parameters are also preconfigured with default values. During commissioning it is useful to activate debugging output (*dbgMode*) in order to be able to locate any errors.

Parameters for cyclic data acquisition

The following parameters are preconfigured with default values:

- Initialisation sequence (consisting of a test command, time synchronisation, station query and counter query);
- Cyclic commands:
  - Test command every 60 s;
  - Time synchronisation every 60 s;
  - Group station query: generally every 60 s;
  - Group counter query: generally every 60 s;

6.2.1.1.1.1 Download: Central station Minimal (master, unbalanced mode)

Simple central station application that uses the TwinCAT IEC 60870-5-101 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984601099.zip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220.**
- Link address size: **2 octets.**
- Cause of transfer size: **2 octets (including originator address).**
- Originator address: **1.**
- Common ASDU address: **7.**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Link mode**: Unbalanced.**

Data point configuration (it should be the same as the data point configuration of the substations):

- Single point information: M_SP_NA_1 (IOA := 100).
- Single command: C_SC_NA_1 (IOA := 10).

Initialization sequence that is executed once/automatically after the connection is established:

- Test command (C_TS_NA_1).
- Clock synchronization command (C_CS_NA_1).

- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN). Counter interrogation Mode C

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 60 seconds.
- Clock synchronization command (C_CS_NA_1) every 60 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 60 seconds.
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN) every 60 seconds. Counter interrogation Mode C.
- If the value of the **bSCS** variable changes, the single command (C_SC_NA_1) is sent (direct command execution)
- When spontaneous data of type (M_SP_NA_1, IOA := 100) is received, the ON/OFF value is mapped to the PLC variable **bSPI.**

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbClient	FB_IEC870_5_101Master (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
bSPI	BOOL	The received ON/OFF value is mapped into this PLC variable directly to the allocated flag address: **AT%MX0.0**.
bSCS	BOOL	The ON/OFF value to be sent is mapped directly from the flag address **AT%MX1.5** of this allocated PLC variable.
AODB	ARRAY of ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
hTable	T_HAODBTable (STRUCT)	Application object database handle. These variables are used to access the application object database and never directly the individual array elements.
F_iecCreateTableHnd	FUNCTION	In the initialization step, all array elements of AODB are initialized once with this function call. If successful, this function call returns the handle.
F_iecAddTableEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.

6.2.1.1.2 Tutorial

The introduction explains how to implement and configure an IEC60870-5-101 central station (master) in the TwinCAT PLC. You can download the sample on the following page [▶ 667].

6.2.1.1.2.1 Download: Central station Tutorial (master, unbalanced mode)

A more complex central station application with different data points in both directions and a connection to the substation. The sample application uses the TwinCAT IEC 60870-5-101 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984604043.zip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 Baud** .
- Link address: **220.**
- Link address size: **2 octets.**
- Cause of transfer size: **2 octets (including originator address).**
- Originator address: **1.**
- Common ASDU address: **7.**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Link mode**: Unbalanced.**
- Class 1 poll cycle: **100ms.**
- Class 2 poll cycle: **100ms.**

Data point configuration (it should be the same as the data point configuration of the substation):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200), M_DP_TA_1 (IOA := 201), M_DP_TB_1 (IOA := 202).
- Regulating step value: M_ST_NA_1 (IOA := 300), M_ST_TA_1 (IOA := 301), M_ST_TB_1 (IOA := 302).
- Bitstring value: M_BO_NA_1 (IOA := 400), M_BO_TA_1 (IOA := 401), M_BO_TB_1 (IOA := 402).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500), M_ME_TA_1 (IOA := 501), M_ME_TD_1 (IOA := 502).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600), M_ME_TB_1 (IOA := 601), M_ME_TD_1 (IOA := 602).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700), M_ME_TC_1 (IOA := 701), M_ME_TF_1 (IOA := 702).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10), C_SC_NA_1 (IOA := 11), C_SC_TA_1 (IOA := 12).
- Double command: C_DC_NA_1 (IOA := 20), C_DC_NA_1 (IOA := 21), C_DC_TA_1 (IOA := 22).
- Bitstring command: C_BO_NA_1 (IOA := 40), C_BO_NA_1 (IOA := 41), C_BO_TA_1 (IOA := 42).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50), C_SE_NA_1 (IOA := 51), C_SE_TA_1 (IOA := 52).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60), C_SE_NB_1 (IOA := 61), C_SE_TB_1 (IOA := 62).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70), C_SE_NC_1 (IOA := 71), C_SE_TC_1 (IOA := 72).

Initialization sequence that is executed once/automatically after the connection is established:

- No test command (C_TS_NA_1, deactivated).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- No counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, disabled). Counter interrogation Mode A (counters are stored locally in the substation and sent to the central station as spontaneous data).

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 60 seconds.

- Clock synchronization command (C_CS_NA_1) every 60 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 60 seconds.
- No counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, disabled). Counter interrogation Mode A (counters are stored locally in the substation and sent to the central station as spontaneous data).
- Simulation and transmission of the commands every 10 seconds if the variable **bSendCmd** was set to TRUE (see upper data point configuration).
- Reception of spontaneous data.
- Reception of M_EI_NA_1 (end of initialization).

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbClient	FB_IEC870_5_101Master (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
hTable	T_HAODBTable (STRUCT)	Application object database handle. These variables are used to access the application object database and never directly the individual array elements.
F_iecCreateTableHnd	FUNCTION	In the initialization step, all array elements of AODB are initialized once with this function call. If successful, this function call returns the handle.
F_iecAddTableEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.
memory	ST_MemoryData (STRUCT)	This structure contains constantly updated process data – either received or to be sent.
fbHandler	FB_IEC870_5_101TableEventHandler (FB)	This function block informs the application when data point data has been changed or updated.

6.2.1.1.2.2 Glossary

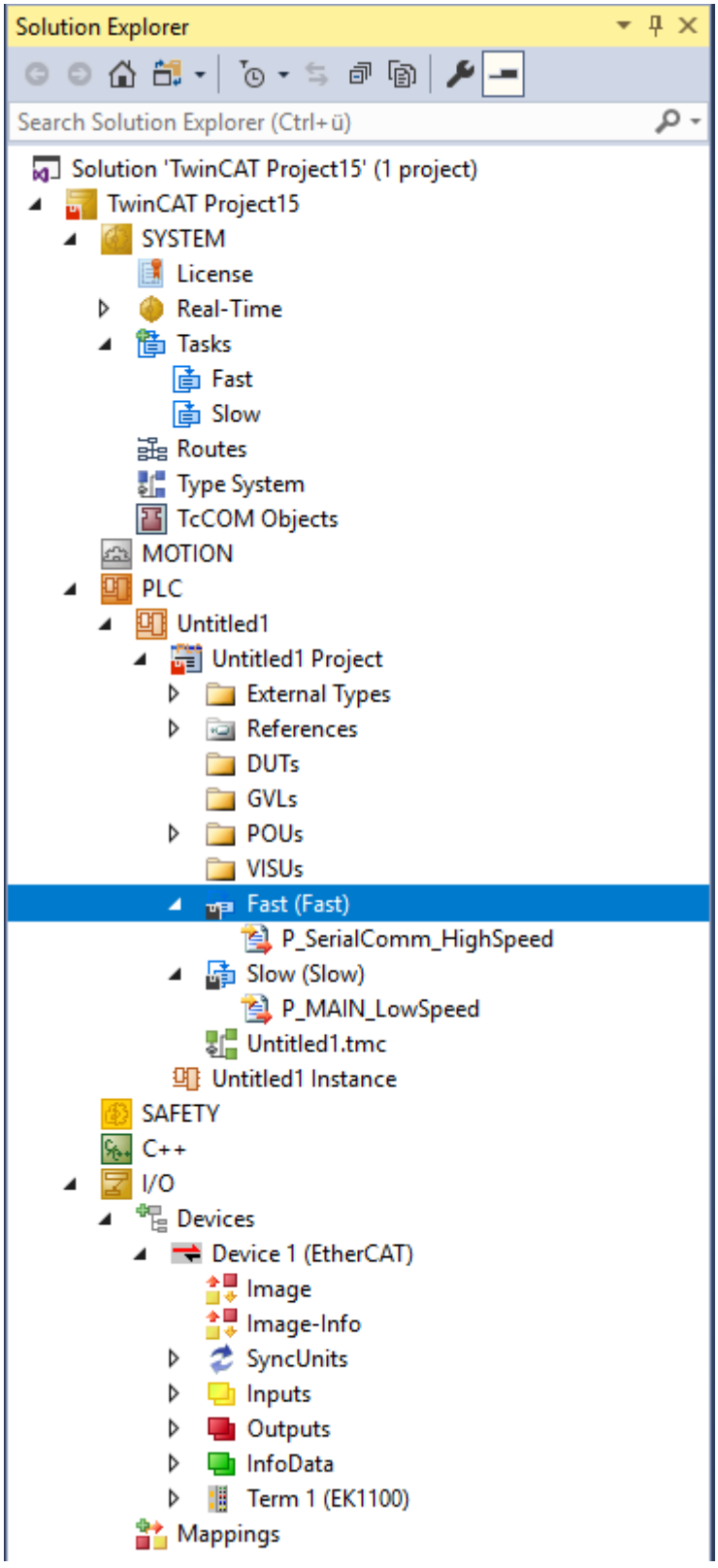
Term	Description
substation, slave, server	Synonyms for a subordinate station (which is monitored)
Central station, control station, master, client	Synonyms for a higher-level station (control station, monitors others stations)
control direction	Data transfer direction from the central station to the substation
monitoring direction	Data transfer direction from the substation to the central station
application objects	IEC information objects in the TwinCAT PLC application (single points, double points, measured values, short floating point values etc.)
APDU	Protocol data unit of the application layer (application protocol data unit)
APCI	Protocol control information of the application layer (application protocol control information)
ASDU	Service data unit of the application layer (application service data unit)
IOA, address of the information object	Address of the single point, double point, etc. (information object address)
primary station	The primary station sends commands (requests) to the secondary station and thus controls the data transfer of the secondary station.
secondary station	The secondary station responds to requests from the primary station.
Combined station	Combined stations can assume the role of primary station and secondary station (symmetric transfer, balanced mode).
Symmetric transfer (balanced mode)	Both stations can act as primary or secondary stations and initialize the data transfer.
Asymmetric transfer (unbalanced mode)	Data transfer is always initiated and controlled by the primary station. The central station always acts as the primary station and the substation as the secondary station.

6.2.1.1.2.3 Creating a PLC project and integrating PLC libraries

✓ Start TwinCAT XAE.

1. A wizard opens with **File > New > Project**. In this window you select **TwinCAT Project** and the **TwinCAT XAE Project** template.
2. Go to the PLC node in the Solution Explorer and select **Add New Item from** the context menu. Select a **Standard PLC Project** in the wizard.
3. In addition to the automatically created MAIN, add another programming block. Go to the **POU** folder and from there via the context menu to **Add POU**. Select the type **Program** and call it *P_MAIN_LowSpeed*.
4. Add another program block and name it *P_SerialComm_HighSpeed*.

5. Configure 2 tasks in the task configuration under System: a fast one (T#1ms) and a slow one (T#10ms). Assign the program block *P_MAIN_HighSpeed to* the fast task and the *P_MAIN_LowSpeed* to the slow task (see figure).



6. In the PLC Project node the library management can be found in the sub-node References. There you select the library **Tc2_IEC60870_5_10x.compiled-library** in the context menu via **Add Library** and confirm this.

⇨ Now all requirements are fulfilled to work through the tutorial step by step in configuration and programming.

6.2.1.1.2.4 The fast PLC task

Add the following PLC code to the declaration part:

```
PROGRAM P_SerialComm_HighSpeed
VAR
     fbSerialLineCtrl    : FB_IEC870_SerialLineCtrl := (
                Mode        := SERIALLINEMODE_PC_COM_PORT, (*SERIALLINEMODE_KL6_5B_STANDARD *)
                Baudrate    := 19200,
                NoDatabits      := 8,
                Parity      := PARITY_EVEN,
                Stopbits    := 1,
                Handshake       := HANDSHAKE_NONE,
                ContinousMode   := FALSE );

    serial_in           AT%IB4000       : PcComInData;
    serial_out      AT%QB4000       : PcComOutData;

    KL6_in      AT%IB4100       : KL6inData5B;
    KL6_out     AT%QB4100       : KL6outData5B;

    hSerial     : T_HSERIALCTRL;
END_VAR
```

and in the program code:

```
fbSerialLineCtrl(       pComIn      := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD,
ADR( serial_in ), ADR( KL6_in ) ),
                    pComOut     := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR(
 serial_out ), ADR( KL6_out ) ),
                    SizeComIn       := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD,
SIZEOF( serial_in ), SIZEOF( KL6_in ) ),
                    hSerial     := hSerial );
```

A TwinCAT System Manager configuration suitable for this sample can be found on the sample overview page. The mode variable can be used to switch between two communication paths.

Communication via standard PC COMx interface

- In this case the mode parameter is set to the value: **SERIALLINEMODE_PC_COM_PORT**.
- At the TwinCAT System Manager the *serial_in* and *serial_out* variables are bound to the corresponding IO variables of the serial interface.
- The interface will and has to be configured within the TwinCAT System Manager (Baudrate, Parity etc.). Other communication parameters at the FB_IEC870_SerialLineCtrl function block are not relevant at this mode.

Communication via serial Beckhoff Bus Terminals KL6xxx

- In this case the mode parameter is set to the value: **SERIALLINEMODE_KL6_5B_STANDARD**.
- At the TwinCAT System Manager the *KL6_in* and *KL6_out* variables are bound to the corresponding IO variables of the serial terminal KL6xxx.
- The interface is configured within the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrlfunction block. Communication parameters like baud rate, parity etc. have to be set via this function block.

Communication via serial Beckhoff Bus Terminals EL6xxx

```
PROGRAM P_SerialComm_HighSpeed
VAR
    fbSerialLineCtrl : FB_IEC870_SerialLineCtrl := ( Mode := SERIALLINEMODE_EL6_22B );

    EL6_in AT%IB4100 : EL6inData22B;
    EL6_out AT%QB4100 : EL6outData22B;
    hSerial : T_HSERIALCTRL;
END_VAR
```

```
fbSerialLineCtrl( pComIn := ADR( EL6_in ),
        pComOut := ADR( EL6_out ),
        SizeComIn := SIZEOF( EL6_in ),
        hSerial := hSerial );
```

- In this case the mode parameter is set to the value: **SERIALLINEMODE_EL6_22B**.

- At the TwinCAT System Manager the *EL6_in a*nd *EL6_out* variables are bound to the corresponding IO variables of the serial terminal EL6xxx.
- The interface will and has to be configured within the TwinCAT System Manager (Baudrate, Parity etc.). Other communication parameters at the FB_IEC870_SerialLineCtrl function block are not relevant at this mode.

6.2.1.1.2.5 Defining and configuring an application object database of controling station

Application objects = single points, double points, measured values, short floating point values, etc.

In this sample, the commands were configured so that the process data in the control direction (commands) are in the same memory area but on a different byte/bit offset than the data of the information in the monitoring direction. If required the commands may have the same byte/bit offset as the information in monitoring direction.

Sample:
C_SC_NA_1 with IOA = 10 on the same byte/bit offset as M_SP_NA_1 with IOA = 100 (both byte offset = 100 and bit offset = 0). If the value of M_SP_NA_1 changes, a new C_SC_NA_1 command is triggered.

As an example we will configure the following application objects as part of the introductory project:

ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset Byte	Offset Bit	Process data width in the TwinCAT PLC
M_SP_NA_1	100	General interrogation	0	Flags	100	0	1-bit
M_SP_TA_1	101	General interrogation	0	Flags	100	1	1-bit
M_SP_TB_1	102	General interrogation	0	Flags	100	2	1-bit
M_DP_NA_1	200	General interrogation	0	Flags	200	0	2 bits
M_DP_TA_1	201	General interrogation	0	Flags	200	2	2 bits
M_DP_TB_1	202	General interrogation	0	Flags	200	4	2 bits
M_ST_NA_1	300	General interrogation	0	Flags	300	0	1 byte
M_ST_TA_1	301	General interrogation	0	Flags	301	0	1 byte
M_ST_TB_1	302	General interrogation	0	Flags	302	0	1 byte
M_BO_NA_1	400	General interrogation	0	Flags	400	0	4 bytes
M_BO_TA_1	401	General interrogation	0	Flags	404	0	4 bytes
M_BO_TB_1	402	General interrogation	0	Flags	408	0	4 bytes
M_ME_NA_1	500	General interrogation	0	Flags	500	0	2 bytes
M_ME_TA_1	501	General interrogation	0	Flags	502	0	2 bytes
M_ME_TD_1	502	General interrogation	0	Flags	504	0	2 bytes
M_ME_NB_1	600	General interrogation	0	Flags	600	0	2 bytes
M_ME_TB_1	601	General interrogation	0	Flags	602	0	2 bytes
M_ME_TE_1	602	General interrogation	0	Flags	604	0	2 bytes
M_ME_NC_1	700	General interrogation	0	Flags	700	0	4 bytes
M_ME_TC_1	701	General interrogation	0	Flags	704	0	4 bytes
M_ME_TF_1	702	General interrogation	0	Flags	708	0	4 bytes
M_IT_NA_1	800	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	800	0	4 bytes

ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset Byte	Offset Bit	Process data width in the TwinCAT PLC
M_IT_TA_1	801	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	804	0	4 bytes
M_IT_TB_1	802	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	808	0	4 bytes
Commands							
C_SC_NA_1	10	-	0	Flags	2100	0	1-bit
C_SC_NA_1	11	-	0	Flags	2100	1	1-bit
C_SC_TA_1	12	-	0	Flags	2100	2	1-bit
C_DC_NA_1	20	-	0	Flags	2200	0	2-bit
C_DC_NA_1	21	-	0	Flags	2200	2	2-bit
C_DC_TA_1	22	-	0	Flags	2200	4	2-bit
C_BO_NA_1	40	-	0	Flags	2400	0	4 bytes
C_BO_NA_1	41	-	0	Flags	2404	0	4 bytes
C_BO_TA_1	42	-	0	Flags	2408	0	4 bytes
C_SE_NA_1	50	-	0	Flags	2500	0	2 bytes
C_SE_NA_1	51	-	0	Flags	2502	0	2 bytes
C_SE_TA_1	52	-	0	Flags	2504	0	2 bytes
C_SE_NB_1	60	-	0	Flags	2600	0	2 bytes
C_SE_NB_1	61	-	0	Flags	2602	0	2 bytes
C_SE_TB_1	62	-	0	Flags	2604	0	2 bytes
C_SE_NC_1	70	-	0	Flags	2700	0	4 bytes
C_SE_NC_1	71	-	0	Flags	2704	0	4 bytes
C_SE_TC_1	72	-	0	Flags	2708	0	4 bytes

Declaring a database variable

The application object database is an array variable of type ST_IEC870_5_101AODBEntry. Each array element corresponds to an application object. The array elements are not manipulated directly but through specially provided functions and a database handle (table handle). The database handle must be initialized via a F_iecCreateTableHnd function call before it can be used. During this process the array elements are linked as a hash table. With a larger number of data points the hash table enables faster access to individual data points.

The maximum number of application objects is freely selectable and is only limited by the available memory. During PLC programming you have to specify a constant maximum number. The maximum number of application objects cannot be changed at runtime. In our sample 50 application objects are declared. This number is sufficient for most applications. Please note that many application objects require adequate memory and runtime resources.

Define the following variables in P_MAIN_LowSpeed:

```
PROGRAM P_MAIN_LowSpeed
VAR
     AODB       : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    hTable      : T_HAODBTable;
END_VAR
```

Configuring application objects

The required application objects are configured during program runtime. Amongst other things, the object type (M_SP_NA_1, M_DP_NA_1, M_ST_NA_1, etc.), the common ASDU address, the object address and further object parameters are defined during the configuration.

After initialization of the database handle is the application object database (database array) is empty and must be filled with the required data (data points). The configuration of the data points in the central station must correspond to the configuration of the data points in the substation! This means that data points configured in the central station must be of the same type, with the same common ASDU address and with the same information object address as in the substation. Other parameters such as the mapping range, byte and bit offset can be configured as desired.

The following functions are available for manipulating the application database:

Function	Description
F_iecCreateTableHnd	Initializes the hash table handle
F_iecAddTableEntry	Configures and inserts a new hash table entry
F_iecRemoveTableEntry	Removes a hash table entry
F_iecLookupTableEntry	Checks whether a certain hash table entry exists
F_iecGetPosOfTableEntry	Determines the linear position of a hash table entry

The database handle must be transmitted to the function via VAR_IN_OUT. Configuration is usually carried out once during PLC program start-up via an Init routine.

To configure the application objects at program start, the following PLC code is added to P_MAIN_LowSpeed:

```
PROGRAM P_MAIN_LowSpeed
VAR
    AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    hTable      : T_HAODBTable;

     init        : BOOL := TRUE;
    initError           : UDINT;
    asduAddr        : UDINT := 7;
END_VAR
```

```
IF init THEN
    init := FALSE;

    initError := F_iecCreateTableHnd( ADR( AODB ), SIZEOF( AODB ), hTable );
    IF initError <> 0 THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG,
            'F_iecCreateTableHnd() error: %s',
            DWORD_TO_HEXSTR( initError, 8, FALSE ) );
        RETURN;
    END_IF


  (* Monitored Single Points *)
 initError := F_iecAddTableEntry( M_SP_NA_1, asduAddr, 100, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
100, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_SP_NA_1, asduAddr, 101, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 100, 1, 0, hTable );
    initError := F_iecAddTableEntry( M_SP_TB_1, asduAddr, 102, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 100, 2, 0, hTable );
  (* Double Points*)
 initError := F_iecAddTableEntry( M_DP_NA_1, asduAddr, 200, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
200, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_DP_NA_1, asduAddr, 201, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 200, 2, 0, hTable );
    initError := F_iecAddTableEntry( M_DP_TB_1, asduAddr, 202, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 200, 4, 0, hTable );
  (* Regulating step value *)
 initError := F_iecAddTableEntry( M_ST_NA_1, asduAddr, 300, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
300, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ST_NA_1, asduAddr, 301, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 301, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ST_TB_1, asduAddr, 302, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 302, 0, 0, hTable );
  (* 32 bit string *)
 initError := F_iecAddTableEntry( M_BO_NA_1, asduAddr, 400, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
400, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_BO_NA_1, asduAddr, 401, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 404, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_BO_TB_1, asduAddr, 402, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 408, 0, 0, hTable );
  (* Measured value, normalized value *)
 initError := F_iecAddTableEntry( M_ME_NA_1, asduAddr, 500, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
500, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_NA_1, asduAddr, 501, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 502, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_TD_1, asduAddr, 502, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 504, 0, 0, hTable );
  (* Measured value, scaled value *)
 initError := F_iecAddTableEntry( M_ME_NB_1, asduAddr, 600, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
600, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_NB_1, asduAddr, 601, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 602, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_TE_1, asduAddr, 602, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 604, 0, 0, hTable );
  (* Measured value , short floating point value *)
 initError := F_iecAddTableEntry( M_ME_NC_1, asduAddr, 700, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
700, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_NC_1, asduAddr, 701, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 704, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_ME_TF_1, asduAddr, 702, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMOR
Y, 708, 0, 0, hTable );
  (* Integrated totals *)
 initError := F_iecAddTableEntry( M_IT_NA_1, asduAddr, 800, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMORY,
 800, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_IT_NA_1, asduAddr, 801, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMO
RY, 804, 0, 0, hTable );
    initError := F_iecAddTableEntry( M_IT_TB_1, asduAddr, 802, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMO
RY, 808, 0, 0, hTable );

  (* Single commands *)
 initError := F_iecAddTableEntry( C_SC_NA_1, asduAddr, 10, 0, 0, MAP_AREA_MEMORY, 2100, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_SC_NA_1, asduAddr, 11, 0, 0, MAP_AREA_MEMORY, 2100, 1, 0, hTa
ble );
    initError := F_iecAddTableEntry( C_SC_TA_1, asduAddr, 12, 0, 0, MAP_AREA_MEMORY, 2100, 2, 0, hTa
ble );
  (* Double commands *)
 initError := F_iecAddTableEntry( C_DC_NA_1, asduAddr, 20, 0, 0, MAP_AREA_MEMORY, 2200, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_DC_NA_1, asduAddr, 21, 0, 0, MAP_AREA_MEMORY, 2200, 2, 0, hTa
```

```
ble );
    initError := F_iecAddTableEntry( C_DC_TA_1, asduAddr, 22, 0, 0, MAP_AREA_MEMORY, 2200, 4, 0, hTa
ble );
  (* 32 bit string commands *)
 initError := F_iecAddTableEntry( C_BO_NA_1, asduAddr, 40, 0, 0, MAP_AREA_MEMORY, 2400, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_BO_NA_1, asduAddr, 41, 0, 0, MAP_AREA_MEMORY, 2404, 0, 0, hTa
ble );
    initError := F_iecAddTableEntry( C_BO_TA_1, asduAddr, 42, 0, 0, MAP_AREA_MEMORY, 2408, 0, 0, hTa
ble );
  (* Set point, normalized values*)
 initError := F_iecAddTableEntry( C_SE_NA_1, asduAddr, 50, 0, 0, MAP_AREA_MEMORY, 2500, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_SE_NA_1, asduAddr, 51, 0, 0, MAP_AREA_MEMORY, 2502, 0, 0, hTa
ble );
    initError := F_iecAddTableEntry( C_SE_TA_1, asduAddr, 52, 0, 0, MAP_AREA_MEMORY, 2504, 0, 0, hTa
ble );
  (* Set point, scaled values *)
 initError := F_iecAddTableEntry( C_SE_NB_1, asduAddr, 60, 0, 0, MAP_AREA_MEMORY, 2600, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_SE_NB_1, asduAddr, 61, 0, 0, MAP_AREA_MEMORY, 2602, 0, 0, hTa
ble );
    initError := F_iecAddTableEntry( C_SE_TB_1, asduAddr, 62, 0, 0, MAP_AREA_MEMORY, 2604, 0, 0, hTa
ble );
  (* Set point, short floating point values *)
 initError := F_iecAddTableEntry( C_SE_NC_1, asduAddr, 70, 0, 0, MAP_AREA_MEMORY, 2700, 0, 0, hTable
 );
    initError := F_iecAddTableEntry( C_SE_NC_1, asduAddr, 71, 0, 0, MAP_AREA_MEMORY, 2704, 0, 0, hTa
ble );
    initError := F_iecAddTableEntry( C_SE_TC_1, asduAddr, 72, 0, 0, MAP_AREA_MEMORY, 2708, 0, 0, hTa
ble );

END_IF
```

6.2.1.1.2.6 Mapping of PLC and IEC process data

The TwinCAT PLC process data are cyclically mapped (copied) into the IEC process data (application objects) and vice versa at program runtime. Up to four process data areas (IO inputs, IO outputs, flag range, data area) can be declared in the PLC program as buffer variables for the mapping of the IEC<->PLC process data. The byte size of the buffers is freely selectable and may be different for each area. Unused ranges need not necessarily be declared.

In our introductory example we declare 4 PLC process data areas with 3000 bytes each:

```
PROGRAM P_MAIN_LowSpeed
VAR
    AODB    : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    hTable  : T_HAODBTable;

    init    : BOOL := TRUE;
    initError       : UDINT;
    asduAddr    : UDINT := 7;


     inputs AT%IB0   : ARRAY[0..2999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..2999] OF BYTE;
    memory AT%MB0   : ARRAY[0..2999] OF BYTE;
    data        : ARRAY[0..2999] OF BYTE;

END_VAR
```

How the process data are to be mapped at runtime is specified during configuration of the application objects via the F_iecAddTableEntry [▶ 555] function.

See also in: Definition and configuration of application objects [▶ 673].

The buffer variables were now declared as byte arrays. In order to improve access to the required data we define the individual variables a second time and allocate them to the corresponding byte/bit offset addresses. In case of a change in the byte array, the corresponding individual variable will be changed and inverted at the same time. However, this is not absolutely necessary. The bytes/bits of the byte array buffer variables can be accessed directly.

```
VAR_GLOBAL(* Memory offset 0..99 unused *)
  (* Single points *)
 msgSingle_0     AT%MX100.0 : BOOL;
```

```
    msgSingle_1     AT%MX100.1 : BOOL;
    msgSingle_2     AT%MX100.2 : BOOL;

  (* Double points *)
    (*      Bit 0..1 = first double point,
         Bit 2..3 = second double point,
         Bit 4..5 = third double point,
         Bit 6..7 = fourth double point *)
    msgDouble_0     AT%MB200    : BYTE;

  (* Regulating step values *)
    msgStep_0           AT%MB300    : BYTE;
    msgStep_1           AT%MB301    : BYTE;
    msgStep_2           AT%MB302    : BYTE;

  (* 32 bit strings *)
 msgBitStr_0     AT%MD400    : DWORD;
    msgBitStr_1     AT%MD404    : DWORD;
    msgBitStr_2     AT%MD408    : DWORD;

  (* Measured values, normalized values *)
 msgNormalized_0     AT%MW500    : WORD;
    msgNormalized_1     AT%MW502    : WORD;
    msgNormalized_2     AT%MW504    : WORD;

  (* Measured values, scaled values *)
 msgScaled_0     AT%MW600    : INT;
    msgScaled_1     AT%MW602    : INT;
    msgScaled_2     AT%MW604    : INT;

  (* Measured values, short floating point values *)
 msgFloating_0   AT%MD700    : REAL;
    msgFloating_1   AT%MD704    : REAL;
    msgFloating_2   AT%MD708    : REAL;

  (* Integrated totals *)
 msgTotal_0      AT%MD800    : UDINT;
    msgTotal_1      AT%MD804    : UDINT;
    msgTotal_2      AT%MD808    : UDINT;

  (* Single commands *)
 cmdSingle_0     AT%MX2100.0 : BOOL;
    cmdSingle_1     AT%MX2100.1 : BOOL;
    cmdSingle_2     AT%MX2100.2 : BOOL;

  (* Double commands *)
    (*      Bit 0..1 = first double command,
         Bit 2..3 = second double command,
         Bit 4..5 = third double command,
         Bit 6..7 = fourth double command *)
 cmdDouble_0     AT%MB2200       : BYTE;


  (* 32 bit string commands *)
 cmdBitStr_0     AT%MD2400       : DWORD;
    cmdBitStr_1     AT%MD2404       : DWORD;
    cmdBitStr_2     AT%MD2408       : DWORD;

  (* Set point, normalized values *)
 cmdNormalized_0     AT%MW2500       : WORD;
    cmdNormalized_1     AT%MW2502       : WORD;
    cmdNormalized_2     AT%MW2504       : WORD;

  (* Set point, scaled values *)
 cmdScaled_0     AT%MW2600       : INT;
    cmdScaled_1     AT%MW2602       : INT;
    cmdScaled_2     AT%MW2604       : INT;

  (* Set point, short floating point values *)
 cmdFloating_0   AT%MD2700       : REAL;
    cmdFloating_1   AT%MD2704       : REAL;
    cmdFloating_2   AT%MD2708       : REAL;
END_VAR
```

Mapping of the IEC<->PLC process data in the central station

Process data in monitoring direction (slave->master information)

Example 1

Single point information (M_SP_NA_1) with the IOA = 100, PLC memory area, byte offset = 100, bit offset = 0.

Controlled station -> ... -> Controlling station FB -> memory[100].0 == msgSingle_0

Example 2

Measured value, short floating point value (M_ME_NC_1) with the IOA = 700, PLC memory area, byte offset = 700, bit offset = 0 (irrelevant).

Controlled station -> ... -> Controlling station FB -> memory[700..703] == msgFloating_0

Process data in control direction (master->slave commands)

Example 1

Single command state (C_SC_NA_1) with the IOA = 10, PLC memory area, byte offset = 2100, bit offset = 0.

cmdSingle_0 == memory[2100].0 -> Controlling station FB -> ... -> Controlled station

Example 2

Set point, short floating point value (C_SE_NC_1) with the IOA = 70, PLC memory area, byte offset = 2700, bit offset = 0 (irrelevant).

cmdFloating_0 == memory[2700..2703] -> Controlling station FB -> ... -> Controlled station

6.2.1.1.2.7 Declaring and calling an instance of the IEC60870-5-101 controlling station

The entire functionality of a central station is encapsulated in the function block FB_IEC870_5_101Master. A connection to the substation can be established with an instance. For a further connection, declare a further instance of the function block.

Insert the following PLC code in the declaration part of P_MAIN_LowSpeed:

```
PROGRAM P_MAIN_LowSpeed
VAR
    AODB        : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
    hTable      : T_HAODBTable;

    init        : BOOL := TRUE;
    initError       : UDINT;
    asduAddr        : UDINT := 7;

    inputs AT%IB0   : ARRAY[0..2999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..2999] OF BYTE;
    memory AT%MB0   : ARRAY[0..2999] OF BYTE;
    data        : ARRAY[0..2999] OF BYTE;

     bEnable         : BOOL := TRUE;
    client      : FB_IEC870_5_101Master;

END_VAR
```

and the instance is called in the program part:

```
IF init THEN
    init := FALSE;
...




ELSE
...
```

```
    client(     pAOEntries := ADR( AODB ),
        cbAOEntries := SIZEOF( AODB ),
        pInputs := ADR( inputs ),
        cbInputs := SIZEOF( inputs ),
        pOutputs := ADR( outputs ),
        cbOutputs := SIZEOF( outputs ),
        pMemory := ADR( memory ),
        cbMemory := SIZEOF( memory ),
        pData := ADR( data ),
        cbData := SIZEOF( data ),
        bEnable := bEnable,
        hSerial := P_SerialComm_HighSpeed.hSerial,
        hTable := hTable );
...
END_IF
```

6.2.1.1.2.8 IEC60870-5-101 protocol parameters

The behavior of the central station can be adapted to the requirements of the substation via the IEC60870-5-101 protocol parameters. Most parameters have preallocated default values that do not have to be changed.

In our sample we configure the connection address, the octet length of the connection address and the cycle time in which the Class 1 and Class 2 data should be polled.

```
IF init THEN
    init := FALSE;
...
  (*Configure protocol parameter *)
 client.protPara.linkAddr := 220;(* link address of remote slave *)
 client.protPara.eLinkAddrSize := eIEC870_LinkAddr_TwoOctets; (* link addres octet size *)
 client.protPara.tClass1Poll := T#0ms; (* poll class 1 data with max. speed *)
 client.protPara.tClass2Poll := T#0ms; (* poll class 2 data with max. speed *)
...

ELSE
    client( pInputs := ADR( inputs ),
        cbInputs := SIZEOF( inputs ),
        pOutputs := ADR( outputs ),
...

END_IF
```

The documentation for all transfer protocol parameters can be found here: ST_IEC870_5_101PotocolParams [▶ 640].

6.2.1.1.2.9 System parameters

The common ASDU address and the user functions of the central station are configured via the system parameters.

In our introduction we configure the following system parameters:

- The common ASDU address is set to 7. (*asduAddr*);
- The octet length of the cause of transfer is set to 2 (1 octet for COT + 1 octet for source address) (*eCOTSize*);
- The octet length of the common ASDU address is set to 2 (*eAsduAddrSize*);
- The octet length of the information object address is set to 3 (*eObjAddrSize*);
- Logging of debugging messages in the application log is activated (*dbgMode*). Changes in the device status and error messages of the link layer are logged;

Add the following PLC code to your PLC project:

```
IF init THEN
    init := FALSE;
...

     client.sysPara.asduAddr := 7;
    client.sysPara.asduFmt.eCOTSize := eIEC870_COT_TwoOctets;(* cause of transfer octet size *)
 client.sysPara.asduFmt.eAsduAddrSize := eIEC870_AsduAddr_TwoOctets;
(* common ASDU address octet size *)
```

```
 client.sysPara.asduFmt.eObjAddrSize := eIEC870_ObjAddr_ThreeOctets;
(* information object address octet size *)
 client.sysPara.dbgMode := IEC870_DEBUGMODE_DEVSTATE OR IEC870_DEBUGMODE_LINKERROR; (* IEC870_DEBUGM
ODE_ASDU OR IEC870_DEBUGMODE_LINKLAYER *)

...
ELSE
    client( pInputs := ADR( inputs ),
        cbInputs := SIZEOF( inputs ),
        pOutputs := ADR( outputs ),
...

END_IF
```

The documentation for all system parameters can be found here: ST_IEC870_5_101SystemParams [▶ 588].

6.2.1.1.2.10 Initialisation sequence

```
client.acqPara.arrInitSeq[0] := eIEC870_ISTEP_TEST; (* Send test command *)
client.acqPara.arrInitSeq[1] := eIEC870_ISTEP_CLOCK; (* Send clock synchronization command *)
client.acqPara.arrInitSeq[2] := eIEC870_ISTEP_GENRO; (* Send general interrogation command *)
client.acqPara.arrInitSeq[3] := eIEC870_ISTEP_CORO; (* Send counter interrogation command *)
client.acqPara.arrInitSeq[5] := eIEC870_ISTEP_UNUSED; (* not used *)
```

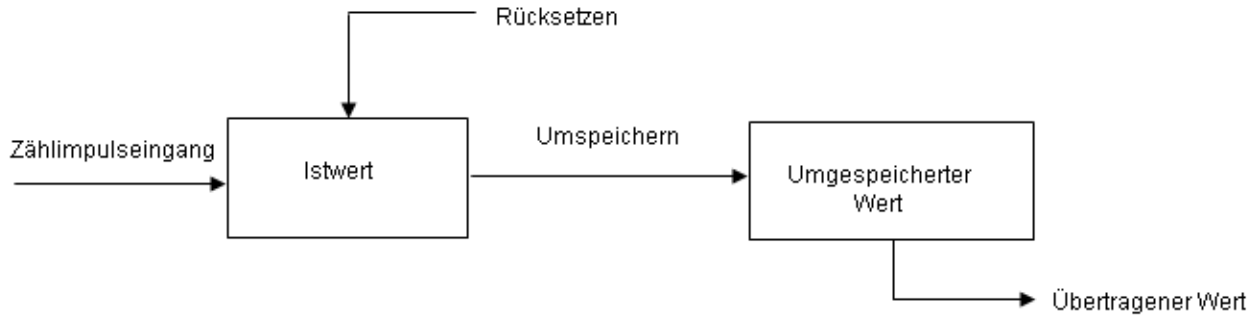
6.2.1.1.2.11 Station interrogation

The station interrogation command is initiated by the central station. The ID field of the command also contains the group (1 to 16 or general). The substation transfers the application objects associated with this group to the central station with cause of transfer <*20*> to <*36*>. Application objects with time tags are transferred without time tags.

```
client.acqPara.arrGenro[0].tPollCycle := T#60s;
client.acqPara.arrGenro[0].eQOI := eIEC870_QOI_INROGEN;
client.acqPara.arrGenro[0].bEnable := TRUE;
```

6.2.1.1.2.12 Transfer of integrated totals (counter interrogation)

General model for the transmission of integrated totals:



The actual values are added by counters. Via a freeze command that is either received by the central station or generated locally (in the substation), the actual values can be frozen (copied) periodically into frozen values. After freezing, the recorded value is either reset to null (logging of incremental values), or the counter continues adding up (logging of counter values).

Application objects with counts are assigned to groups. The groups are frozen individually, reset, or transferred. The central station sends counter interrogation commands to the substation. The task to be carried out (FRZ) and the group (RQT) are specified in an ID field of the command (QCC).

The allocation of the application objects to the individual groups (1 to 4 or general) is specified via the group flag parameter during configuration. There are four operation modes for recording counter readings and incremental values. Each mode includes notes about the configuration of the system parameters or the application objects.

Mode A: local freeze with spontaneous transfer

The substation internally initiates freeze or freeze with reset. The frozen counts are transferred spontaneously, once the freeze or freeze with reset function was executed. In this mode the central station does not issue any count query commands.

Configuration of the system parameters:

Configuration of the application objects:

Mode B: Local freeze with counter interrogation

The substation internally initiates freeze or freeze with reset. The central station queries the frozen counts via count query commands. In this case the central station must not use freeze or freeze with reset in the command ID field (FRZ=0). The counts are queried generally or in groups 1 to 4.

Configuration of the system parameters:

Configuration of the application objects:

Mode C: The central station initiates freeze, freeze with reset, or reset

The central station periodically issues a count query command to the substation for controlling the freeze or (and) reset process. This command does not result in a count transfer. The central station sends a subsequent count query command for collecting the frozen counts. This is similar to operating mode B.

Configuration of the system parameters:

Configuration of the application objects:

Operating mode D: The central station initiates freeze and (or) reset, and the frozen values are transferred spontaneously

This mode is a combination of the count command from the central station as in mode C and spontaneous transfer of the counts as in mode A.

Configuration of the system parameters:

Configuration of the application objects:

6.2.1.1.2.13 Testing the communication

A simple simulation of the commands in the control direction is activated by setting the *bExecuteCmd* variable to TRUE and deactivated by setting it to FALSE. With an active connection, a single command (C_SC_NA_1, IOA = 10) is cyclically transmitted to the substation every 10 seconds in the sample used.

```
PROGRAM MAIN
VAR
    ...

    bExecuteCmd : BOOL;
    timer : TON;

    ...
END_VAR
```

```
...

(* Simple command simulation *)
timer( IN := bExecuteCmd, PT := T#10s ); (* Send cyclic command *)
IF timer.Q THEN
    timer( IN := FALSE );
    cmdSingle_0 := NOT cmdSingle_0; (* toggle single command ON<->OFF *)

(*      cmdDouble_0 := SEL( cmdDouble_0 = 1, 1, 2 );

    cmdBitStr_0 := cmdBitStr_0 + 1;

    cmdNormalized_0 := cmdNormalized_0 + 2;

    cmdScaled_0 := cmdScaled_0 + 4;
```

```
    cmdFloating_0 := cmdFloating_0 + 1.2; *)
END_IF

...
```

6.2.1.1.2.14 Protocol and data transmission errors

The station error messages are stored in a FIFO memory. Up to 10 error messages can be buffered. In the case of fatal communication errors (e.g. errors of the link layer, the checksum of the frame is wrong) the connection is interrupted and must be re-established. Errors in the application layer (e.g. the ASDU transmit buffer has overflowed due to too many frames) are only logged and do not lead to interruption of the connection. Even with these errors it is still possible to interrupt the connection from the application. Apart from the error code, the error source is also stored in the error message. This makes it easier to locate the error.

Sample

The accumulated error messages of an IEC 60870-5-101 central station can be read out using the following call:

```
PROGRAM MAIN
VAR
...

    client : FB_IEC870_5_101Master;

...
END_VAR
```

```
....

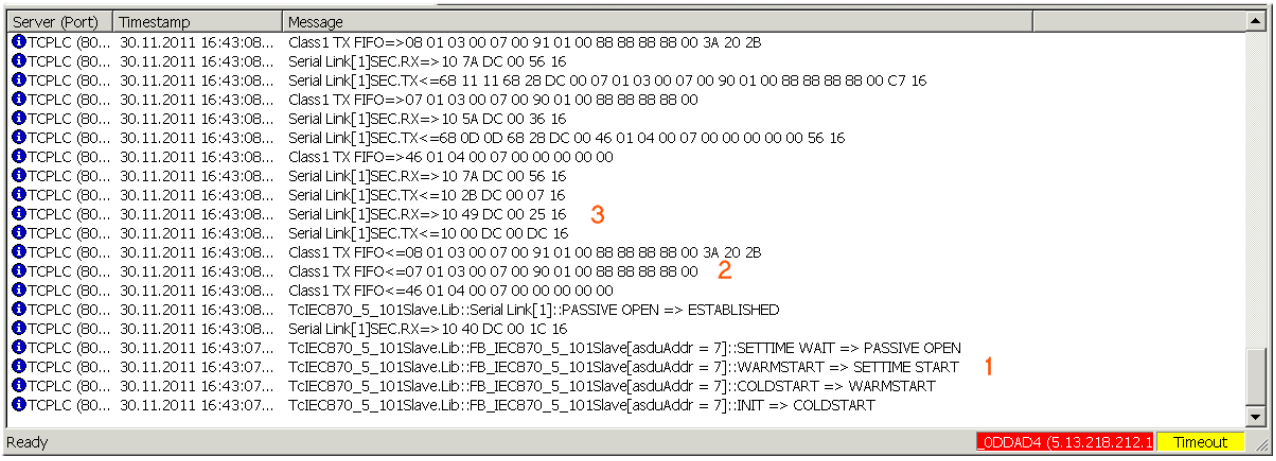

REPEAT
    client.system.device.errors.RemoveError( );
    IF client.system.device.errors.bOk THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG,
            'IEC60870-5-101 master error: 0x%s',
            DWORD_TO_HEXSTR( client.system.device.errors.getError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT client.system.device.errors.bOk
END_REPEAT
...
```

6.2.1.1.2.15 Troubleshooting and debugging

Debugging messages written to the application log facilitate troubleshooting of the system. Currently, three levels of debug messages can be activated in an IEC application. These messages can be activated via the dbgMode system parameter of the control station (ST_IEC870_5_101SystemParams [▶ 588]).

1. Station status messages (dbgMode: IEC870_DEBUGMODE_DEVSTATE);
2. Hexadecimal output of the ASDU's (without link layer control header, dbgMode: IEC870_DEBUGMODE_ASDU). 32 ASDU data bytes per row are output as hexadecimal numbers. Longer ASDUs are spread across several rows;
3. Hexadecimal output of the APDU's (serial port telegrams, dbgMode: IEC870_DEBUGMODE_LINKLAYER). 32 APDU data bytes per row are output as hexadecimal numbers. Like in 2., longer APDUs are spread across several rows;

Optionally, link layer errors can also be output (dbgMode: IEC870_DEBUGMODE_LINKERROR). In order to view the activated debug messages start the TwinCAT System Manager and activate log view. A debugging output is shown below. The three different message types are identified with the respective numbers.



Further diagnostic tools:

- Portmon for Windows (v3.02, Windows Sysinternals);
- Diverse protocol test suite products;

6.2.1.1.2.16 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 685]
- Configuration of the serial KL6xxx Bus Terminals [▶ 686]
- Configuration of the serial EL6xxx Bus Terminals [▶ 686]

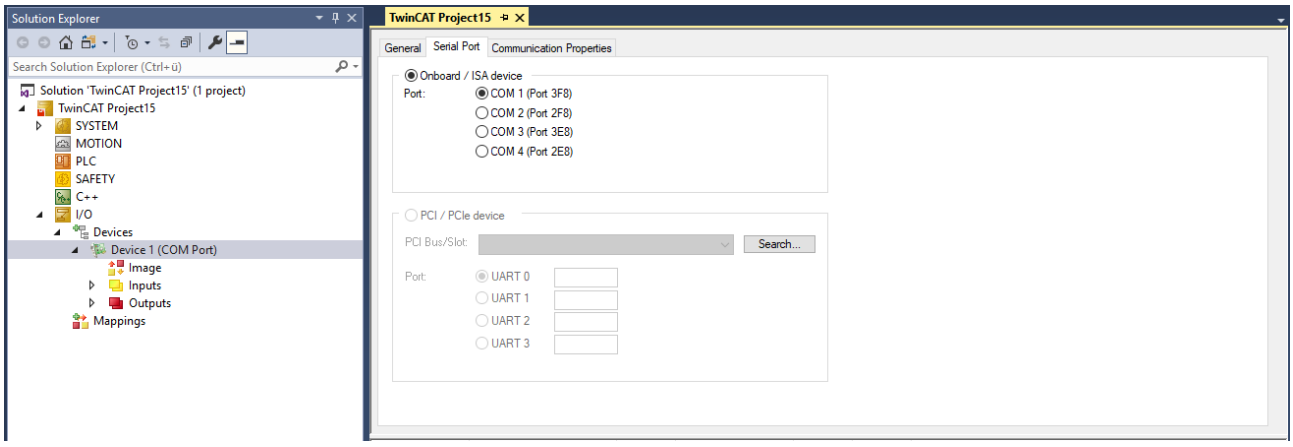
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

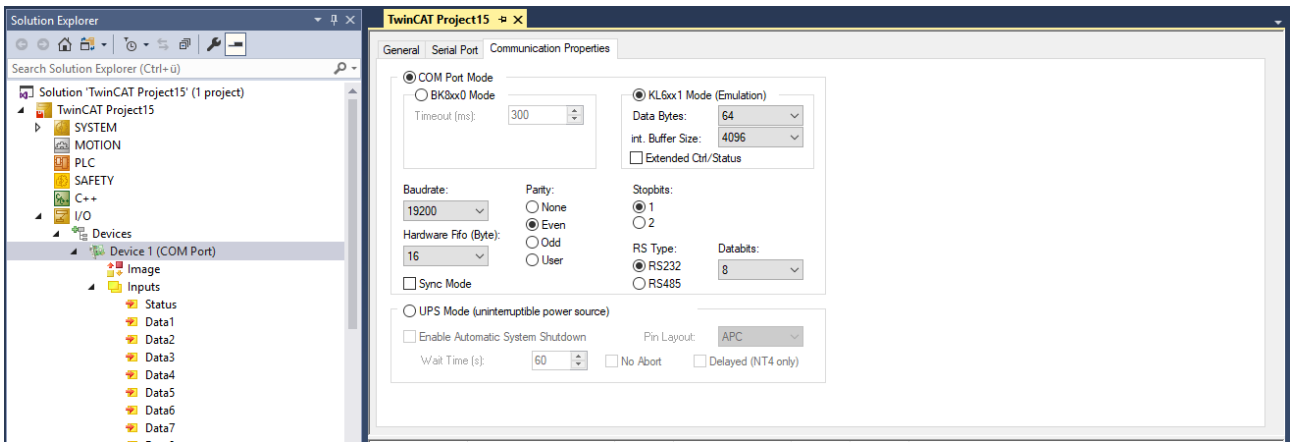
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

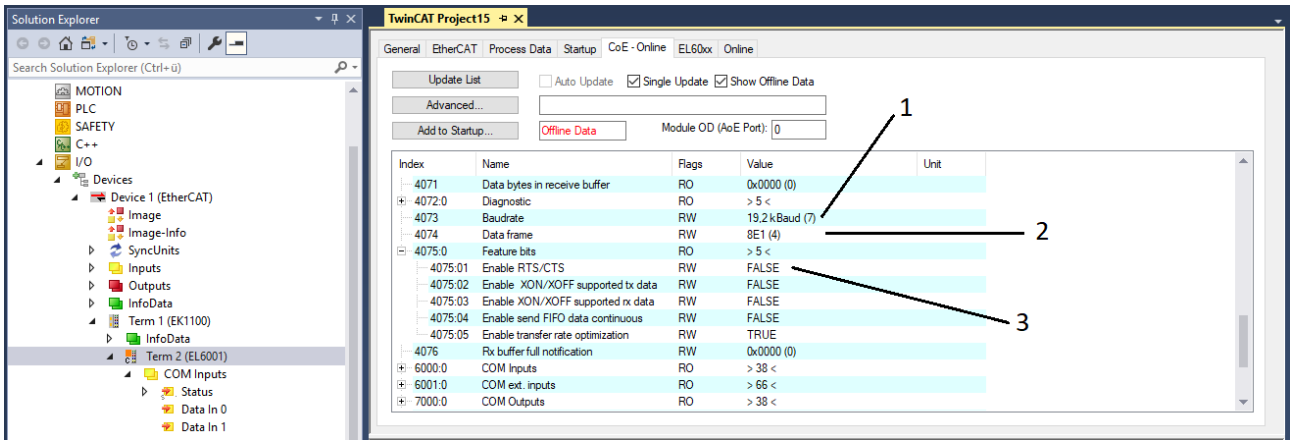
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

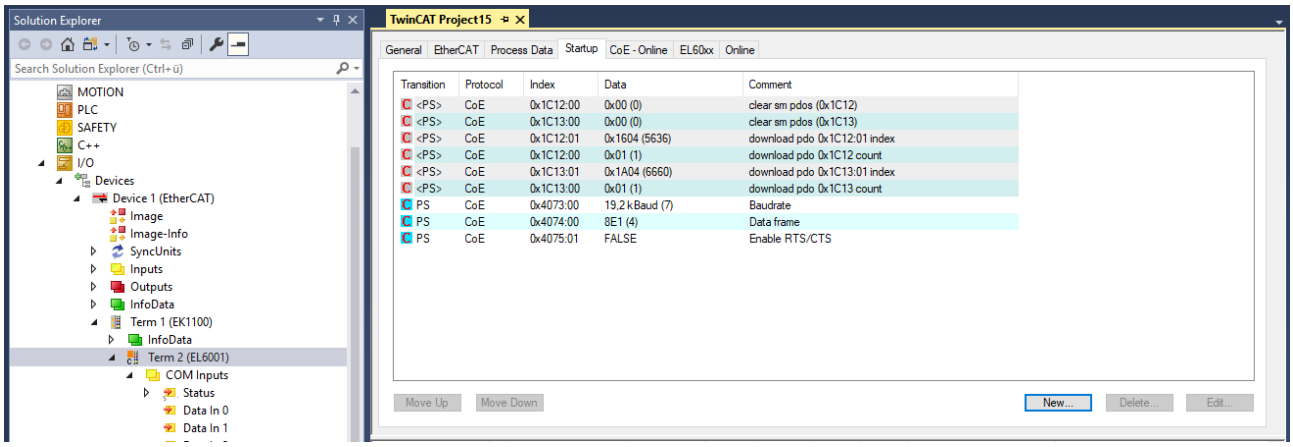
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.2.1.2 Low-Level Interface

6.2.1.2.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-101 Central station low-level sample [▶ 687].

A PLC application that is to communicate with a substation via the "low-level" interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_101TProtocol [▶ 510];
- An instance of the TX/RX data buffer: ST_IEC870_5_101TBuffer [▶ 583];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_101TBufferCtrl [▶ 506];

6.2.1.2.1.1 Download: Central station (master, unbalanced mode)

Simple central station application that uses the TwinCAT IEC 60870-5-101 low-level interface. This sample application can communicate with up to three substations via an RS232 interface. However, in the program only the communication to the first substation is activated.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984606987.zip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**

- Link address: **220 (first substation), 221 (second substation), 222 (third substation).**
- Link address size: **1 octet.**
- Cause of transfer size: **1 octet.**
- Originator address: **not used (0).**
- Common ASDU address: **7 (first substation), 8 (second substation), 9 (third substation).**
- Common ASDU address size: **1 octet.**
- Information object address size: **2 octets.**
- Link mode**: Unbalanced.**

Data point configuration (it should be the same as the data point configuration of all three substations):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200).
- Regulating step value: M_ST_NA_1 (IOA := 300).
- Bitstring value: M_BO_NA_1 (IOA := 400).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10).
- Double command: C_DC_NA_1 (IOA := 20).
- Regulating step command: C_RC_NA_1 (IOA := 30).
- Bitstring command: C_BO_NA_1 (IOA := 40).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70).

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 10 seconds.
- Clock synchronization command (C_CS_NA_1) every 20 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 40 seconds.
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN) every 60 seconds. Counter interrogation Mode A.
- Simulation and transmission of the commands every 60 seconds if the variable **bSimulate** was set to TRUE (see upper data point configuration).
- Reception of spontaneous data.
- Reception of M_EI_NA_1 (end of initialization).

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
aClients	ARRAY OF FB_IEC870_5_101M (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_101MServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
aProcData	ARRAY OF ST_IEC870_5_101MProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_101MSysPara (STRUCT)	Configuration of the system parameters of the central station and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.2.1.2.1.2 Download: Central station (master, balanced mode)

Simple central station application that uses the TwinCAT IEC 60870-5-101 low-level interface. In the sample application it is possible to communicate with a substation via an RS232 interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984609931.zip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220.**
- Link address size: **1 octet.**
- Cause of transfer size: **1 octet.**
- Originator address: **0 (not used).**
- Common ASDU address: **7.**
- Common ASDU address size: **1 octet.**
- Information object address size: **2 octets.**
- Link mode: **Balanced.**

Data point configuration (it should be the same as the data point configuration of the substations):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200).
- Regulating step value: M_ST_NA_1 (IOA := 300).
- Bitstring value: M_BO_NA_1 (IOA := 400).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).

- Single command: C_SC_NA_1 (IOA := 10).
- Double command: C_DC_NA_1 (IOA := 20).
- Regulating step command: C_RC_NA_1 (IOA := 30).
- Bitstring command: C_BO_NA_1 (IOA := 40).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70).

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 10 seconds.
- Clock synchronization command (C_CS_NA_1) every 20 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 40 seconds.
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN) every 60 seconds. Counter interrogation Mode A.
- Simulation and transmission of the commands every 60 seconds if the variable **bSimulate** was set to TRUE (see upper data point configuration).
- Reception of spontaneous data.
- Reception of M_EI_NA_1 (end of initialization).

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbClient	FB_IEC870_5_101M (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_101MServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
stProcData	ST_IEC870_5_101MProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_101MSysPara (STRUCT)	Configuration of the system parameters of the central station and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.2.1.2.1.3 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 690]
- Configuration of the serial KL6xxx Bus Terminals [▶ 691]
- Configuration of the serial EL6xxx Bus Terminals [▶ 691]

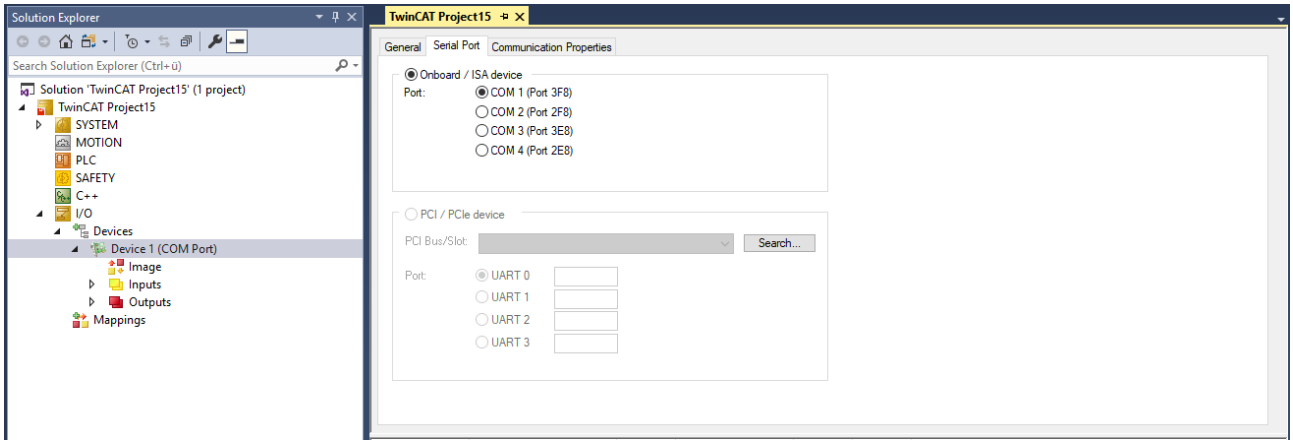
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

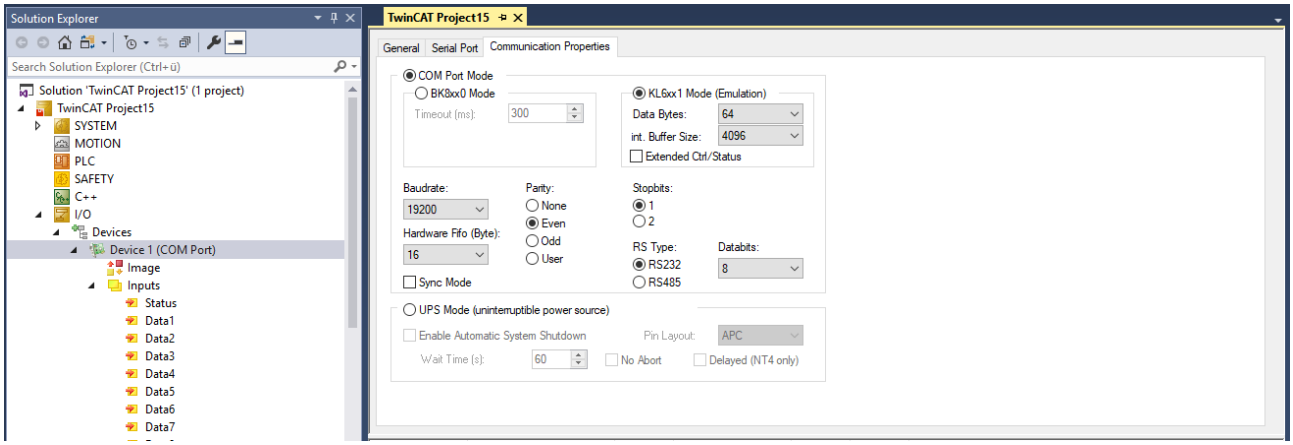
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

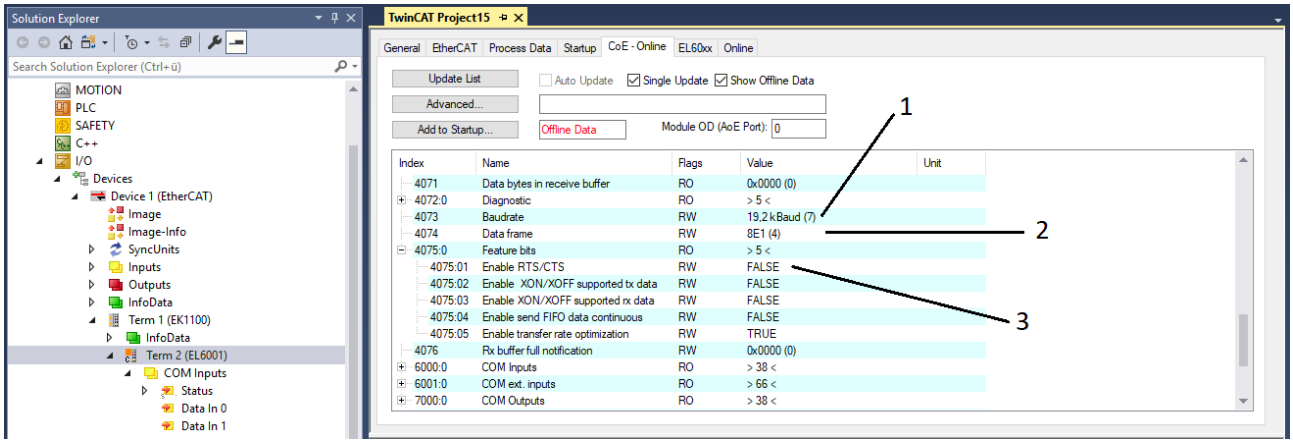
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

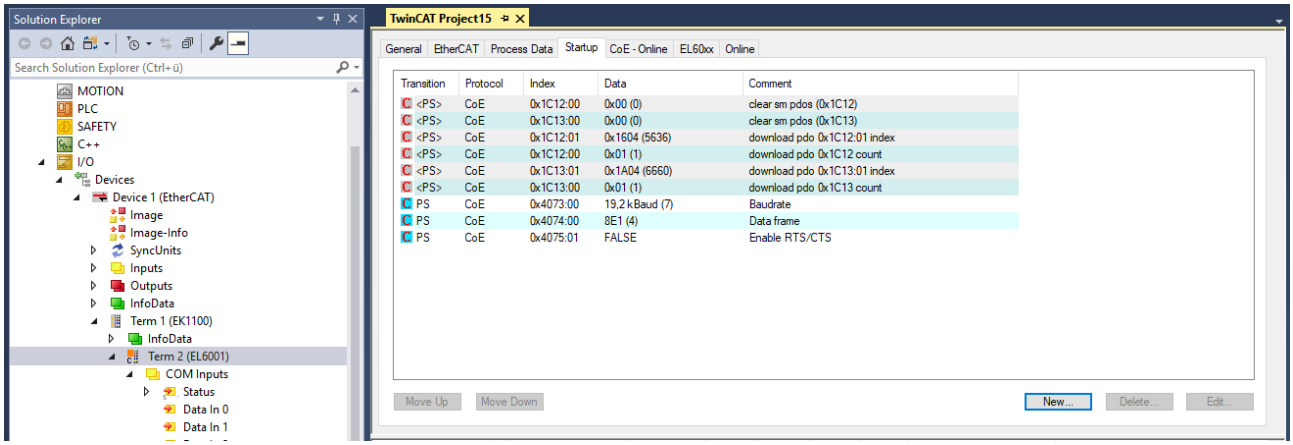
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.2.2 Controlled station

6.2.2.1 High-Level Interface

6.2.2.1.1 Quick start

Simple project including complete sources is to be found here: IEC 60870-5-101 Controlled Station-High-Level example [▶ 693].

Interoperability list is to be found here: Interoperability check list [▶ 30]

Overview of error codes is to be found here: Error codes [▶ 784]

Brief instruction

1. Create a new PLC project and link the PLC library: **Tc2_IEC60870_5_10x**.
2. Apply two PLC tasks, one fast (e.g. with T#1ms cycle time) and one slow (e.g. with T#1ms cycle time). Apply two function blocks (e.g. *P_SerialComm_HighSpeed* and *P_MAIN_LowSpeed*). *P_SerialComm_HighSpeed* is called by the fast and *P_MAIN_LowSpeed* is called by the slow task.
3. Apply an instance of the function block FB_IEC870_SerialLineCtrl [▶ 519] at *P_SerialComm_HighSpeed*, configure it and call it.
   In case of communication via Beckhoff's serial bus terminals or the serial interface of the PC, apply the buffers *KL6inData5B, KL6outData5B* or *PcComInData, PcComOutData* and link with the according IO process data at the TwinCAT System Manager.
4. The instance of the T_HSERIALCTRL [▶ 632] variable is used for exchanging Tx/Rx telegrams between both tasks (programs). Apply this variable for example as a global variable.
5. Configuration of data points: Apply an array variable of ST_IEC870_5_101AODBEntry [▶ 584] type. Every array element corresponds to a data point. Configure the data points via the function F_iecInitAOEntry [▶ 537] at runtime (e.g. at an init step).
6. Apply an instance of the des protocol block FB_IEC870_5_101Slave [▶ 490] at the *P_MAIN_LowSpeed*, configure it and call it.
7. The system and protocol parameter are to be configured according to the parameter of the control station.

6.2.2.1.1.1 Download: Substation Minimal (slave, unbalanced mode)

Simple substation application using the TwinCAT IEC 60870-5-101 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984612875.zip

In the sample project, the following station parameters are set for the substation:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220.**
- Link address size: **2 octets.**
- Cause of transfer size: **2 octets (including originator address).**
- Originator address: **1.**
- Common ASDU address: **7.**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Link mode**: Unbalanced.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100).
- Single command: C_SC_NA_1 (IOA := 10).

Sample implementation of the following services:

- End of initialization (M_EI_NA_1) is sent once.
- Test command (C_TS_NA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN). Counter interrogation Mode C.
- Single command processing (C_SC_NA_1) (direct command execution)
- Sending spontaneous data of the type (M_SP_NA_1, IOA := 100) when the value of the **bSPI** variable changes.

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbServer	FB_IEC870_5_101Slave (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
bSPI	BOOL	The ON/OFF value to be sent is mapped directly from the flag address **AT%MX1.5** of this allocated PLC variable.
bSCS	BOOL	The received ON/OFF value is mapped into this PLC variable directly to the allocated flag address: **AT%MX0.0**.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
F_iecInitAOEntry	STRUCT	In the initialization step, the individual data points of the station are configured once with this function call.

6.2.2.1.2 Tutorial

The tutorial explains how to implement and configure an IEC60870-5-101 substation (slave) in the TwinCAT PLC. You can download the sample on the following page [▶ 694].

6.2.2.1.2.1 Download: Substation tutorial (slave, unbalanced mode)

A more complex substation application with different data points in both directions and a connection to the central station. The sample application uses the TwinCAT IEC 60870-5-101 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984615819.zip

In the sample project, the following station parameters are set for the substation:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220.**

- Link address size: **2 octets.**
- Cause of transfer size: **2 octets (including originator address).**
- Originator address: **1.**
- Common ASDU address: **7.**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Link mode**: Unbalanced.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200), M_DP_TA_1 (IOA := 201), M_DP_TB_1 (IOA := 202).
- Regulating step value: M_ST_NA_1 (IOA := 300), M_ST_TA_1 (IOA := 301), M_ST_TB_1 (IOA := 302).
- Bitstring value: M_BO_NA_1 (IOA := 400), M_BO_TA_1 (IOA := 401), M_BO_TB_1 (IOA := 402).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500), M_ME_TA_1 (IOA := 501), M_ME_TD_1 (IOA := 502).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600), M_ME_TB_1 (IOA := 601), M_ME_TD_1 (IOA := 602).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700), M_ME_TC_1 (IOA := 701), M_ME_TF_1 (IOA := 702).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10), C_SC_NA_1 (IOA := 11), C_SC_TA_1 (IOA := 12).
- Double command: C_DC_NA_1 (IOA := 20), C_DC_NA_1 (IOA := 21), C_DC_TA_1 (IOA := 22).
- Bitstring command: C_BO_NA_1 (IOA := 40), C_BO_NA_1 (IOA := 41), C_BO_TA_1 (IOA := 42).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50), C_SE_NA_1 (IOA := 51), C_SE_TA_1 (IOA := 52).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60), C_SE_NB_1 (IOA := 61), C_SE_TB_1 (IOA := 62).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70), C_SE_NC_1 (IOA := 71), C_SE_TC_1 (IOA := 72).

Sample implementation of the following services:

- End of initialization (M_EI_NA_1) is sent once.
- Test command (C_TS_NA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, Mode A). Counters are stored locally in the substation every 15 seconds and transmitted as spontaneous data to the central station.
- Command execution of some commands (direct commands, see upper data point configuration).
- Simulation and transmission of spontaneous data every 10 seconds if the variable **bSendSpont** was set to TRUE (see upper data point configuration).

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbServer	FB_IEC870_5_101Slave (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
F_iecInitAOEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.
memory	ST_MemoryData (STRUCT)	This structure contains constantly updated process data – either received or to be sent.
F_iecSetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be set/reset.
F_iecGetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be queried.

6.2.2.1.2.2 Glossary

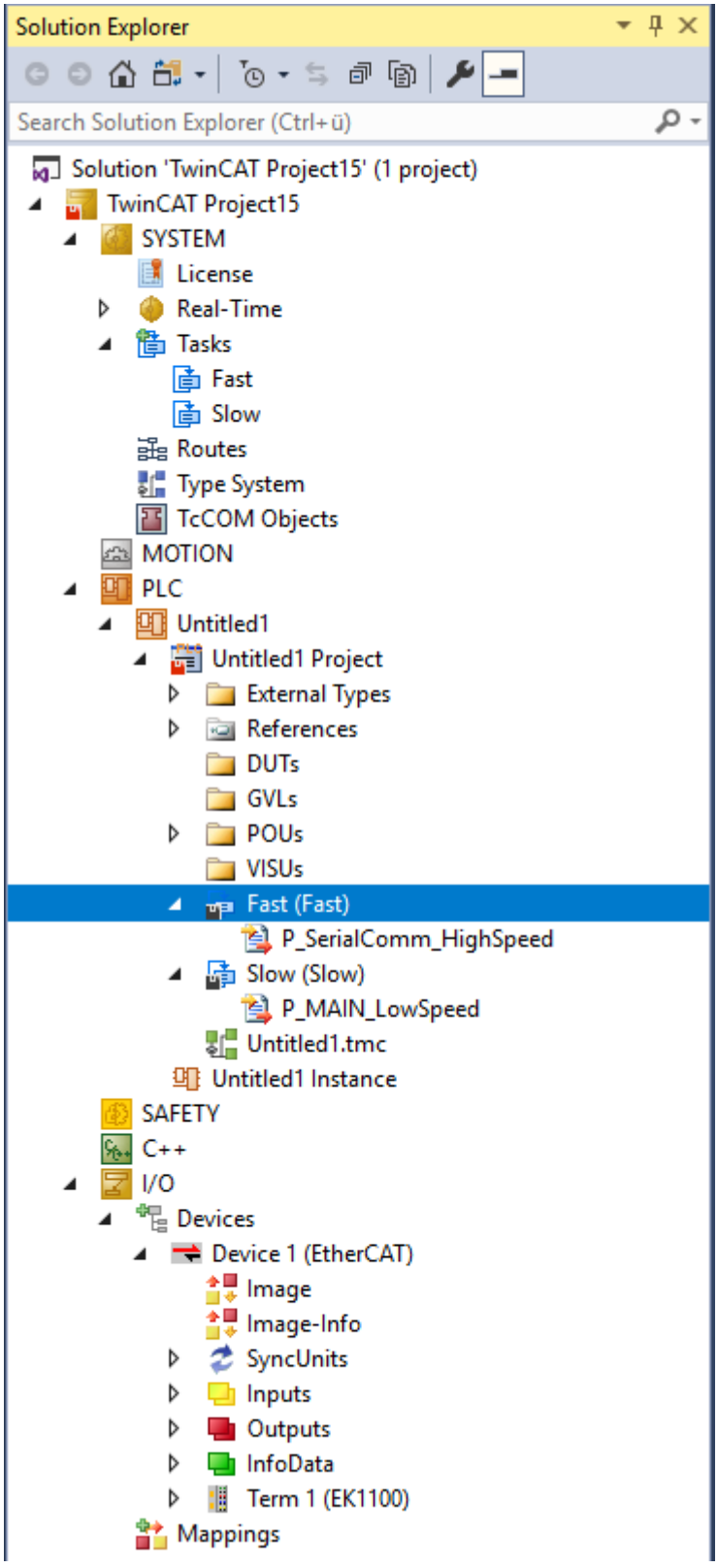
Term	Description
substation, slave, server	Synonyms for a subordinate station (which is monitored)
Central station, control station, master, client	Synonyms for a higher-level station (control station, monitors others stations)
control direction	Data transfer direction from the central station to the substation
monitoring direction	Data transfer direction from the substation to the central station
application objects	IEC information objects in the TwinCAT PLC application (single points, double points, measured values, short floating point values etc.)
APDU	Protocol data unit of the application layer (application protocol data unit)
APCI	Protocol control information of the application layer (application protocol control information)
ASDU	Service data unit of the application layer (application service data unit)
IOA, address of the information object	Address of the single point, double point, etc. (information object address)
primary station	The primary station sends commands (requests) to the secondary station and thus controls the data transfer of the secondary station.
secondary station	The secondary station responds to requests from the primary station.
Combined station	Combined stations can assume the role of primary station and secondary station (symmetric transfer, balanced mode).
Symmetric transfer (balanced mode)	Both stations can act as primary or secondary stations and initialize the data transfer.
Asymmetric transfer (unbalanced mode)	Data transfer is always initiated and controlled by the primary station. The central station always acts as the primary station and the substation as the secondary station.

6.2.2.1.2.3 Creating a PLC project and integrating PLC libraries

✓ Start TwinCAT XAE.

1. A wizard opens with **File > New > Project**. In this window you select **TwinCAT Project** and the **TwinCAT XAE Project** template.
2. Go to the PLC node in the Solution Explorer and select **Add New Item from** the context menu. Select a **Standard PLC Project** in the wizard.
3. In addition to the automatically created MAIN, add another programming block. Go to the **POU** folder and from there via the context menu to **Add POU**. Select the type **Program** and call it *P_MAIN_LowSpeed*.
4. Add another program block and name it *P_SerialComm_HighSpeed*.

5. Configure 2 tasks in the task configuration under System: a fast one (T#1ms) and a slow one (T#10ms). Assign the program block *P_MAIN_HighSpeed to* the fast task and the *P_MAIN_LowSpeed* to the slow task (see figure).



6. In the PLC Project node the library management can be found in the sub-node References. There you select the library **Tc2_IEC60870_5_10x.compiled-library** in the context menu via **Add Library** and confirm this.

⇨ Now all requirements are fulfilled to work through the tutorial step by step in configuration and programming.

6.2.2.1.2.4 The fast PLC task

Add the following PLC code to the declaration part:

```
PROGRAM P_SerialComm_HighSpeed
VAR
     fbSerialLineCtrl    : FB_IEC870_SerialLineCtrl := (
                Mode        := SERIALLINEMODE_PC_COM_PORT, (*SERIALLINEMODE_KL6_5B_STANDARD *)
                Baudrate    := 19200,
                NoDatabits      := 8,
                Parity      := PARITY_EVEN,
                Stopbits    := 1,
                Handshake       := HANDSHAKE_NONE,
                ContinousMode   := FALSE );

    serial_in           AT%IB4000       : PcComInData;
    serial_out      AT%QB4000       : PcComOutData;

    KL6_in      AT%IB4100       : KL6inData5B;
    KL6_out     AT%QB4100       : KL6outData5B;

    hSerial     : T_HSERIALCTRL;
END_VAR
```

and in the program code:

```
fbSerialLineCtrl(       pComIn      := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD,
ADR( serial_in ), ADR( KL6_in ) ),
                    pComOut     := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD, ADR(
 serial_out ), ADR( KL6_out ) ),
                    SizeComIn       := SEL( fbSerialLineCtrl.Mode = SERIALLINEMODE_KL6_5B_STANDARD,
SIZEOF( serial_in ), SIZEOF( KL6_in ) ),
                    hSerial     := hSerial );
```

A TwinCAT System Manager configuration suitable for this sample can be found on the sample overview page. The mode variable can be used to switch between two communication paths.

Communication via the standard PC COMx interfaces

- In this case, the mode parameter is set to the value: **SERIALLINEMODE_PC_COM_PORT**.
- In the TwinCAT System Manager, the *serial_in* and *serial_out* variables are linked with the corresponding IO variables of the serial interface.
- The interface must be configured in the TwinCAT System Manager (baud rate, parity, etc.). Other communication parameters at the FB_IEC870_SerialLineCtrl function block are irrelevant in this mode.

Communication via serial Beckhoff Bus Terminals KL6xxx

- In this case the mode parameter is set to the value: **SERIALLINEMODE_KL6_5B_STANDARD**.
- At the TwinCAT System Manager the *KL6_in* and *KL6_out* variables are bound to the corresponding IO variables of the serial terminal KL6xxx.
- The interface is configured within the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrlfunction block. Communication parameters like baud rate, parity etc. have to be set via this function block.

Communication via serial Beckhoff Bus Terminals EL6xxx

```
PROGRAM P_SerialComm_HighSpeed
VAR
    fbSerialLineCtrl : FB_IEC870_SerialLineCtrl := ( Mode := SERIALLINEMODE_EL6_22B );

    EL6_in AT%IB4100 : EL6inData22B;
    EL6_out AT%QB4100 : EL6outData22B;
    hSerial : T_HSERIALCTRL;
END_VAR
```

```
fbSerialLineCtrl( pComIn := ADR( EL6_in ),
        pComOut := ADR( EL6_out ),
        SizeComIn := SIZEOF( EL6_in ),
        hSerial := hSerial );
```

- In this case the mode parameter is set to the value: **SERIALLINEMODE_EL6_22B**.
- At the TwinCAT System Manager the *EL6_in* and *EL6_out* variables are bound to the corresponding IO variables of the serial terminal EL6xxx.

- The interface will and has to be configured within the TwinCAT System Manager (Baudrate, Parity etc.). Other communication parameters at the FB_IEC870_SerialLineCtrl function block are not relevant at this mode.

6.2.2.1.2.5 Defining and configuring an application object database of controlled station

Application objects = single points, double points, measured values, short floating point values, etc.

In this sample the commands were configured such that the process data for the commands are located in the same memory area as the information data in monitoring direction, although in a different byte and bit offset. If required the commands may have the same byte/bit offset as the information in monitoring direction.

Sample:

C_SC_NA_1 with IOA = 10 on the same byte/bit offset as M_SP_NA_1 with IOA = 100 (both byte offset = 100 and bit offset = 0). In this case a value change due to a command from the control station will result in a transmission of the M_SP_NA_1 with the object address 100 and cause of transfer <11> (returned by remote command).

As an example we will configure the following application objects as part of the introductory project:

Array element	ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset Byte	Offset Bit	Process data width in the TwinCAT PLC
0	M_SP_NA_1	100	General interrogation	0	Flags	100	0	1-bit
1	M_SP_NA_1	101	General interrogation	0	Flags	100	1	1-bit
2	M_SP_TB_1	102	General interrogation	0	Flags	100	2	1-bit
3	M_DP_NA_1	200	General interrogation	0	Flags	200	0	2 bits
4	M_DP_NA_1	201	General interrogation	0	Flags	200	2	2 bits
5	M_DP_TB_1	202	General interrogation	0	Flags	200	4	2 bits
6	M_ST_NA_1	300	General interrogation	0	Flags	300	0	1 byte
7	M_ST_NA_1	301	General interrogation	0	Flags	301	0	1 byte
8	M_ST_TB_1	302	General interrogation	0	Flags	302	0	1 byte
9	M_BO_NA_1	400	General interrogation	0	Flags	400	0	4 bytes
10	M_BO_NA_1	401	General interrogation	0	Flags	404	0	4 bytes
11	M_BO_TB_1	402	General interrogation	0	Flags	408	0	4 bytes
12	M_ME_NA_1	500	General interrogation	0	Flags	500	0	2 bytes
13	M_ME_NA_1	501	General interrogation	0	Flags	502	0	2 bytes
14	M_ME_TD_1	502	General interrogation	0	Flags	504	0	2 bytes
15	M_ME_NB_1	600	General interrogation	0	Flags	600	0	2 bytes
16	M_ME_NB_1	601	General interrogation	0	Flags	602	0	2 bytes

Array element	ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset Byte	Offset Bit	Process data width in the TwinCAT PLC
17	M_ME_TE_1	602	General interrogation	0	Flags	604	0	2 bytes
18	M_ME_NC_1	700	General interrogation	0	Flags	700	0	4 bytes
19	M_ME_NC_1	701	General interrogation	0	Flags	704	0	4 bytes
20	M_ME_TF_1	702	General interrogation	0	Flags	708	0	4 bytes
21	M_IT_NA_1	800	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	800	0	4 bytes
22	M_IT_NA_1	801	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	804	0	4 bytes
23	M_IT_TB_1	802	General counter interrogation and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	808	0	4 bytes
Commands								
24	C_SC_NA_1	10	-	0	Flags	2100	0	1-bit
25	C_SC_NA_1	11	-	0	Flags	2100	1	1-bit

Array element	ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset Byte	Offset Bit	Process data width in the TwinCAT PLC
26	C_SC_TA_1	12	-	0	Flags	2100	2	1-bit
27	C_DC_NA_1	20	-	0	Flags	2200	0	2-bit
28	C_DC_NA_1	21	-	0	Flags	2200	2	2-bit
29	C_DC_TA_1	22	-	0	Flags	2200	4	2-bit
30	C_RC_NA_1	30	-	0	Flags	2300	0	1 byte
31	C_RC_NA_1	31	-	0	Flags	2301	0	1 byte
32	C_RC_TA_1	32	-	0	Flags	2302	0	1 byte
33	C_BO_NA_1	40	-	0	Flags	2400	0	4 bytes
34	C_BO_NA_1	41	-	0	Flags	2404	0	4 bytes
35	C_BO_TA_1	42	-	0	Flags	2408	0	4 bytes
36	C_SE_NA_1	50	-	0	Flags	2500	0	2 bytes
37	C_SE_NA_1	51	-	0	Flags	2502	0	2 bytes
38	C_SE_TA_1	52	-	0	Flags	2504	0	2 bytes
39	C_SE_NB_1	60	-	0	Flags	2600	0	2 bytes
40	C_SE_NB_1	61	-	0	Flags	2602	0	2 bytes
41	C_SE_TB_1	62	-	0	Flags	2604	0	2 bytes
42	C_SE_NC_1	70	-	0	Flags	2700	0	4 bytes
43	C_SE_NC_1	71	-	0	Flags	2704	0	4 bytes
44	C_SE_TC_1	72	-	0	Flags	2708	0	4 bytes

Declaring a database variable

The application object database is an array variable of type ST_IEC870_5_101AODBEntry [▶ 584]

Each array element corresponds to an application object. The maximum number of application objects is freely selectable and is only limited by the available memory. During PLC programming you have to specify a constant maximum number. The maximum number of application objects cannot be changed at runtime.

In our sample 50 application objects are declared. This number is sufficient for most applications. Please note that many application objects require adequate memory and runtime resources.

Define the following variable in MAIN:

```
PROGRAM MAIN
VAR
     AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
END_VAR
```

Configuring application objects

The object type (M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 etc.), the object address and further object parameters are specified during configuration of the individual application objects.

The required application objects are configured during program runtime. Each application object (database array element) is configured by calling the F_iecInitAOEntry [▶ 537] function once. The array element to be configured is transferred to the function via VAR_IN_OUT. Configuration is usually carried out once during PLC program start-up via an Init routine. The F_iecInitAOEntry [▶ 537] function expects the following function parameters (from left to right):

```
FUNCTION F_iecInitAOEntry : UDINT
VAR_INPUT
    eType       : E_IEC870_5_101TcTypeID;
    objAddr     : DWORD := 0;
    group       : DWORD := 0;
    multiplier  : BYTE := 0;
    ioMapType   : E_IEC870_5_;
    byteOffs    : UDINT := 0;
    bitOffs     : UDINT := 0;
END_VAR
VAR_IN_OUT
    dbEntry     : ST_IEC870_5_101AODBEntry;
END_VAR
```

eType: application object type (ASDU identifier, e.g.: M_SP_NA_1 for single point or M_DP_NA_1 for double point). Please note that only the ASDU types listed in the compatibility list can be used. Invalid types are ignored.

objAddr: object address, e.g. 100. Each application object should be configured with a unique address.

group: group configuration parameters. The available group parameters are defined as constants and can be combined with an OR operator. E.g.: IEC870_GRP_INROGEN OR IEC870_GRP_PERCYC.

A description of all group configuration parameters [▶ 655] can be found here

mutiplier: base time multiplier for cyclic/periodic data transfer (0=deactivated). The base time is configured via the system parameters. If the base time was set to T#10 s, and the multiplier to 2, for example, the periodic/cyclic data of the application object are sent every 20 seconds.

ioMapType: This parameter defines from or in which process data area of the TwinCAT PLC the IEC process data are to be mapped at runtime (inputs, outputs, memory, data).

byteOffs: process data area byte offset;

bitOffs: process data area bit offset;

dbEntry: application object to be configured (a database variable array element that is transferred to the function via VAR_IN_OUT).

In order to configure the application objects during program start-up, add the following PLC code in MAIN:

```
PROGRAM MAIN
VAR
    AODB            : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
     init           : BOOL := TRUE;
    initError       : UDINT;
END_VAR
```

```
IF init THEN
    init := FALSE;

  (* Monitored Single Points *)
 initError := F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 0, AODB
[0] );
    initError := F_iecInitAOEntry( M_SP_NA_1, 101, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 1, A
ODB[1] );
    initError := F_iecInitAOEntry( M_SP_TB_1, 102, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 2, A
ODB[2]);
```

```
  (* Double Points*)
 initError := F_iecInitAOEntry( M_DP_NA_1, 200, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 0, AODB
[3] );
    initError := F_iecInitAOEntry( M_DP_NA_1, 201, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 2, A
ODB[4] );
    initError := F_iecInitAOEntry( M_DP_TB_1, 202, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 4, A
ODB[5] );
  (* Regulating step value *)
 initError := F_iecInitAOEntry( M_ST_NA_1, 300, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 300, 0, AODB
[6] );
    initError := F_iecInitAOEntry( M_ST_NA_1, 301, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 301, 0, A
ODB[7] );
    initError := F_iecInitAOEntry( M_ST_TB_1, 302, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 302, 0, A
ODB[8] );
  (* 32 bit string*)
 initError := F_iecInitAOEntry( M_BO_NA_1, 400, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 400, 0, AODB
[9] );
    initError := F_iecInitAOEntry( M_BO_NA_1, 401, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 404, 0, A
ODB[10] );
    initError := F_iecInitAOEntry( M_BO_TB_1, 402, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 408, 0, A
ODB[11] );
  (* Measured value, normalized value *)
 initError := F_iecInitAOEntry( M_ME_NA_1, 500, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 500, 0, AODB
[12] );
    initError := F_iecInitAOEntry( M_ME_NA_1, 501, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 502, 0, A
ODB[13] );
    initError := F_iecInitAOEntry( M_ME_TD_1, 502, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 504, 0, A
ODB[14] );
  (* Mesured value, scaled value *)
 initError := F_iecInitAOEntry( M_ME_NB_1, 600, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 600, 0, AODB
[15] );
    initError := F_iecInitAOEntry( M_ME_NB_1, 601, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 602, 0, A
ODB[16] );
    initError := F_iecInitAOEntry( M_ME_TE_1, 602, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 604, 0, A
ODB[17] );
  (* Measured value , short floating point value *)
 initError := F_iecInitAOEntry( M_ME_NC_1, 700, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 700, 0, AODB
[18] );
    initError := F_iecInitAOEntry( M_ME_NC_1, 701, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 704, 0, A
ODB[19] );
    initError := F_iecInitAOEntry( M_ME_TF_1, 702, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 708, 0, A
ODB[20] );
  (* Integrated totals *)
 initError := F_iecInitAOEntry( M_IT_NA_1, 800, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0, MAP_
AREA_MEMORY, 800, 0, AODB[21] );
    initError := F_iecInitAOEntry( M_IT_NA_1, 801, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0, M
AP_AREA_MEMORY, 804, 0, AODB[22] );
    initError := F_iecInitAOEntry( M_IT_TB_1, 802, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0, M
AP_AREA_MEMORY, 808, 0, AODB[23] );

  (* Single commands *)
 initError := F_iecInitAOEntry( C_SC_NA_1, 10, 0, 0, MAP_AREA_MEMORY, 2100, 0, AODB[24] );
    initError := F_iecInitAOEntry( C_SC_NA_1, 11, 0, 0, MAP_AREA_MEMORY, 2100, 1, AODB[25] );
    initError := F_iecInitAOEntry( C_SC_TA_1, 12, 0, 0, MAP_AREA_MEMORY, 2100, 2, AODB[26] );
  (* Double commands *)
 initError := F_iecInitAOEntry( C_DC_NA_1, 20, 0, 0, MAP_AREA_MEMORY, 2200, 0, AODB[27] );
    initError := F_iecInitAOEntry( C_DC_NA_1, 21, 0, 0, MAP_AREA_MEMORY, 2200, 2, AODB[28] );
    initError := F_iecInitAOEntry( C_DC_TA_1, 22, 0, 0, MAP_AREA_MEMORY, 2200, 4, AODB[29] );
  (* Regulating step commands *)
 initError := F_iecInitAOEntry( C_RC_NA_1, 30, 0, 0, MAP_AREA_MEMORY, 2300, 0, AODB[30] );
    initError := F_iecInitAOEntry( C_RC_NA_1, 31, 0, 0, MAP_AREA_MEMORY, 2301, 0, AODB[31] );
    initError := F_iecInitAOEntry( C_RC_TA_1, 32, 0, 0, MAP_AREA_MEMORY, 2302, 0, AODB[32] );
  (* 32 bit string commands *)
 initError := F_iecInitAOEntry( C_BO_NA_1, 40, 0, 0, MAP_AREA_MEMORY, 2400, 0, AODB[33] );
    initError := F_iecInitAOEntry( C_BO_NA_1, 41, 0, 0, MAP_AREA_MEMORY, 2404, 0, AODB[34] );
    initError := F_iecInitAOEntry( C_BO_TA_1, 42, 0, 0, MAP_AREA_MEMORY, 2408, 0, AODB[35] );
  (* Set point, normalized values*)
 initError := F_iecInitAOEntry( C_SE_NA_1, 50, 0, 0, MAP_AREA_MEMORY, 2500, 0, AODB[36] );
    initError := F_iecInitAOEntry( C_SE_NA_1, 51, 0, 0, MAP_AREA_MEMORY, 2502, 0, AODB[37] );
    initError := F_iecInitAOEntry( C_SE_TA_1, 52, 0, 0, MAP_AREA_MEMORY, 2504, 0, AODB[38] );
  (* Set point, scaled valuess *)
 initError := F_iecInitAOEntry( C_SE_NB_1, 60, 0, 0, MAP_AREA_MEMORY, 2600, 0, AODB[39] );
    initError := F_iecInitAOEntry( C_SE_NB_1, 61, 0, 0, MAP_AREA_MEMORY, 2602, 0, AODB[40] );
    initError := F_iecInitAOEntry( C_SE_TB_1, 62, 0, 0, MAP_AREA_MEMORY, 2604, 0, AODB[41] );
  (* Set point, short floating point values *)
 initError := F_iecInitAOEntry( C_SE_NC_1, 70, 0, 0, MAP_AREA_MEMORY, 2700, 0, AODB[42] );
    initError := F_iecInitAOEntry( C_SE_NC_1, 71, 0, 0, MAP_AREA_MEMORY, 2704, 0, AODB[43] );
    initError := F_iecInitAOEntry( C_SE_TC_1, 72, 0, 0, MAP_AREA_MEMORY, 2708, 0, AODB[44] );

END_IF
```

6.2.2.1.2.6 Mapping of PLC and IEC process data

The TwinCAT PLC process data are cyclically mapped (copied) into the IEC process data (application objects) and vice versa at program runtime. Up to four process data areas (IO inputs, IO outputs, flag range, data area) can be declared in the PLC program as buffer variables for the mapping of the IEC<->PLC process data. The byte size of the buffers is freely selectable and may be different for each area. Unused ranges need not necessarily be declared.

In our introductory example we declare 4 PLC process data areas with 3000 bytes each:

```
PROGRAM MAIN
VAR
    AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;

    init : BOOL := TRUE;
    initError : UDINT;

     inputs AT%IB0 : ARRAY[0..2999] OF BYTE;
    outputs AT%QB0 : ARRAY[0..2999] OF BYTE;
    memory AT%MB0 : ARRAY[0..2999] OF BYTE;
    data : ARRAY[0..2999] OF BYTE;

END_VAR
```

How the process data are to be mapped at runtime is specified during configuration of the application objects via the F_iecInitAOEntry function.
See also in: Definition and configuration of application objects [▶ 537].

The buffer variables were now declared as byte arrays. In order to improve access to the required data we define the individual variables a second time and allocate them to the corresponding byte/bit offset addresses. In case of a change in the byte array, the corresponding individual variable will be changed and inverted at the same time. However, this is not absolutely necessary. The bytes/bits of the byte array buffer variables can be accessed directly.

```
VAR_GLOBAL(* Memory offset 0..99 unused *)
  (* Single points *)
 msgSingle_0     AT%MX100.0 : BOOL;
    msgSingle_1     AT%MX100.1 : BOOL;
    msgSingle_2     AT%MX100.2 : BOOL;

  (* Double points *)
    (*      Bit 0..1 = first double point,
        Bit 2..3 = second double point,
        Bit 4..5 = third double point,
        Bit 6..7 = fourth double point *)
    msgDouble_0     AT%MB200    : BYTE;

  (* Regulating step values *)
    msgStep_0           AT%MB300    : BYTE;
    msgStep_1           AT%MB301    : BYTE;
    msgStep_2           AT%MB302    : BYTE;

  (* 32 bit strings *)
 msgBitStr_0     AT%MD400    : DWORD := 2#10001000_10001000_10001000_10001000;
    msgBitStr_1     AT%MD404    : DWORD := 2#10001000_10001000_10001000_10001000;
    msgBitStr_2     AT%MD408    : DWORD := 2#10001000_10001000_10001000_10001000;

  (* Measured values, normalized values *)
 msgNormalized_0     AT%MW500    : WORD;
    msgNormalized_1     AT%MW502    : WORD;
    msgNormalized_2     AT%MW504    : WORD;

  (* Measured values, scaled values *)
 msgScaled_0     AT%MW600    : INT;
    msgScaled_1     AT%MW602    : INT;
    msgScaled_2     AT%MW604    : INT;

  (* Measured values, short floating point values *)
 msgFloating_0   AT%MD700    : REAL;
    msgFloating_1   AT%MD704    : REAL;
    msgFloating_2   AT%MD708    : REAL;

  (* Integrated totals *)
 msgTotal_0      AT%MD800    : UDINT;
    msgTotal_1      AT%MD804    : UDINT;
    msgTotal_2      AT%MD808    : UDINT;
```

```
  (*#################################################################*)
    (* Single commands *)
 cmdSingle_0     AT%MX2100.0 : BOOL;
    cmdSingle_1     AT%MX2100.1 : BOOL;
    cmdSingle_2     AT%MX2100.2 : BOOL;

  (* Double commands *)
    (*      Bit 0..1 = first double command,
        Bit 2..3 = second double command,
        Bit 4..5 = third double command,
        Bit 6..7 = fourth double command *)
 cmdDouble_0     AT%MB2200       : BYTE;

  (* Regulating step commands *)
 cmdStep_0           AT%MB2300 : BYTE;
    cmdStep_1           AT%MB2301 : BYTE;
    cmdStep_2           AT%MB2302 : BYTE;

  (* 32 bit string commands *)
 cmdBitStr_0     AT%MD2400       : DWORD;
    cmdBitStr_1     AT%MD2404       : DWORD;
    cmdBitStr_2     AT%MD2408       : DWORD;

  (* Set point, normalized values *)
 cmdNormalized_0     AT%MW2500       : WORD;
    cmdNormalized_1     AT%MW2502       : WORD;
    cmdNormalized_2     AT%MW2504       : WORD;

  (* Set point, scaled values *)
 cmdScaled_0     AT%MW2600       : INT;
    cmdScaled_1     AT%MW2602       : INT;
    cmdScaled_2     AT%MW2604       : INT;

  (* Set point, short floating point values *)
 cmdFloating_0   AT%MD2700       : REAL;
    cmdFloating_1   AT%MD2704       : REAL;
    cmdFloating_2   AT%MD2708       : REAL;
END_VAR
```

Mapping of the IEC<->PLC process data in the substation

Process data in monitoring direction (slave->master information)

Example 1

Single point information (M_SP_NA_1) with the IOA = 100, PLC memory area, byte offset = 100, bit offset = 0.

msgSingle_0 == memory[100].0 -> Controlled station FB -> ... -> Controlling station

Example 2

Measured value, short floating point value (M_ME_NC_1) with the IOA = 700, PLC memory area, byte offset = 700, bit offset = 0 (irrelevant).

msgFloating_0 == memory[700..703] -> Controlled station FB -> ... -> Controlling station

Process data in control direction (master->slave commands)

Example 1

Single command state (C_SC_NA_1) with the IOA = 10, PLC memory area, byte offset = 2100, bit offset = 0.

Controlling station -> ... -> Controlled station FB -> memory[2100].0 == cmdSingle_0

Example 2

Set point, short floating point value (C_SE_NC_1) with the IOA = 70, PLC memory area, byte offset = 2700, bit offset = 0 (irrelevant).

Controlling station -> ... -> Controlled station FB -> memory[2700..2703] == cmdFloating_0

6.2.2.1.2.7 Declaring and calling an instance of the IEC60870-5-101 substation

You can download the complete PLC sources here [▶ 694].

The complete functionality of a substation is encapsulated in the FB_IEC870_5_101Slave function block. An instance can be used to establish a connection to the master. The *hSerial* connection handle of the fast task must be passed to the substation as VAR_IN_OUT variable.

Insert the following PLC code in the declaration part of *P_MAIN_LowSpeed*:

```
PROGRAM P_MAIN_LowSpeed
VAR
    AODB        : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;

    init        : BOOL := TRUE;
    error       : UDINT;

    inputs AT%IB0   : ARRAY[0..2999] OF BYTE;
    outputs AT%QB0  : ARRAY[0..2999] OF BYTE;
    memory AT%MB0   : ARRAY[0..2999] OF BYTE;
    data        : ARRAY[0..2999] OF BYTE;


     bEnable         : BOOL := TRUE;
    server      : FB_IEC870_5_101Slave;
END_VAR
```

and the instance is called in the program part:

```
IF init THEN
    init := FALSE;
...



ELSE
...
 server(     pInputs := ADR( inputs ),     cbInputs := SIZEOF( inputs ),     pOutputs := ADR( output
s ),     cbOutputs := SIZEOF( outputs ),     pMemory := ADR( memory ),     cbMemory := SIZEOF( memor
y ),     pData := ADR( data ),     cbData := SIZEOF( data ),     pAOEntries := ADR( AODB ),     cbAO
Entries := SIZEOF( AODB ),     hSerial := P_SerialComm_HighSpeed.hSerial, (* serial link interface c
onnection handle from fast task *)     bEnable := bEnable );
...
END_IF
```

6.2.2.1.2.8 IEC60870-5-101 protocol parameters

The behavior of the substation can be adapted to the requirements of the master via the IEC60870-5-101 protocol parameters. Most parameters have preallocated default values that do not have to be changed.

In our sample we set the link address and the byte length of the link address:

```
IF init THEN
    init := FALSE;
...

    server.protPara.linkAddr   := 220;(* slave link address *)
 server.protPara.eLinkAddrSize := eIEC870_LinkAddr_TwoOctets; (* link address octet size *)

...

ELSE
    server( pInputs  := ADR( inputs ),
        cbInputs     := SIZEOF( inputs ),
        pOutputs     := ADR( outputs ),
...

END_IF
```

The documentation for all transfer protocol parameters can be found here: ST_IEC870_5_101PotocolParams [▶ 640]

6.2.2.1.2.9 System parameters

The common ASDU address and the user functions of the substation are configured via the system parameters.

In our introduction we configure the following system parameters:

- The common ASDU address is set to 7. (*asduAddr*)
- During initialization the system time of the substation is synchronized with the system time of the local TwinCAT PC (*bUsePCTime*).
- Synchronization of the substation system time via the clock synchronization command is activated (*bSyncTime*).
- During the synchronization of the system time in the substation, the system time of the TwinCAT PC should not be synchronized (*bSyncPCTime*).
- Sending of M_EI_NA_1 (End of init) to the central station is activated (*bEndOfInit*).
- Sending of the periodic/cyclic data is deactivated (*bPerCyclic*). The base time for sending of these data is set to 5 s.
- Background scan is deactivated (*bBackScan*). The background scan cycle time is set to 30 s.
- Local freeze and resetting of the counter readings is activated (*bPerFRZ*), and the cycle time for freeze and resetting is set to 15 s.
- Logging of debugging messages in the application log is activated (*dbgMode*). Changes in the device status are logged.

Add the following PLC code to your PLC project:

```
IF init THEN
    init := FALSE;
...

    server.sysPara.asduAddr := 7;
    server.sysPara.bUsePCTime := TRUE;
    server.sysPara.bSyncTime := TRUE;
    server.sysPara.bSyncPCTime := FALSE;
    server.sysPara.bEndOfInit := TRUE;
    server.sysPara.bPerCyclic := FALSE;
    server.sysPara.tPerCyclicBase := T#5s;
    server.sysPara.bBackScan := FALSE;
    server.sysPara.tBackScanCycle := T#30s;
    server.sysPara.bPerFRZ := TRUE;
    server.sysPara.tPerFRZCycle := T#15s;
    server.sysPara.dbgMode :=  (*IEC870_DEBUGMODE_ASDU OR*) IEC870_DEBUGMODE_DEVSTATE;

...
ELSE
    server( pInputs := ADR( inputs ),
        cbInputs := SIZEOF( inputs ),
        pOutputs := ADR( outputs ),
...

END_IF
```

The documentation for all system parameters can be found here: ST_IEC870_5_101SystemParams [▶ 588].

6.2.2.1.2.10 Station interrogation

The station interrogation command is initiated by the central station. The ID field of the command also contains the group (1 to 16 or general). The substation transfers the application objects associated with this group to the central station with cause of transfer <*20*> to <*36*>. Application objects with time tags are transferred without time tags.

Configuration of the system parameters

- No special system parameters need to be set;

Configuration of the application objects

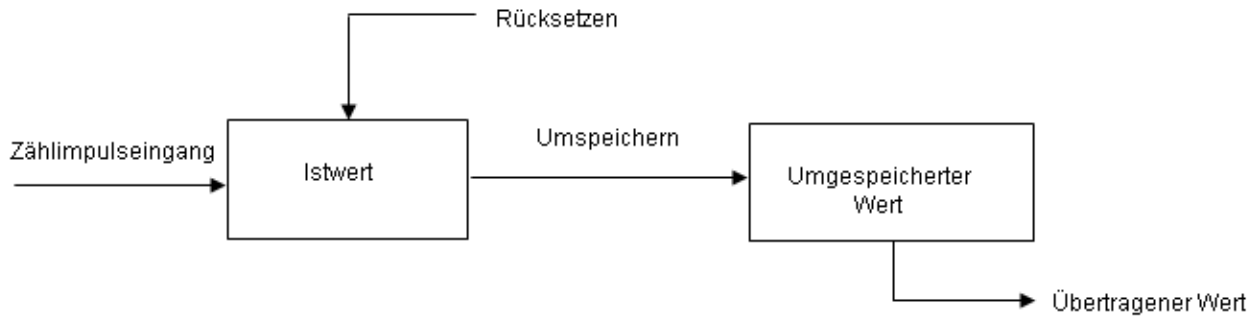
- The data point must be assigned to one or more groups. The group parameter must be set. An overview of all available groups can be found here: Group configuration flags [▶ 655].

Sample configuration for a data point assigned to the group: 1 and the group: General.

```
F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN OR IEC870_GRP_INRO1, 0, MAP_AREA_MEMORY, 100, 0
, AODB[0] );
```

6.2.2.1.2.11 Transfer of integrated totals (counter interrogation)

General model for the transmission of integrated totals:



The actual values are added by counters. Via a freeze command that is either received by the central station or generated locally (in the substation), the actual values can be frozen (copied) periodically into frozen values. After freezing, the recorded value is either reset to null (logging of incremental values), or the counter continues adding up (logging of counter values).

Application objects with counts are assigned to groups. The groups are frozen individually, reset, or transferred. The central station sends counter interrogation commands to the substation. The task to be carried out (FRZ) and the group (RQT) are specified in an ID field of the command (QCC).

The allocation of the application objects to the individual groups (1 to 4 or general) is specified via the group flag parameter during configuration. There are four operation modes for recording counter readings and incremental values. Each mode includes notes about the configuration of the system parameters or the application objects.

Mode A: local freeze with spontaneous transfer

The substation internally initiates freeze or freeze with reset. The frozen counts are transferred spontaneously, once the freeze or freeze with reset function was executed. In this mode the central station does not issue any counter interrogation commands.

Configuration of the system parameters:

```
bPerFRZ  := TRUE
tPerFRZCycle := T#60s
```

The first parameter activates local freeze or freeze with reset. The second parameter specifies the cycle time with which freeze or freeze with reset is carried out (e.g. every 60 seconds).

Configuration of the application objects:

- IEC870_GRP_SPONTOFF group parameter must not be set. This would prevent the spontaneous transmission of the counts.
- The counter value is frozen if IEC870_GRP_LOCFREEZE group parameter has been set.
- The counter value is reset if IEC870_GRP_LOCRESET group parameter has been set.
- Local freeze or freeze with reset is carried out simultaneously for all groups (1 to 4 or general).

Operation mode B: Local freeze with counter interrogation

The substation internally initiates freeze or freeze with reset. The central station queries the frozen counts via counter interrogation commands. In this case the central station must not use freeze or freeze with reset in the command ID field (FRZ=0). The counts are queried generally or in groups 1 to 4.

Configuration of the system parameters:

```
bPerFRZ  := TRUE
tPerFRZCycle := T#60s
```

The first parameter activates local freeze or (and) reset. The second parameter specifies the cycle time with which freeze or freeze with reset is carried out (e.g. every 60 seconds).

Configuration of the application objects:

- The IEC870_GRP_SPONTOFF group parameter must be set. The counts are not to be transferred spontaneously to the central station.
- The counter value is frozen if IEC870_GRP_LOCFREEZE group parameter has been set.
- The counter value is reset if IEC870_GRP_LOCRESET group parameter has been set.
- Local freeze or freeze with reset is carried out simultaneously for all groups (1 to 4 or general).

Operating mode C: The central station initiates freeze, freeze with reset, or reset

The central station periodically issues a counter interrogation command to the substation for controlling the freeze or (and) reset process. This command does not result in a count transfer. The central station sends a subsequent counter interrogation command for collecting the frozen counts. This is similar to operating mode B.

Configuration of the system parameters:

```
bPerFRZ  := FALSE
tPerFRZCycle := T#60s
```

Local freeze or (and) reset must be deactivated. The second parameter is ignored.

Configuration of the application objects:

- IEC870_GRP_SPONTOFF must be set. The counts are not to be transferred spontaneously to the central station.
- IEC870_GRP_LOCFREEZE and IEC870_GRP_LOCRESET group parameters must not be set. The central station initiates freeze or (and) reset.
- The counts can be assigned to individual groups (1 to 4 or general) and queried (IEC870_GRP_REQCOGEN, IEC870_GRP_REQCO1, IEC870_GRP_REQCO2, IEC870_GRP_REQCO3, IEC870_GRP_REQCO4).

Operating mode D: The central station initiates freeze and (or) reset, and the frozen values are transferred spontaneously

This mode is a combination of the count command from the central station as in mode C and spontaneous transfer of the counts as in mode A.

Configuration of the system parameters:

```
bPerFRZ  := FALSE
tPerFRZCycle := T#60s
```

Local freeze or (and) reset must be deactivated. The second parameter is ignored.

Configuration of the application objects:

- IEC870_GRP_SPONTOFF group parameter must not be set. This would prevent the spontaneous transmission of the counts.
- IEC870_GRP_LOCFREEZE and IEC870_GRP_LOCRESET group parameters must not be set. The central station initiates freeze or (and) reset.
- The counts can be assigned to individual groups (1 to 4 or general) and queried (IEC870_GRP_REQCOGEN, IEC870_GRP_REQCO1, IEC870_GRP_REQCO2, IEC870_GRP_REQCO3, IEC870_GRP_REQCO4).

6.2.2.1.2.12 Clock (time) synchronisation

The way in which the system time of the substation is to be synchronized can be configured via the system parameters [▶ 588].

- During initialization the system time of the substation can be synchronized with the system time of the local TwinCAT PC;
- When a time synchronization command is received from the central station, the system time of the substation can also be synchronized;
- The system time of the local TwinCAT PC can also be synchronized when a time synchronization command is received.

6.2.2.1.2.13 Background scan

The background scan is used for refreshing the process information sent from the substation to the central station as an additional safety contribution to the station interrogation and for spontaneous transfers. Application objects with the same type IDs as for the station interrogation may be transferred continuously with low priority, and with <2> background scan as the cause of transfer. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transfer). The background scan is initiated by the substation and is independent of the station interrogation commands.

Configuration of the system parameters

The transfer cycle is specified via system parameters [▶ 588] in the substation.

```
bBackScan := TRUE;
tBackScanCycle := T#30s;
```

Configuration of the application objects

Application objects whose process data is to be transferred as a background scan must be configured with the Group flag: IEC870_GRP_BACKGROUND.

Sample:

```
F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN OR IEC870_GRP_BACKGROUND, 0, MAP_AREA_MEMORY, 1
00, 0, AODB[0] );
```

6.2.2.1.2.14 Cyclic data transmission

Cyclic data transfer is initiated in a similar way as the background scan from the substation. It is independent of other commands from the central station. Cyclic data transfer continuously refreshes the process data of the central station. The process data are usually measured values that are recorded at regular intervals. Cyclic data transfer is often used for monitoring non-time-critical or relatively slowly changing process data (e.g. temperature sensor data). Cyclic/periodic data are transferred to the central station with cause of transfer *<1> periodic/cyclic*. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transfer). Cyclic data transfer can be configured via the system parameters and the configuration parameters of the application objects.

Configuration of the system parameters:

```
bPerCyclic       : BOOL := TRUE;
tPerCyclicBase   : TIME := T#60s;
```

The first parameter activates cyclic transfer. The second parameter is the base time for the cyclic/periodic data transfer (in this case 60 seconds).

Configuration of the application objects:

- IEC870_GRP_PERCYC group parameter has to be set;
- The multiplier parameter *(multiplier)* of the F_iecInitAOEntry function has to be set to a value <> null. Sample: With a multiplier = 2 and a base time of 60 seconds the process data of the application object are sent to the central station every 120 seconds;

Sample configuration for a measured value that is to be transmitted cyclically every 120 seconds to the central station (measured value, normalized value without time tag, M_ME_NA_1).

```
F_iecInitAOEntry( M_ME_NA_1, 222, IEC870_GRP_INROGEN OR IEC870_GRP_PERCYC, 2, MAP_AREA_MEMORY, 6, 0,
 AODB[2] );
```

6.2.2.1.2.15 Command transmission

Commands can be sent from the central station in control direction (to the substation).
A single command with type ID 45 (C_SC_NA_1) is used for controlling an application object that is transferred in monitoring direction as a single point (M_SP_NA_1, M_SP_TA_1 or M_SP_TB_1). A dual command (C_DC_NA_1) is used for controlling an application object that is transferred in monitoring direction as a double point (M_DP_NA1, M_DP_TA_1 or M_DP_TB_1), etc.

Configuration of the system parameters:

- No special system parameters need to be set;

Configuration of the application objects:

- The application objects must be configured as commands (data types in the control direction);
- The addresses of the information objects (IOAs) must correspond to the addresses in the control station;

Samples:

Single command with IOA = 10. The received value is copied into the flag range buffer, byte offset = 100, bit offset = 0.

```
F_iecInitAOEntry( C_SC_NA_1, 10, 0, 0, MAP_AREA_MEMORY, 100, 0, AODB[24] );
```

Double command with IOA = 20. The received value is copied into the flag range buffer, byte offset = 200, bit offset = 0..1.

```
F_iecInitAOEntry( C_DC_NA_1, 20, 0, 0, MAP_AREA_MEMORY, 200, 0, AODB[27] );
```

Set point, scaled value with IOA = 60. The received value is copied into the flag range buffer, byte offset = 600..601, bit offset = 0.

```
F_iecInitAOEntry( C_SE_NB_1, 60, 0, 0, MAP_AREA_MEMORY, 600, 0, AODB[39] )
```

6.2.2.1.2.16 Interrogation / Read command

The central station sends interrogation commands to the substation.
The interrogation command contains the address of the application object to be interrogated. The data of this application object are to be sent to the central station. The substation sends the data with cause of transfer *<5> interrogation or interrogated*. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transfer).

Configuration of the system parameters

- No special system parameters need to be set;

Configuration of the application objects

- No special parameters need to be set;

6.2.2.1.2.17 Double transmission

All application objects (information objects) that are transferred with cause of transfer *<3> spontaneous* may be transferred twice, with or without time tag. This mode is referred to as "double transmission".
Double transmission is currently not supported by the substation. A custom implementation is possible via the low-level interface. [▶ 112]

6.2.2.1.2.18 Quality Flags

The quality flags (quality descriptor) provide additional information for the central station on the quality of an application object. The quality flags can be set/reset independent of each other from the PLC application via the F_iecSetAOQuality [▶ 539] function. The F_iecGetAOQuality function can be used to query the state of the quality flags. Any change in the quality flags leads to a spontaneous transfer of the data to the central station.

The following quality flags are internally analyzed by the substation at runtime:

- IECQ_BL_ON (blocked). If the process data of the application object were blocked for the transfer, mapping of the PLC and IEC process data is not executed for this application;

The following quality flags are internally set/reset by the substation at runtime:

- IECQ_IV_ON (Invalid). The substation sets the invalid flag if mapping of the PLC and IEC process data could not be carried out (e.g. due to faulty configuration of the application object). This behavior can be deactivated by setting a Group parameter: IEC870_GRP_IV_OFF.

All other quality flags are sent to the central station without change.

6.2.2.1.2.19 Testing the communication

A simple simulation of the data points in the monitoring direction is activated by setting the *bChangeIO* variable to TRUE and deactivated by setting it to FALSE. If a connection is active, the values are transferred cyclically to the control station every 3 seconds.

```
PROGRAM MAIN
VAR
    ...

    bChangeIO : BOOL; (* TRUE => simulate/modify plc process data *)
    timer : TON;
    i : INT;

    ...
END_VAR
```

```
...


(*modify plc process data *)
timer( IN := bChangeIO, PT := T#3s );
IF timer.Q THEN
    timer( IN := FALSE );

    msgSingle_0 := NOT msgSingle_0;
    msgSingle_1 := NOT msgSingle_1;
    msgSingle_2 := NOT msgSingle_2;

    FOR i:= 0 TO 3 DO
        IF F_iecGetDPI(msgDouble_0, i) = eIEC870_DPI_ON THEN (* the value of double point allready O
N? *)
            msgDouble_0 := F_iecSetDPI(     msgDouble_0, i, eIEC870_DPI_OFF ); (* change ON => OFF *
)
        ELSE
            msgDouble_0 := F_iecSetDPI(     msgDouble_0, i, eIEC870_DPI_ON );(* change OFF => ON *)
        END_IF
    END_FOR

    F_iecIncVTI( msgStep_0 );
    F_iecDecVTI( msgStep_1 );

    msgBitStr_0 := ROL( msgBitStr_0, 1 );
    msgBitStr_1 := ROR( msgBitStr_1, 1 );

    msgNormalized_0 := msgNormalized_0 + 1;
    msgNormalized_1 := msgNormalized_1 + 2;

    msgScaled_0 := msgScaled_0 + 3;
    msgScaled_1 := msgScaled_1 - 3;

    msgFloating_0 := msgFloating_0 + 0.1;
    msgFloating_1 := msgFloating_1 + 1.5;

    msgTotal_0 := msgTotal_0 + 1;
    msgTotal_1 := msgTotal_1 + 2;
END_IF


...
```

6.2.2.1.2.20 Protocol and data transmission errors

The station error messages are stored in a FIFO memory. Up to 10 error messages can be buffered. In the case of fatal communication errors (e.g. errors of the link layer, the checksum of the frame is wrong) the connection is interrupted and must be re-established. Errors in the application layer (e.g. the ASDU transmit buffer has overflowed due to too many frames) are only logged and do not lead to interruption of the connection. Even with these errors it is still possible to interrupt the connection from the application. Apart from the error code, the error source is also stored in the error message. This makes it easier to locate the error.

Sample

The accumulated error messages of an IEC 60870-5-101 substation can be read out using the following call:

```
PROGRAM MAIN
VAR
...

    server : FB_IEC870_5_101Slave;

...
END_VAR
```

```
....


REPEAT
    server.system.device.errors.RemoveError();
    IF server.system.device.errors.bOk  THEN
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG,
            'IEC60870-5-101 slave error: 0x%s',
            DWORD_TO_HEXSTR( server.system.device.errors.getError.nErrId, 8, FALSE) );
    END_IF
UNTIL NOT server.system.device.errors.bOk
END_REPEAT

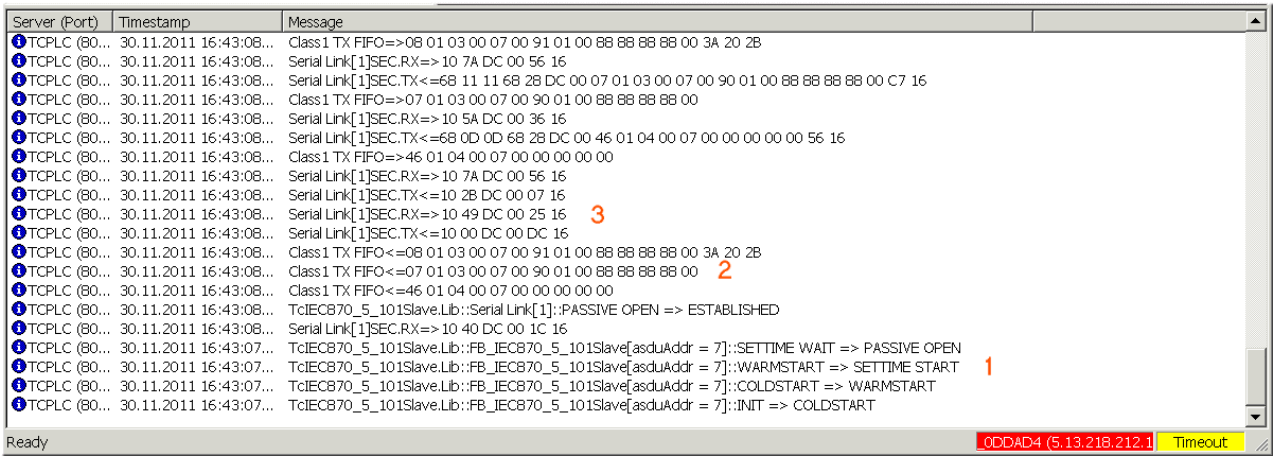
...
```

6.2.2.1.2.21 Troubleshooting and debugging

Debugging messages written to the application log facilitate troubleshooting of the system. Currently, three levels of debug messages can be activated in an IEC application. These messages can be activated via the dbgMode system parameter of the substation (ST_IEC870_5_101SystemParams [▶ 588]).

1. Station status messages (dbgMode: IEC870_DEBUGMODE_DEVSTATE);
2. Hexadecimal output of the ASDU's (without link layer control header, dbgMode: IEC870_DEBUGMODE_ASDU). 32 ASDU data bytes per row are output as hexadecimal numbers. Longer ASDUs are spread across several rows;
3. Hexadecimal output of the APDU's (serial port telegrams, dbgMode: IEC870_DEBUGMODE_LINKLAYER). 32 APDU data bytes per row are output as hexadecimal numbers. Like in 2., longer APDUs are spread across several rows;

Optionally, link layer errors can also be output (dbgMode: IEC870_DEBUGMODE_LINKERROR). In order to view the activated debug messages start the TwinCAT System Manager and activate log view. A debugging output is shown below. The three different message types are identified with the respective numbers.



Further diagnostic tools:

- Portmon for Windows (v3.02, Windows Sysinternals)
- Diverse protocol test suite products

6.2.2.1.2.22 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 716]
- Configuration of the serial KL6xxx Bus Terminals [▶ 717]
- Configuration of the serial EL6xxx Bus Terminals [▶ 717]

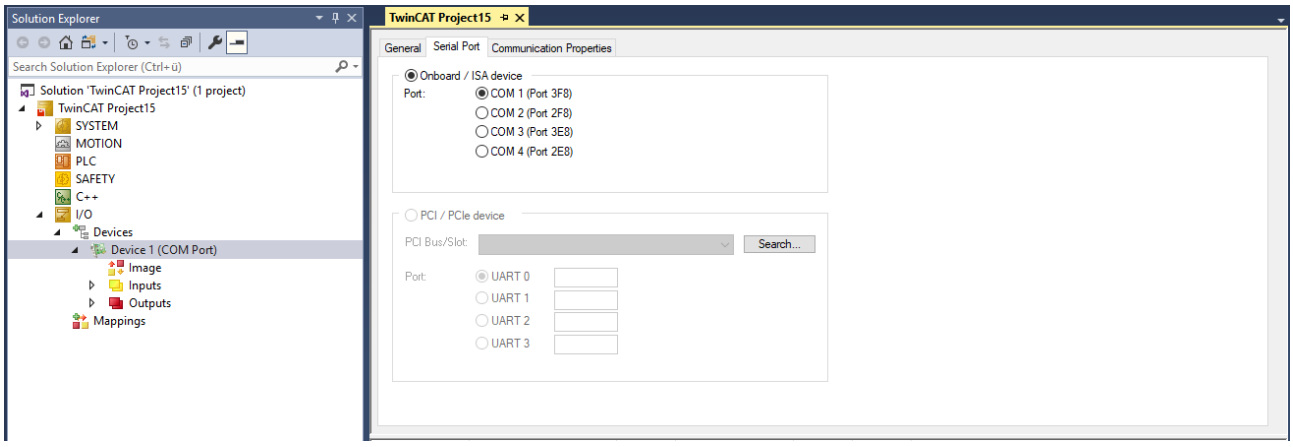
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

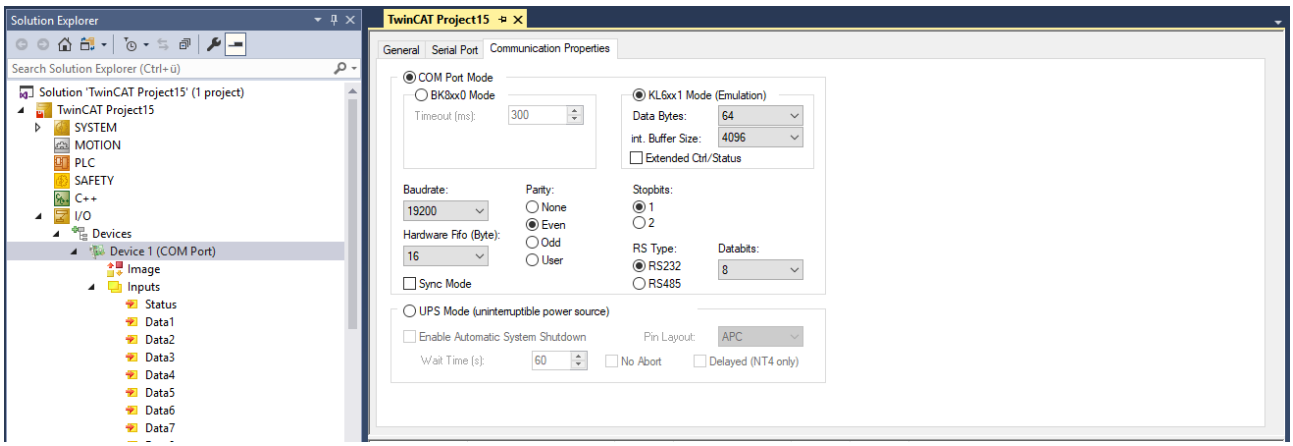
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

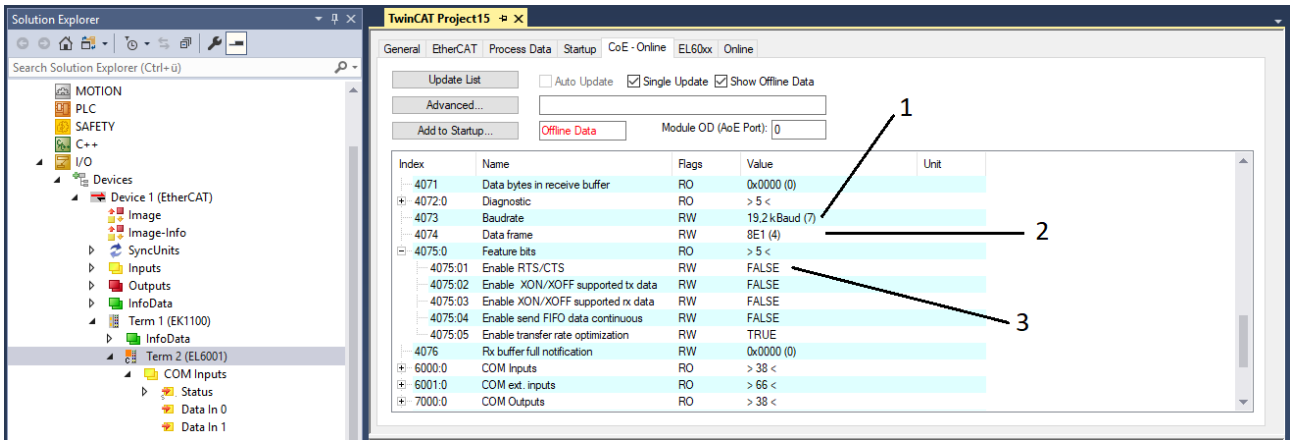
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

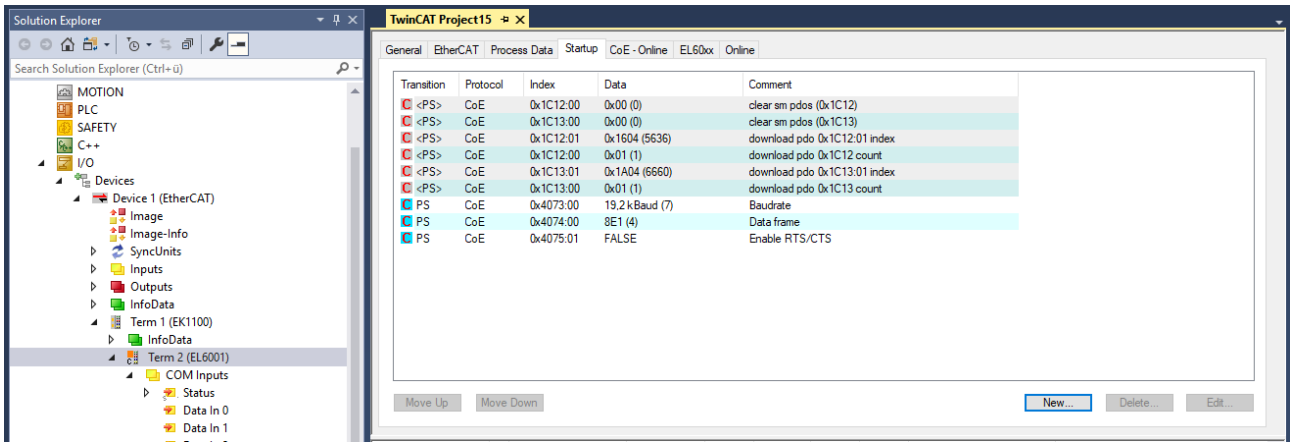
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.2.2.2 Low-Level Interface

6.2.2.2.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-101 Substation low-level sample [▶ 718].

A PLC application that is to communicate with a central station via the "low-level" interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_101TProtocol [▶ 510];
- An instance of the TX/RX data buffer: ST_IEC870_5_101TBuffer [▶ 583];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_101TBufferCtrl [▶ 506];

6.2.2.2.1.1 Download: Substation (slave, unbalanced)

Simple substation application using the TwinCAT IEC 60870-5-101 low-level interface. This sample application can communicate with a central station via the RS232 interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984618763.zip

In the sample project, the following station parameters are set for the substation:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**

- Link address: **220.**
- Link address size: **1 octet.**
- Cause of transfer size: **1 octet.**
- Originator address: **0 (not used).**
- Common ASDU address: **7.**
- Common ASDU address size: **1 octet.**
- Information object address size: **2 octets.**
- Link mode: **Unbalanced.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200).
- Regulating step value: M_ST_NA_1 (IOA := 300).
- Bitstring value: M_BO_NA_1 (IOA := 400).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10).
- Double command: C_DC_NA_1 (IOA := 20).
- Regulating step command: C_RC_NA_1 (IOA := 30).
- Bitstring command: C_BO_NA_1 (IOA := 40).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70).

Sample implementation of the following services:

- Test command (C_TS_NA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN), Mode A. The counter values are frozen locally every 60 seconds.
- Command execution of some commands (see upper data point configuration).
- End of initialization (M_EI_NA_1) is sent once to the central station (after the connection has been established).
- Cyclic data transfer every 60 seconds.
- Simulation and transfer of spontaneous data every 60 seconds if the variable **bSimulate** is set to TRUE.

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbServer	FB_IEC870_5_101S (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_101SServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
stProcData	ST_IEC870_5_101SProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_101SSysPara (STRUCT)	Configuration of the system parameters of the substation and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.2.2.2.1.2 Download: Substation (slave, balanced mode)

Simple substation application using the TwinCAT IEC 60870-5-101 low-level interface. This sample application can communicate with a central station via an RS232 interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984621707.zip

In the sample project, the following station parameters are set for the substation:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220.**
- Link address size: **1 octet.**
- Cause of transfer size: **1 octet.**
- Originator address: **0 (not used).**
- Common ASDU address: **7.**
- Common ASDU address size: **1 octet.**
- Information object address size: **2 octets.**
- Link mode: **Balanced.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_TA_1(IOA := 101), M_SP_TB_1(IOA := 102)
- Double point information: M_DP_NA_1 (IOA := 200)
- Regulating step value: M_ST_NA_1 (IOA := 300)
- Bitstring value: M_BO_NA_1 (IOA := 400)
- Measured value, normalized value: M_ME_NA_1 (IOA := 500)
- Measured value, scaled value: M_ME_NB_1 (IOA := 600)
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700)
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_TA_1 (IOA := 801), M_IT_TB_1 (IOA := 802)

- Single command: C_SC_NA_1 (IOA := 10)
- Double command: C_DC_NA_1 (IOA := 20)
- Regulating step command: C_RC_NA_1 (IOA := 30)
- Bitstring command: C_BO_NA_1 (IOA := 40)
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50)
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60)
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70)

Sample implementation of the following services:

- Test command (C_TS_NA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN), Mode A. The counter values are frozen locally every 60 seconds.
- Command execution of some commands (see upper data point configuration).
- End of initialization (M_EI_NA_1) is sent once to the central station (after the connection has been established).
- Cyclic data transfer every 60 seconds.
- Simulation and transfer of spontaneous data every 60 seconds if the variable **bSimulate** is set to TRUE.

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
fbServer	FB_IEC870_5_101S (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_101SServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
stProcData	ST_IEC870_5_101SProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_101SSysPara (STRUCT)	Configuration of the system parameters of the substation and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.2.2.2.1.3 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 721]
- Configuration of the serial KL6xxx Bus Terminals [▶ 722]
- Configuration of the serial EL6xxx Bus Terminals [▶ 722]

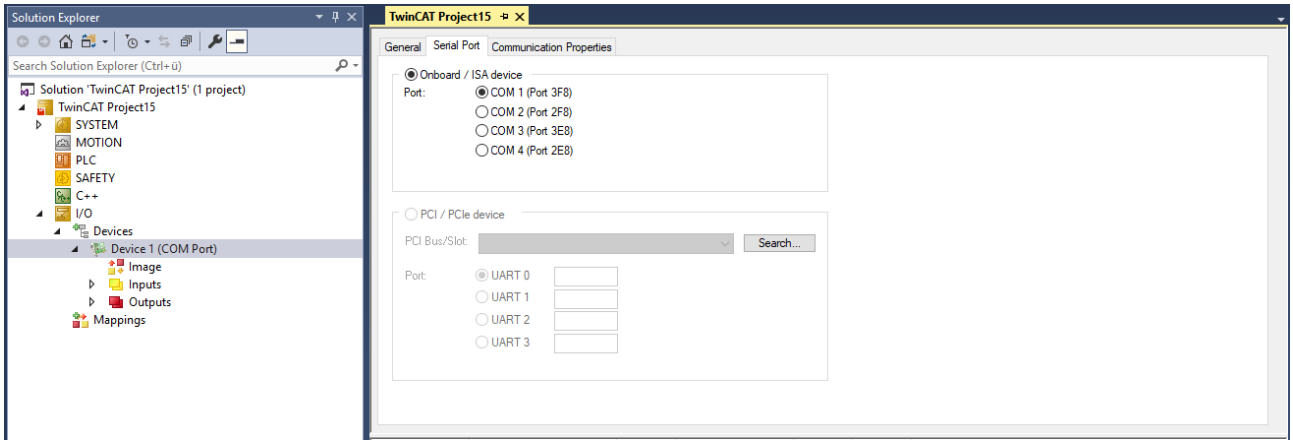
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

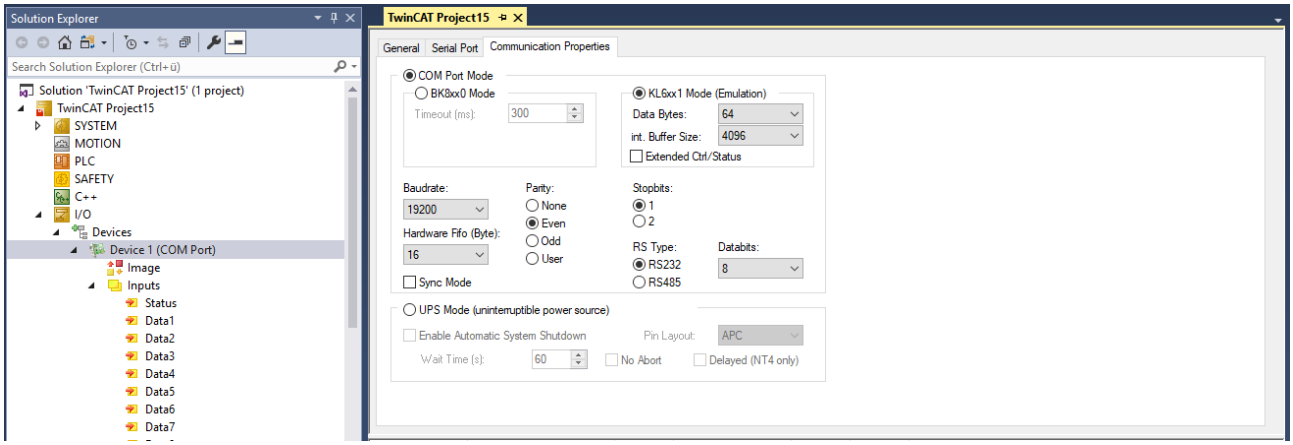
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be

adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

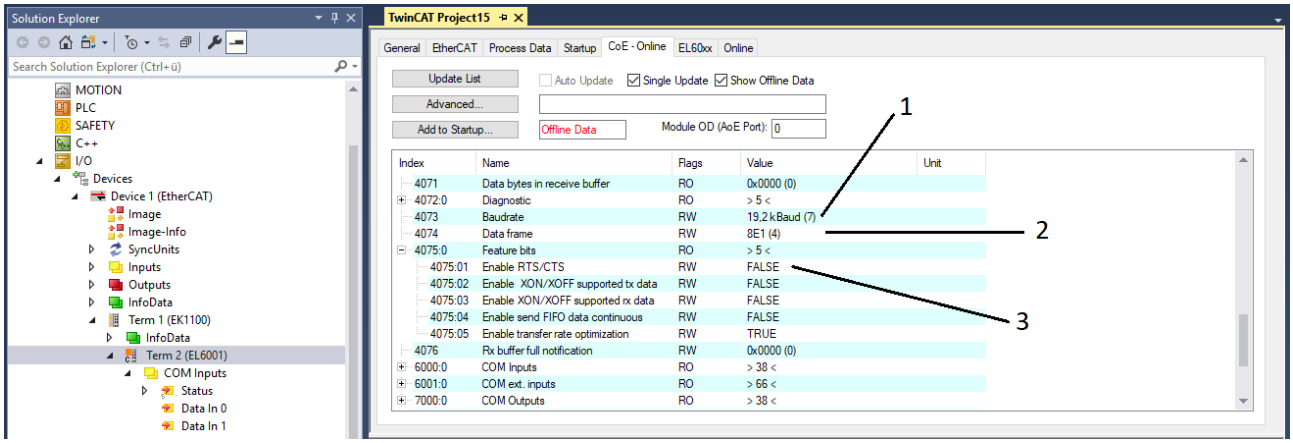
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

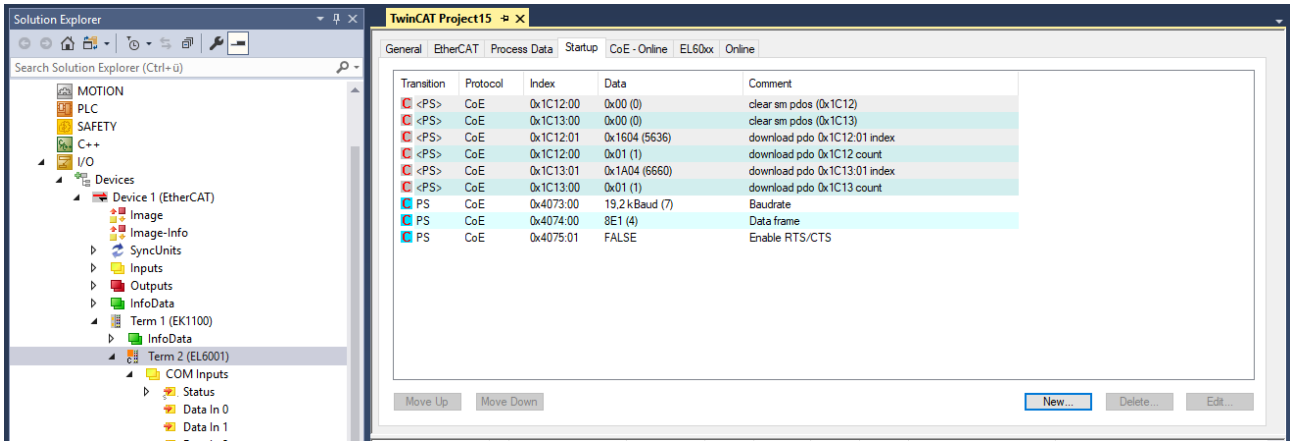
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.3 IEC 60870-5-102

6.3.1 Low-Level Interface

6.3.1.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-102 Central station low-level sample [▶ 724].

A PLC application that is to communicate with a substation via the transport interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_102TProtocol [▶ 511];
- An instance of the TX/RX data buffer: ST_IEC870_5_102TBuffer [▶ 645];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_102TBufferCtrl [▶ 508];

6.3.1.1.1 Download: Central station (master)

Simple central station application that uses the TwinCAT IEC 60870-5-102 low-level interface. This sample application can communicate with up to three substations via an RS232 interface. However, in the program only the communication to the first substation is activated.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984624651.zip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 baud.**
- Link address: **220 (first substation), 221 (second substation), 222 (third substation).**
- Link address size: **2 octets.**
- Common ASDU address: **7 (first substation), 8 (second substation), 9 (third substation).**
- Common ASDU address size: **2 octets.**

Sample implementation of the following services:

- Retrieves a list of single points with time tag of a selected period (C_SP_NB_2) every 30 seconds.
- Retrieves the manufacturer and product specification (C_RD_NA_2) every 10 seconds.

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
aClients	ARRAY OF FB_IEC870_5_102M (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_102MServices (FB)	Includes a simple implementation of some services such as reading the manufacturer/product specification, retrieving the list of single points, etc.
aProcData	ARRAY OF ST_IEC870_5_102MProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_102MSysPara (STRUCT)	Configuration of the system parameters of the central station and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.3.1.1.2 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 725]
- Configuration of the serial KL6xxx Bus Terminals [▶ 726]
- Configuration of the serial EL6xxx Bus Terminals [▶ 726]

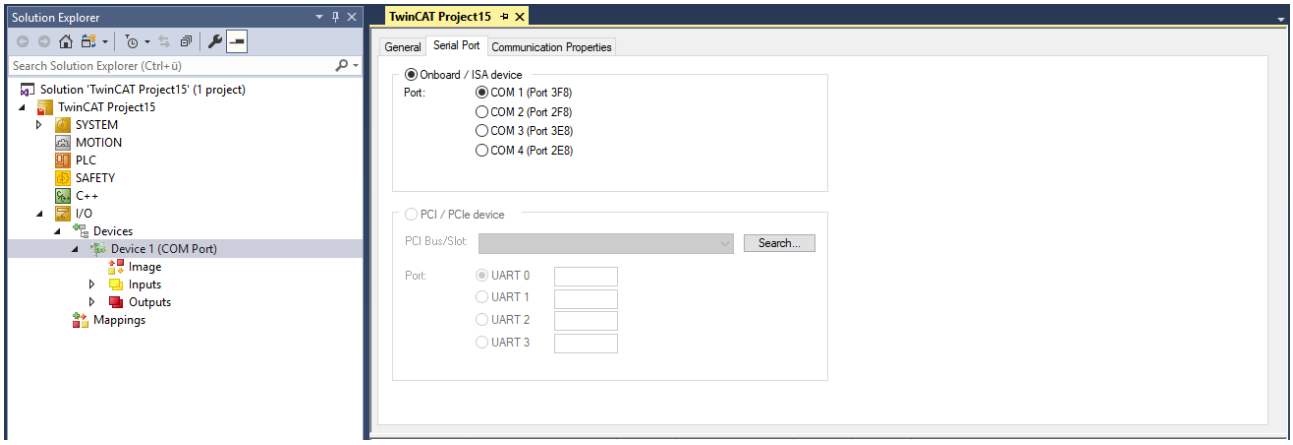
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

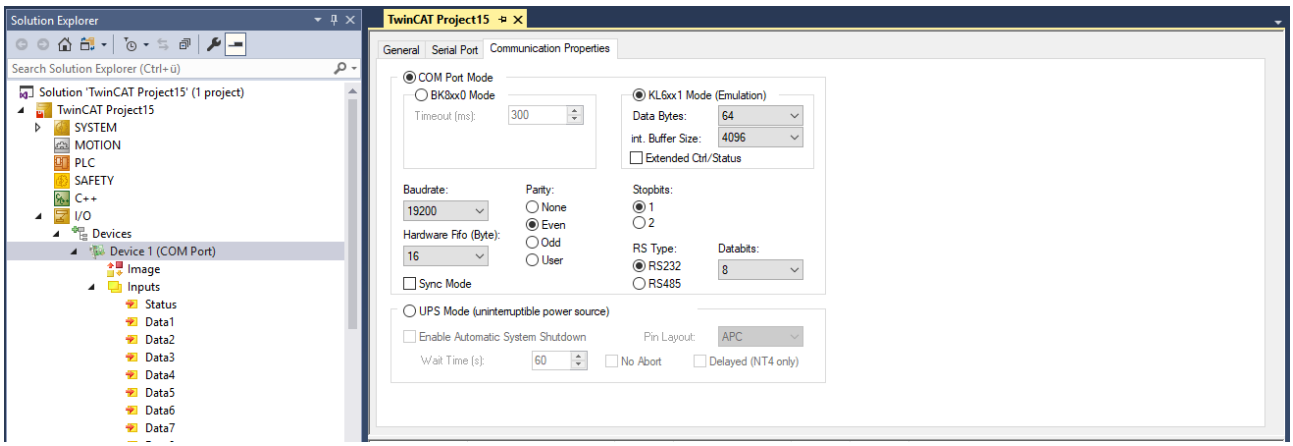
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

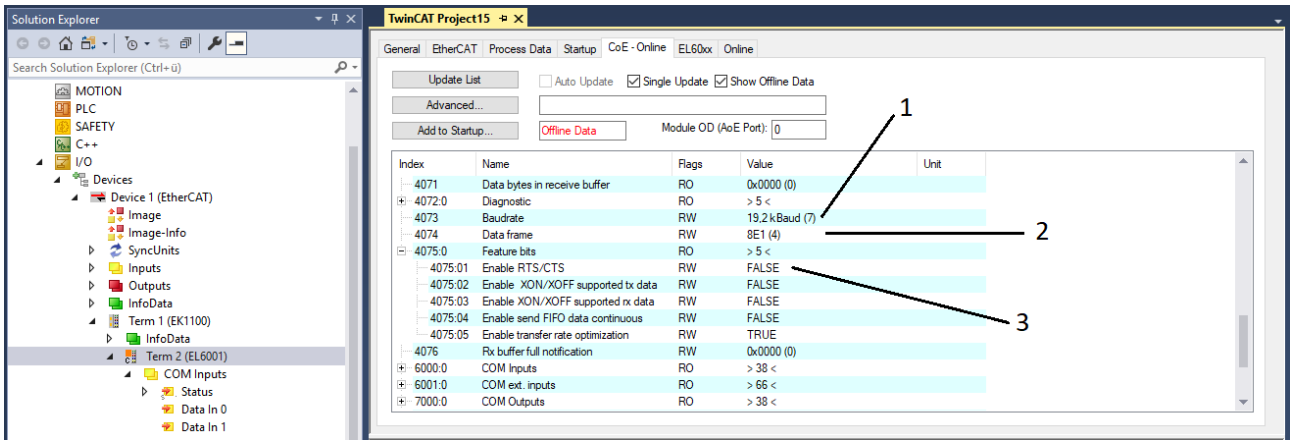
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

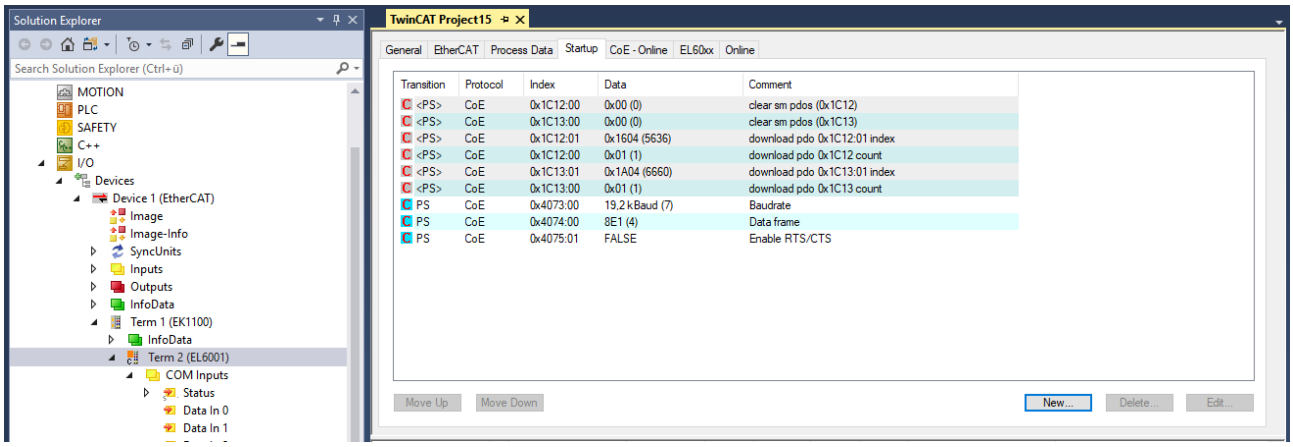
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.4 IEC 60870-5-103

6.4.1 Low-Level Interface

6.4.1.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-103 Central station low-level sample [▶ 727].

A PLC application that is to communicate with a substation via the transport interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_103TProtocol [▶ 513];
- An instance of the TX/RX data buffer: ST_IEC870_5_103TBuffer [▶ 648];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_103TBufferCtrl [▶ 509];

6.4.1.1.1 Download: Central station (master)

Simple central station application that uses the TwinCAT IEC 60870-5-103 low-level interface. This sample application can communicate with up to 10 substations via an RS232 interface. However, in the program only the communication to the first substation is activated.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984627595.tszip

In the sample project, the following station parameters are set for the central station:

- Standard PC interface: **COM1.**
- Baud rate: **19200 Baud** .
- Link address: **220 (first substation), 221 (second substation), 222 (third substation), ..., 229 (tenth substation).**
- Link address size: **1 octet (fixed).**
- Common ASDU address: **220 (first substation), 221 (second substation), 222 (third substation), ..., 229 (tenth substation).**
- Common ASDU address size: **1 octet (fixed).**

Sample implementation of the following services:

- Clock synchronization (C_SYN_TA_3) every 60 seconds.
- General interrogation (C_IGI_NA_3) every 60 seconds.
- General command (C_GRC_NA_3) if the member variable **ST_IEC870_5_103MProcData.eCmdCtrl** has been set to the value eIEC870_CMD_EXECUTE.
- Receipt of messages with timestamp (M_TTM_TA_3).

The following table briefly describes the main components of the project:

Component	Type	Description
P_MAIN_LowSpeed	PRG	This program part implements the telecontrol protocol (slow task).
P_SerialComm_HighSpeed	PRG	This program part implements TwinCAT background communication via the serial interfaces of the PC (fast task).
aClients	ARRAY OF FB_IEC870_5_103M (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_103MServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, etc.
aProcPara	ARRAY OF ST_IEC870_5_103MProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_103MSysPara (STRUCT)	Configuration of the system parameters of the central station and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.4.1.1.2 Configuration of serial interface

- Configuration of the standard PC COMx interfaces [▶ 728]
- Configuration of the serial KL6xxx Bus Terminals [▶ 729]
- Configuration of the serial EL6xxx Bus Terminals [▶ 729]

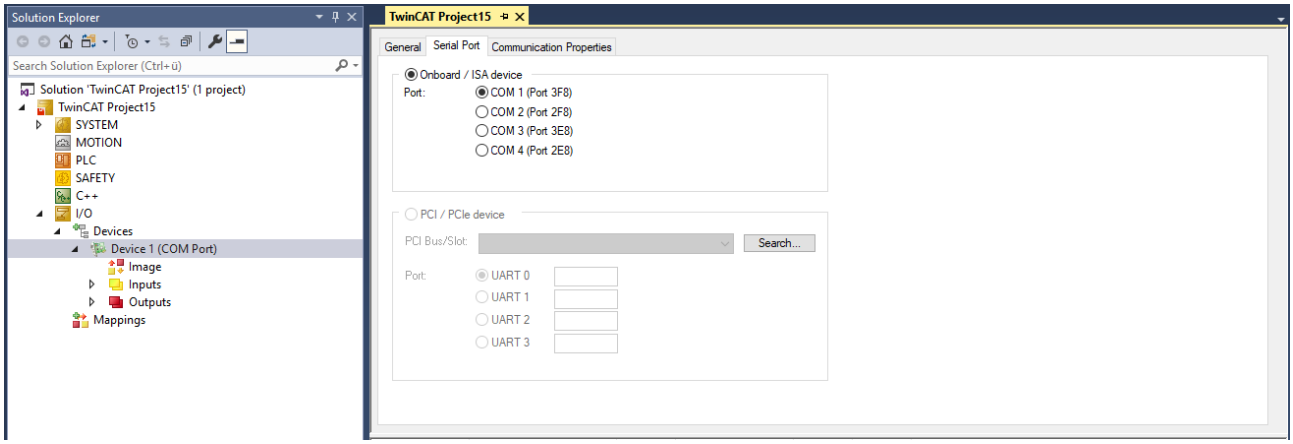
1. Configuration of the standard PC COMx interfaces

The serial interface can only be added in PC/CX systems.

Right-click on "I/O devices".Select "Append Device". Select "Serial Interface" under "Miscellaneous". Then set the following settings.

1.1 Serial Port tab

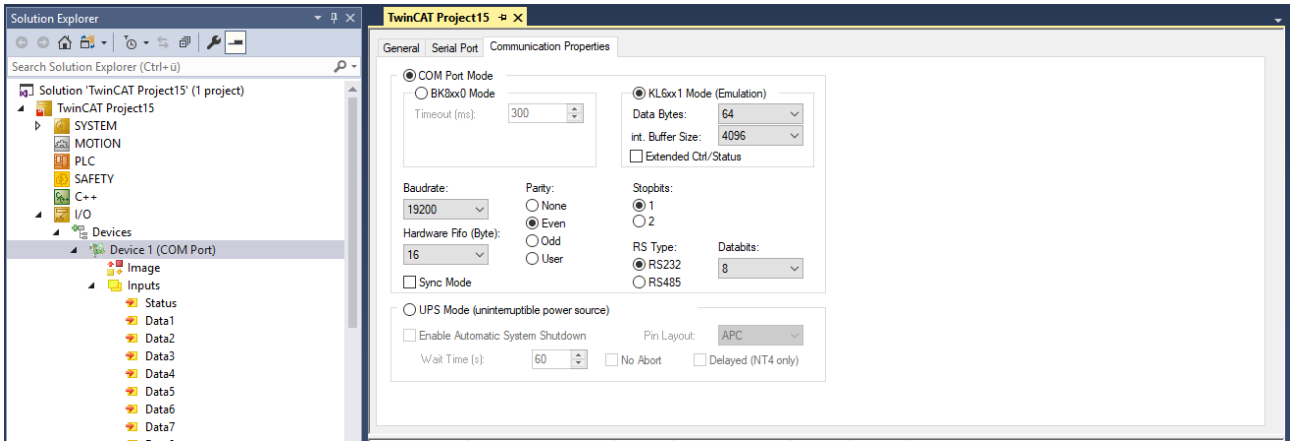
Serial interface COM1



1.2 Properties Tab

Select the option KL6xxx1 Mode (Emulation) and then set the communication parameters:

Baud rate, here 19200 baud, 8 data bits, parity=even, 1 stop bit



2. Configuration of the serial KL6xxx Bus Terminals

The interface is configured in the TwinCAT PLC by the instance of the FB_IEC870_SerialLineCtrl function block. The communication parameters such as baud rate, parity etc. must be set on this function block.

2.1 RS485 operating mode

In the RS485 operating mode (e.g. KL6041), the data is exchanged in half-duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. For half duplex mode, the handshake input of the FB_IEC870_SerialLineCtrl function block must be set to the value: RS485_HALFDUPLEX.

3. Configuration of the serial EL6xxx Bus Terminals

The serial interface is set in the TwinCAT System Manager CoE-Online tab (CoE=**C**anOpen **o**ver **E**therCAT). The tab is only available for online access to the terminal, i.e. the settings cannot be modified if the System Manager is not linked to the hardware. The settings are stored in the terminal in a non-volatile manner, i.e. the settings are retained even if the voltage supply fails. These data do not have to be re-entered after a terminal scan. If the terminal is replaced, the data from the new terminal become active and have to be adjusted. To prevent this, the changed data must be entered in the *Startup* tab. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

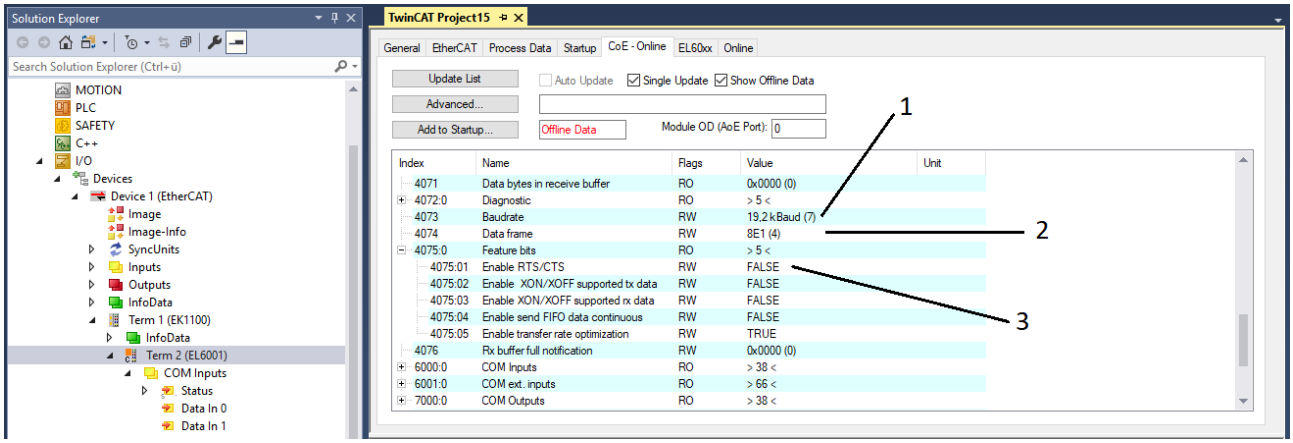
3.1 CoE Tab

(1) Baud rate, here 19200 baud

(2) Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

(3) Feature bits -> Enable RTS/CTS = *FALSE* for EL6001

Double-clicking on the corresponding row brings up a menu in which the settings can be modified.



3.1.1 RS485 operating mode

In RS485 mode, data are exchanged in half duplex mode. The transmit and receive lines are connected to one another in RS485 operating mode. As a result, the terminal receives not only the data from other devices, but also its own send data. This data can interfere with communication. With the option "Enable half duplex" in the "COM Settings" object this can be suppressed (Index 8000:06).

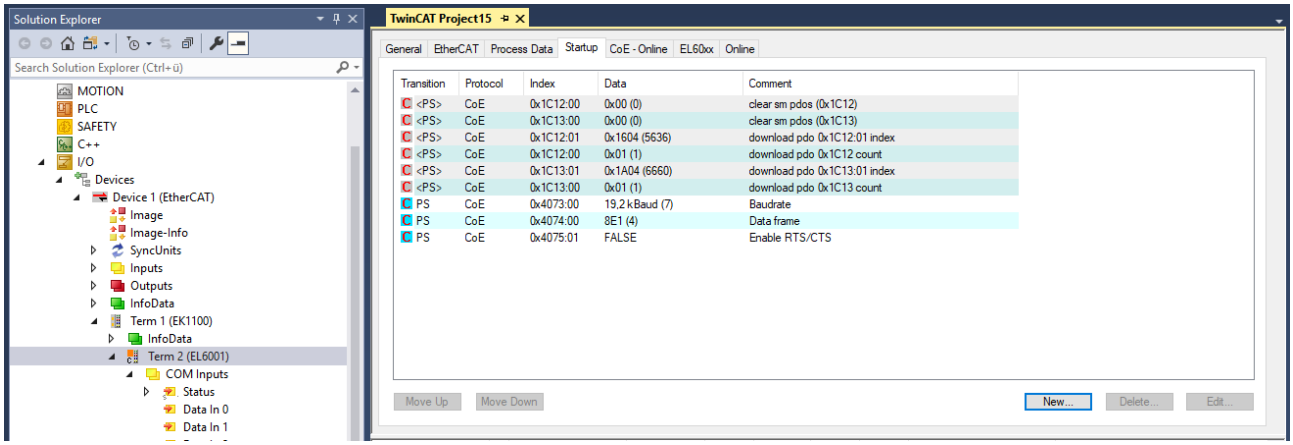
3.2 Startup tab

All non-standard settings should be entered in the Startup list. During system startup, the data from the *Startup* tab are transferred to the terminal. Terminals can then be replaced without problem. After a terminal scan, the data have to be re-entered in the *Startup* tab.

Baud rate, here 19200 baud

Date frame, 8E1 corresponds to 8 data bits, parity=even, 1 stop bit

Feature bits -> Enable RTS/CTS = *FALSE* for EL6001



6.5 IEC 60870-5-104

6.5.1 Central station

6.5.1.1 High-Level Interface

6.5.1.1.1 Quick start

Simple project with complete sources can be found here: IEC 60870-5-104 Control Station-High-Level example [▶ 732].

Interoperability check list can be found here: Interoperability check list [▶ 67]

Communication- and/or device error code overview can be found here: Error codes [▶ 805]

Short guide

Application object database

The application object database of the central station must be configured as hash table with the function F_iecCreateTableHnd [▶ 554]. The individual array elements are linked with each other in the form of a hash table. This enables faster access to the individual data points, but also has certain disadvantages that have to be taken into account:

- The size of the application database (array size) must not modified at runtime (e.g. through online change). The central station stops execution immediately and reports an error. The reason: The hash table links no longer match. When the program is modified it is best to load the complete project into the runtime system.
- The individual array elements must not be accessed via the index but via the special functions (e.g. F_iecAddTableEntry [▶ 555] etc.).
- With indexed table element access the internal configuration parameters must not be overwritten or modified. If the type, the ASDU address or the object address is changed the data point can no longer be found. To reconfigure a data point it should first be removed from the table via function call F_iecRemoveTableEntry [▶ 562]. The new data point can then be added.

An implementation in the form of linear table would mean that for each received ASDU (data unit) the central station would have to search the complete array for a suitable element. With a large number of data points this would lead to long execution times.

Protocol parameters

The TCP/IP transport layer characteristics are configured via the protocol parameters. Most protocol parameters are preconfigured with default values and do not have to be set explicitly. However, the PLC application has to configure the IP address (*sRemoteHost*) and the port address (*nRemotePort*) of the substation as a one-off operation.

System parameters

The system parameters are also preconfigured with default values. During commissioning it is useful to activate debugging output (*dbgMode*) in order to be able to locate any errors.

Parameters for cyclic data acquisition

The following parameters are preconfigured with default values:

- Initialisation sequence (consisting of a test command, time synchronisation, station query and counter query);
- Cyclic commands:
  - Test command every 60 s;
  - Time synchronisation every 60 s;
  - Group station query: generally every 60 s;
  - Group counter query: generally every 60 s;

6.5.1.1.2 Central station Minimal (Master)

Simple central station application that uses the TwinCAT IEC 60870-5-104 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984630539.zip

In the sample project, the following station parameters are set for the central station:

- Server host address (IP address of the substation): 127.0.0.1 (You must at least adapt this parameter on your target platform!).
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Originator address: **1.**
- Common ASDU address**: 7.**
- Max. APDU length**: 253.**

Data point configuration (it should be the same as the data point configuration of the substations):

- Single point information: M_SP_NA_1 (IOA := 100).
- Single command: C_SC_NA_1 (IOA := 10).

Initialization sequence that is executed once/automatically after the connection is established:

- Test command (C_TS_TA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN). Counter interrogation Mode C.

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 60 seconds.
- Clock synchronization command (C_CS_NA_1) every 60 seconds.

- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 60 seconds.
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN) every 60 seconds. Counter interrogation Mode C.
- If the value of the **bSCS** variable changes, the single command (C_SC_NA_1) is sent (direct command execution)
- When spontaneous data of type (M_SP_NA_1, IOA := 100) is received, the ON/OFF value is mapped to the PLC variable **bSPI.**

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol (slow task).
fbClient	FB_IEC870_5_104Master (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
bSPI	BOOL	The received ON/OFF value is mapped into this PLC variable directly to the allocated flag address: **AT%MX0.0**.
bSCS	BOOL	The ON/OFF value to be sent is mapped directly from the flag address **AT%MX1.5** of this allocated PLC variable.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
hTable	T_HAODBTable (STRUCT)	Application object database handle. These variables are used to access the application object database and never directly the individual array elements.
F_iecCreateTableHnd	FUNCTION	In the initialization step, all array elements of AODB are initialized once with this function call. If successful, this function call returns the handle.
F_iecAddTableEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.

6.5.1.1.3 Tutorial

This tutorial is a detailed, step-by-step guide to the configuration and programming of an IEC 60870-5-104 central station. The descriptions listed here are based on this PLC example tutorial [▶ 747].

6.5.1.1.3.1 Configuration of the programming environment

Proceed as follows for the basic configuration of the programming environment.

✓ Open Microsoft Visual Studio

1. Go to: File / New / Project
2. Select a TwinCAT project
3. Once the solution is open with the TwinCAT project, you can open the context menu by right-clicking on the PLC entry and select Add New Item
4. Select a standard PLC project
5. Expand the PLC project tree and select Add Library via the context menu of the References sub-point.

6. Add the following libraries as the basis for your IEC 60870-5-10x application.
   Tc2_Utilities
   Tc2_TcpIp
   Tc2_IEC60870_5_10x

⇨ You have thus completed the basic configuration of the programming environment.

6.5.1.1.3.2 Definition and configuration of application objects

Application objects mean the data types to be defined, such as single points, double points, measured values, short floating point values, etc.

In this example the commands were configured such that the process data for the commands are located in the same memory area as the information data in monitoring direction, although in a different byte and bit offset. If required the commands may have the same byte and bit offset as the information in monitoring direction.

Example:
C_SC_NA_1 with IOA = 10 on the same byte/bit offset as M_SP_NA_1 with IOA = 100 (both byte offset = 100 and bit offset = 0). In this case a change in value of the single point in the substation (M_SP_NA_1 with object address 100) would trigger a data transfer with cause of transmission <3> (spontaneous) to the control station. In the control station the new value is copied to the bit/byte offset address of the command (C_SC_NA_1 with object address 10) and triggers a command transfer to the substation.

As an example we will configure the following application objects as part of the introductory project:

ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset byte	Offset bit	Process data width in the TwinCAT PLC
M_SP_NA_1	100	General query	0	Flags	100	0	1 bit
M_SP_NA_1	101	General query	0	Flags	100	1	1 bit
M_SP_TB_1	102	General query	0	Flags	100	2	1 bit
M_DP_NA_1	200	General query	0	Flags	200	0	2 bits
M_DP_NA_1	201	General query	0	Flags	200	2	2 bits
M_DP_TB_1	202	General query	0	Flags	200	4	2 bits
M_ST_NA_1	300	General query	0	Flags	300	0	1 byte
M_ST_NA_1	301	General query	0	Flags	301	0	1 byte
M_ST_TB_1	302	General query	0	Flags	302	0	1 byte
M_BO_NA_1	400	General query	0	Flags	400	0	4 byte
M_BO_NA_1	401	General query	0	Flags	404	0	4 byte
M_BO_TB_1	402	General query	0	Flags	408	0	4 byte
M_ME_NA_1	500	General query	0	Flags	500	0	2 byte
M_ME_NA_1	501	General query	0	Flags	502	0	2 byte
M_ME_TD_1	502	General query	0	Flags	504	0	2 byte
M_ME_NB_1	600	General query	0	Flags	600	0	2 byte
M_ME_NB_1	601	General query	0	Flags	602	0	2 byte
M_ME_TE_1	602	General query	0	Flags	604	0	2 byte
M_ME_NC_1	700	General query	0	Flags	700	0	4 byte
M_ME_NC_1	701	General query	0	Flags	704	0	4 byte
M_ME_TF_1	702	General query	0	Flags	708	0	4 byte
M_IT_NA_1	800	General counter query	0	Flags	800	0	4 byte
M_IT_NA_1	801	General counter query	0	Flags	804	0	4 byte
M_IT_TB_1	802	General counter query	0	Flags	808	0	4 byte
Commands							

C_SC_NA_1	10	-	0	Flags	2100	0	1 bit
C_SC_NA_1	11	-	0	Flags	2100	1	1 bit
C_SC_TA_1	12	-	0	Flags	2100	2	1 bit
C_DC_NA_1	20	-	0	Flags	2200	0	2 bit
C_DC_NA_1	21	-	0	Flags	2200	2	2 bit
C_DC_TA_1	22	-	0	Flags	2200	4	2 bit
C_BO_NA_1	40	-	0	Flags	2400	0	4 byte
C_BO_NA_1	41	-	0	Flags	2404	0	4 byte
C_BO_TA_1	42	-	0	Flags	2408	0	4 byte
C_SE_NA_1	50	-	0	Flags	2500	0	2 byte
C_SE_NA_1	51	-	0	Flags	2502	0	2 byte
C_SE_TA_1	52	-	0	Flags	2504	0	2 byte
C_SE_NB_1	60	-	0	Flags	2600	0	2 byte
C_SE_NB_1	61	-	0	Flags	2602	0	2 byte
C_SE_TB_1	62	-	0	Flags	2604	0	2 byte
C_SE_NC_1	70	-	0	Flags	2700	0	4 byte
C_SE_NC_1	71	-	0	Flags	2704	0	4 byte
C_SE_TC_1	72	-	0	Flags	2708	0	4 byte

Declaring a database variable

The application object database is an array variable of type ST_IEC870_5_101AODBEntry [▶ 584]. Each array element corresponds to an application object. The array elements are not manipulated directly but through specially provided functions and a database handle (table handle). The database handle must be initialized via a F_iecCreateTableHnd [▶ 554] function call before it can be used. During this process the array elements are linked as a hash table. With a larger number of data points the hash table enables faster access to individual data points.

The maximum number of application objects is freely selectable and is only limited by the available memory. During PLC programming you have to specify a constant maximum number. The maximum number of application objects cannot be changed at runtime. In our example 50 application objects are declared. This number is sufficient for most applications. Please note that many application objects require adequate memory and runtime resources.

Define the following variables in MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry [▶ 584];
hTable : T_HAODBTable [▶ 633];
```

```
END_VAR
```

Configuring application objects

The required application objects are configured during program runtime. Amongst other things, the object type (M_SP_NA_1, M_DP_NA_1, M_ST_NA_1, etc.), the common ASDU address, the object address and further object parameters are defined during the configuration.

After initialization of the database handle is the application object database (database array) is empty and must be filled with the required data (data points). The configuration of the data points in the central station must correspond to the configuration of the data points in the substation! This means that data points configured in the central station must be of the same type, with the same common ASDU address and with the same information object address as in the substation. Other parameters such as the mapping range, byte and bit offset can be configured as desired.

The following functions are available for manipulating the application database:

Function	Description
F_iecCreateTableHnd [▶ 554]	Initializes the hash table handle
F_iecAddTableEntry [▶ 555]	Configures and inserts a new hash table entry
F_iecRemoveTableEntry [▶ 562]	Removes a hash table entry
F_iecLookupTableEntry [▶ 560]	Checks whether a certain hash table entry exists
F_iecGetPosOfTableEntry [▶ 558]	Determines the linear position of a hash table entry

The database handle must be transmitted to the function via VAR_IN_OUT. Configuration is usually carried out once during PLC program start-up via an Init routine.

In order to configure the application objects during program start-up, add the following PLC code in MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry [▶ 584];
hTable : T_HAODBTable [▶ 633];

init : BOOL := TRUE;
initError : UDINT;
asduAddr : UDINT := 7;
END_VAR

IF init THEN
init := FALSE;

initError := F_iecCreateTableHnd [▶ 554]( ADR( AODB ), SIZEOF( AODB ), hTable );
IF initError <> 0 THEN
ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG,
'F_iecCreateTableHnd() error: %s',
DWORD_TO_HEXSTR( initError, 8, FALSE ) );
RETURN;
END_IF


(* Monitored Single Points *)
initError := F_iecAddTableEntry( M_SP_NA_1, asduAddr, 100, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
100, 0, 0, hTable );
initError := F_iecAddTableEntry( M_SP_NA_1, asduAddr, 101, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
100, 1, 0, hTable );
initError := F_iecAddTableEntry( M_SP_TB_1, asduAddr, 102, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
100, 2, 0, hTable );
(* Double Points*)
initError := F_iecAddTableEntry( M_DP_NA_1, asduAddr, 200, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
200, 0, 0, hTable );
initError := F_iecAddTableEntry( M_DP_NA_1, asduAddr, 201, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
200, 2, 0, hTable );
initError := F_iecAddTableEntry( M_DP_TB_1, asduAddr, 202, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
200, 4, 0, hTable );
(* Regulating step value *)
initError := F_iecAddTableEntry( M_ST_NA_1, asduAddr, 300, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
300, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ST_NA_1, asduAddr, 301, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
301, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ST_TB_1, asduAddr, 302, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
```

```
302, 0, 0, hTable );
(* 32 bit string *)
initError := F_iecAddTableEntry( M_BO_NA_1, asduAddr, 400, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
400, 0, 0, hTable );
initError := F_iecAddTableEntry( M_BO_NA_1, asduAddr, 401, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
404, 0, 0, hTable );
initError := F_iecAddTableEntry( M_BO_TB_1, asduAddr, 402, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
408, 0, 0, hTable );
(* Measured value, normalized value *)
initError := F_iecAddTableEntry( M_ME_NA_1, asduAddr, 500, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
500, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_NA_1, asduAddr, 501, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
502, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_TD_1, asduAddr, 502, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
504, 0, 0, hTable );
(* Mesured value, scaled value *)
initError := F_iecAddTableEntry( M_ME_NB_1, asduAddr, 600, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
600, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_NB_1, asduAddr, 601, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
602, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_TE_1, asduAddr, 602, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
604, 0, 0, hTable );
(* Measured value , short floating point value *)
initError := F_iecAddTableEntry( M_ME_NC_1, asduAddr, 700, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
700, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_NC_1, asduAddr, 701, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
704, 0, 0, hTable );
initError := F_iecAddTableEntry( M_ME_TF_1, asduAddr, 702, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY,
708, 0, 0, hTable );
(* Integrated totals *)
initError := F_iecAddTableEntry( M_IT_NA_1, asduAddr, 800, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMORY,
800, 0, 0, hTable );
initError := F_iecAddTableEntry( M_IT_NA_1, asduAddr, 801, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMORY,
804, 0, 0, hTable );
initError := F_iecAddTableEntry( M_IT_TB_1, asduAddr, 802, IEC870_GRP_REQCOGEN, 0, MAP_AREA_MEMORY,
808, 0, 0, hTable );

(* Single commands *)
initError := F_iecAddTableEntry( C_SC_NA_1, asduAddr, 10, 0, 0, MAP_AREA_MEMORY, 2100, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SC_NA_1, asduAddr, 11, 0, 0, MAP_AREA_MEMORY, 2100, 1, 0,
hTable );
initError := F_iecAddTableEntry( C_SC_TA_1, asduAddr, 12, 0, 0, MAP_AREA_MEMORY, 2100, 2, 0,
hTable );
(* Double commands *)
initError := F_iecAddTableEntry( C_DC_NA_1, asduAddr, 20, 0, 0, MAP_AREA_MEMORY, 2200, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_DC_NA_1, asduAddr, 21, 0, 0, MAP_AREA_MEMORY, 2200, 2, 0,
hTable );
initError := F_iecAddTableEntry( C_DC_TA_1, asduAddr, 22, 0, 0, MAP_AREA_MEMORY, 2200, 4, 0,
hTable );
(* 32 bit string commands *)
initError := F_iecAddTableEntry( C_BO_NA_1, asduAddr, 40, 0, 0, MAP_AREA_MEMORY, 2400, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_BO_NA_1, asduAddr, 41, 0, 0, MAP_AREA_MEMORY, 2404, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_BO_TA_1, asduAddr, 42, 0, 0, MAP_AREA_MEMORY, 2408, 0, 0,
hTable );
(* Set point, normalized values*)
initError := F_iecAddTableEntry( C_SE_NA_1, asduAddr, 50, 0, 0, MAP_AREA_MEMORY, 2500, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_NA_1, asduAddr, 51, 0, 0, MAP_AREA_MEMORY, 2502, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_TA_1, asduAddr, 52, 0, 0, MAP_AREA_MEMORY, 2504, 0, 0,
hTable );
(* Set point, scaled valuess *)
initError := F_iecAddTableEntry( C_SE_NB_1, asduAddr, 60, 0, 0, MAP_AREA_MEMORY, 2600, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_NB_1, asduAddr, 61, 0, 0, MAP_AREA_MEMORY, 2602, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_TB_1, asduAddr, 62, 0, 0, MAP_AREA_MEMORY, 2604, 0, 0,
hTable );
(* Set point, short floating point values *)
initError := F_iecAddTableEntry( C_SE_NC_1, asduAddr, 70, 0, 0, MAP_AREA_MEMORY, 2700, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_NC_1, asduAddr, 71, 0, 0, MAP_AREA_MEMORY, 2704, 0, 0,
hTable );
initError := F_iecAddTableEntry( C_SE_TC_1, asduAddr, 72, 0, 0, MAP_AREA_MEMORY, 2708, 0, 0,
```

```
hTable );

END_IF
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.3 Mapping the PLC and IEC process data

The TwinCAT PLC process data are cyclically mapped (copied) into the IEC process data (application objects) and vice versa at program runtime. Up to four process data ranges (IO inputs, IO outputs, flag range, data range) can be declared in the PLC program as buffer variables for the mapping of the IEC<->PLC process data. The byte size of the buffers is freely selectable and may be different for each area. Unused ranges need not necessarily be declared.

In our introductory example we declare 4 PLC process data areas with 3000 bytes each:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry [▶ 584];
hTable : T_HAODBTable;

init : BOOL := TRUE;
initError : UDINT;
asduAddr : UDINT := 7;


inputs AT%IB0 : ARRAY[0..2999] OF BYTE;
outputs AT%QB0 : ARRAY[0..2999] OF BYTE;
memory AT%MB0 : ARRAY[0..2999] OF BYTE;
data : ARRAY[0..2999] OF BYTE;

END_VAR
```

How the process data are to be mapped at runtime is specified during configuration of the application objects via the F_iecAddTableEntry [▶ 555] function.

See also in: Definition and configuration of application objects [▶ 734].

The buffer variables were now declared as byte arrays. In order to improve access to the required data we define the individual variables a second time and allocate them to the corresponding byte/bit offset addresses. In case of a change in the byte array, the corresponding individual variable will be changed and inverted at the same time. However, this is not absolutely necessary. The bytes/bits of the byte array buffer variables can be accessed directly.

```
VAR_GLOBAL(* Memory offset 0..99 unused *)
(* Single points *)
msgSingle_0 AT%MX100.0 : BOOL;
msgSingle_1 AT%MX100.1 : BOOL;
msgSingle_2 AT%MX100.2 : BOOL;

(* Double points *)
(* Bit 0..1 = first double point,
Bit 2..3 = second double point,
Bit 4..5 = third double point,
Bit 6..7 = fourth double point *)
msgDouble_0 AT%MB200 : BYTE;

(* Regulating step values *)
msgStep_0 AT%MB300 : BYTE;
msgStep_1 AT%MB301 : BYTE;
msgStep_2 AT%MB302 : BYTE;

(* 32 bit strings *)
msgBitStr_0 AT%MD400 : DWORD;
msgBitStr_1 AT%MD404 : DWORD;
msgBitStr_2 AT%MD408 : DWORD;

(* Measured values, normalized values *)
msgNormalized_0 AT%MW500 : WORD;
msgNormalized_1 AT%MW502 : WORD;
msgNormalized_2 AT%MW504 : WORD;

(* Measured values, scaled values *)
```

```
msgScaled_0 AT%MW600 : INT;
msgScaled_1 AT%MW602 : INT;
msgScaled_2 AT%MW604 : INT;

(* Measured values, short floating point values *)
msgFloating_0 AT%MD700 : REAL;
msgFloating_1 AT%MD704 : REAL;
msgFloating_2 AT%MD708 : REAL;

(* Integrated totals *)
msgTotal_0 AT%MD800 : UDINT;
msgTotal_1 AT%MD804 : UDINT;
msgTotal_2 AT%MD808 : UDINT;

(* Single commands *)
cmdSingle_0 AT%MX2100.0 : BOOL;
cmdSingle_1 AT%MX2100.1 : BOOL;
cmdSingle_2 AT%MX2100.2 : BOOL;

(* Double commands *)
(* Bit 0..1 = first double command,
Bit 2..3 = second double command,
Bit 4..5 = third double command,
Bit 6..7 = fourth double command *)
cmdDouble_0 AT%MB2200 : BYTE;


(* 32 bit string commands *)
cmdBitStr_0 AT%MD2400 : DWORD;
cmdBitStr_1 AT%MD2404 : DWORD;
cmdBitStr_2 AT%MD2408 : DWORD;

(* Set point, normalized values *)
cmdNormalized_0 AT%MW2500 : WORD;
cmdNormalized_1 AT%MW2502 : WORD;
cmdNormalized_2 AT%MW2504 : WORD;

(* Set point, scaled values *)
cmdScaled_0 AT%MW2600 : INT;
cmdScaled_1 AT%MW2602 : INT;
cmdScaled_2 AT%MW2604 : INT;

(* Set point, short floating point values *)
cmdFloating_0 AT%MD2700 : REAL;
cmdFloating_1 AT%MD2704 : REAL;
cmdFloating_2 AT%MD2708 : REAL;
END_VAR
```

Mapping of the IEC<->PLC process data in the central station

Process data in the monitoring direction (Slave->Master information)

Example 1

Single point information (M_SP_NA_1) with the IOA 100, PLC flag range, byte offset = 100, bit offset = 0.

Substation -> ... -> Central Station FB -> Memory[100].0 == msgSingle_0

Example 2

Measured value, short floating point value (M_ME_NC_1) with the IOA 700, PLC flag range, byte offset = 700, bit offset = 0 (no meaning).

Substation -> ... -> Central Station FB -> Memory[700 to 703] == msgFloating_0

Process data in the control direction (Master->Slave commands)

Example 1

Single command state (C_SC_NA_1) with the IOA 10, PLC flag range, byte offset = 2100, bit offset = 0.

cmdSingle_0 == memory[2100].0 -> Central Station FB -> ... -> Substation

Example 2

Set point, short floating point value (C_SE_NC_1) with the IOA 70, PLC flag range, byte offset = 2700, bit offset = 0 (no meaning).

cmdFloating_0 == memory[2700 to 2703] -> Central Station FB -> ... -> Substation

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.4 Calling and declaration of the central station

The entire functionality of a central station is encapsulated in the function block FB_IEC870_5_104Master. A connection to the substation can be established with an instance. For a further connection, declare a further instance of the function block.

Insert the following PLC code in the declaration part of MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;
hTable : T_HAODBTable;

init : BOOL := TRUE;
initError : UDINT;
asduAddr : UDINT := 7;

inputs AT%IB0 : ARRAY[0..2999] OF BYTE;
outputs AT%QB0 : ARRAY[0..2999] OF BYTE;
memory AT%MB0 : ARRAY[0..2999] OF BYTE;
data : ARRAY[0..2999] OF BYTE;

bEnable : BOOL := TRUE;
client : FB_IEC870_5_104Master;

END_VAR
```

The instance is called in the program part:

```
IF init THEN
init := FALSE;
...



ELSE
...
client( pAOEntries := ADR( AODB ),
cbAOEntries := SIZEOF( AODB ),
pInputs := ADR( inputs ),
cbInputs := SIZEOF( inputs ),
pOutputs := ADR( outputs ),
cbOutputs := SIZEOF( outputs ),
pMemory := ADR( memory ),
cbMemory := SIZEOF( memory ),
pData := ADR( data ),
cbData := SIZEOF( data ),
bEnable := bEnable,
hTable := hTable );
...
END_IF
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.5 Basic settings: Protocol parameters

The behavior of the central station can be adapted to the requirements of the substation via the IEC60870-5-104 protocol parameters. Most parameters have preallocated default values that do not have to be changed.

In our example we change the values of the iK and iW parameters and configure the IP address and port number of the substation to which the connection is to be established.

```
IF init THEN
        init := FALSE;
...

        client.protPara.sRemoteHost := '127.0.0.1';
        client.protPara.nRemotePort := 2404;
        client.protPara.iK := 12;
        client.protPara.iW := 8;
        client.protPara.bThrottleMode := TRUE;
        client.protPara.bPackFrames := TRUE;

ELSE
    client( pInputs := ADR( inputs ),
                cbInputs := SIZEOF( inputs ),
                pOutputs := ADR( outputs ),
...

END_IF
```

The documentation of all transmission protocol parameters can be found under ST_IEC870_5_104ProtocolParams [▶ 651].

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.6 Basic settings: System parameters

The common ASDU address and the user functions of the central station are configured via the system parameters. In our introduction we configure the following system parameters:

The common ASDU address is set to 7. (*asduAddr*)

The logging of the debug messages in the application log is activated (*dbgMode*). Changes in the device status are logged.

Add the following PLC code to your PLC project:

```
IF init THEN
init := FALSE;
...

client.sysPara.asduAddr := 7;
client.sysPara.dbgMode := IEC870_DEBUGMODE_DEVSTATE(* OR IEC870_DEBUGMODE_LINKERROR OR
IEC870_DEBUGMODE_ASDU OR IEC870_DEBUGMODE_LINKLAYER *);

...
ELSE
client( pInputs := ADR( inputs ),
cbInputs := SIZEOF( inputs ),
pOutputs := ADR( outputs ),
...

END_IF
```

The documentation for all system parameters can be found here: ST_IEC870_5_101SystemParams [▶ 588].

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.7 Initialization sequence

The initialization sequence is as follows:

```
client.acqPara.arrInitSeq[0] := eIEC870_ISTEP_TEST; (* Send test command *)
client.acqPara.arrInitSeq[1] := eIEC870_ISTEP_CLOCK; (* Send clock synchronization command *)
client.acqPara.arrInitSeq[2] := eIEC870_ISTEP_GENRO; (* Send general interrogation command *)
client.acqPara.arrInitSeq[3] := eIEC870_ISTEP_CORO; (* Send counter interrogation command *)
client.acqPara.arrInitSeq[5] := eIEC870_ISTEP_UNUSED; (* not used *)
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.8 Station query / general query

The station query command, also called the general query, is initiated from the central station. The ID field of the command also contains the group (1 to 16 or general). The substation transfers the application objects associated with this group to the central station with cause of transmission <*20*> to <*36*>.

Time mark in the case of general queries

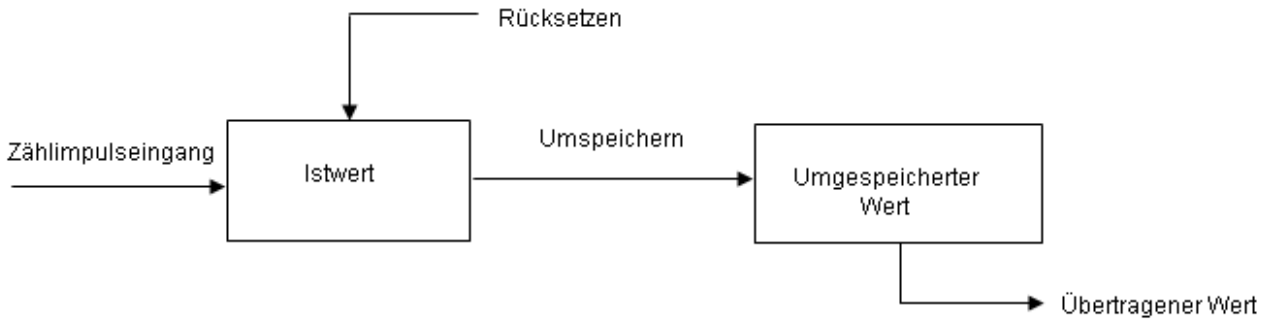
Application objects with time marks are transmitted without time mark in the case of a general query.

```
client.acqPara.arrGenro[0].tPollCycle := T#60s;
client.acqPara.arrGenro[0].eQOI := eIEC870_QOI_INROGEN;
client.acqPara.arrGenro[0].bEnable := TRUE;
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.9 Count transfer

For explanation purposes, here is a general model for the transmission of integrated totals:



The actual values are added by counters. Via a re-store command that is either received by the central station or generated locally (in the substation), the actual values can be re-stored (copied) periodically into re-stored values. After freezing, the recorded value is either reset to zero (logging of incremental values), or the counter continues adding up (logging of counter readings).

Application objects with counts are assigned to groups. The groups are frozen individually, reset, or transferred. The central station sends count query commands to the substation. The task to be carried out (FRZ) and the group (RQT) are specified in an ID field of the command (QCC).

The allocation of the application objects to the individual groups (1 to 4 or general) is specified via the group flag parameter during configuration. There are four operating modes for recording counter readings and incremental values. Each mode includes notes about the configuration of the system parameters or the application objects.

Mode A: local freeze with spontaneous transfer

The substation internally initiates freeze or freeze with reset. The frozen counts are transferred spontaneously, once the freeze or freeze with reset function was executed. In this mode the central station does not issue any count query commands.

Configuration of the system parameters:

Configuration of the application objects:

Operating mode B: Local freeze with counter interrogation

The substation internally initiates freeze or freeze with reset. The central station queries the frozen counts via count query commands. In this case the central station must not use freeze or freeze with reset in the command ID field (FRZ=0). The counts are queried generally or in groups 1 to 4.

Configuration of the system parameters:

Configuration of the application objects:

Operating mode C: The central station initiates freeze, freeze with reset, or reset

The central station periodically issues a count query command to the substation for controlling the freeze or (and) reset process. This command does not result in a count transfer. The central station sends a subsequent count query command for collecting the frozen counts. This is similar to operating mode B.

Configuration of the system parameters:

Configuration of the application objects:

Operating mode D: The central station initiates freeze and (or) reset, and the frozen values are transferred spontaneously

This mode is a combination of the count command from the central station as in mode C and spontaneous transfer of the counts as in mode A.

Configuration of the system parameters:

Configuration of the application objects:

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.10 Clock synchronization

The configuration parameters for the cyclic exchange of data are located in the Acquisition Parameters [▶ 593]. The clock synchronization can also be switched on and configured there. The following values must be set in the initialization:

```
 //activate clock sync
fbMaster.acqPara.clockSync.bEnable := TRUE;
fbMaster.acqPara.clockSync.tPollCycle := T#120S;
```

Furthermore, several settings can be made in the System Parameters [▶ 588] with regard to the time synchronization. The following is relevant for the central station:

bSyncPCTime: if this value is TRUE, the system time of the TwinCAT PC (the Windows system time in the taskbar) should be synchronized in addition to the IEC system time. The default value is FALSE.

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.11 Test of the communication

This chapter refers directly to the example tutorial. A simple simulation of the commands in the control direction is activated by setting the *bExecuteCmd* variable to TRUE and deactivated by setting it to FALSE. If there is an active connection, a single command (C_SC_NA_1, IOA = 10) is transmitted cyclically to the substation every 10 seconds in this example.

```
PROGRAM MAIN
VAR
...

bExecuteCmd : BOOL;
timer : TON;

...
END_VAR

...

(* Simple command simulation *)
timer( IN := bExecuteCmd, PT := T#10s ); (* Send cyclic command *)
IF timer.Q THEN
timer( IN := FALSE );
cmdSingle_0 := NOT cmdSingle_0; (* toggle single command ON<->OFF *)
```

```
(* cmdDouble_0 := SEL( cmdDouble_0 = 1, 1, 2 );

cmdBitStr_0 := cmdBitStr_0 + 1;

cmdNormalized_0 := cmdNormalized_0 + 2;

cmdScaled_0 := cmdScaled_0 + 4;

cmdFloating_0 := cmdFloating_0 + 1.2; *)
END_IF

...
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.12 Protocol and station errors

The station error messages are stored in a FIFO memory. Up to 10 error messages can be buffered. In the case of fatal communication errors (e.g. errors of the connection layer, the checksum of the frame is wrong) the connection is interrupted and must be re-established. Errors in the application layer (e.g. the ASDU transmit buffer has overflowed due to too many frames) are only logged and do not lead to interruption of the connection. Even with these errors it is still possible to interrupt the connection from the application. Apart from the error code, the error source is also stored in the error message. This makes it easier to locate the error.

Example

The accumulated error messages of a 60870-5-104 central station can be read out using the following call:

```
PROGRAM MAIN
VAR
...

client : FB_IEC870_5_104Master;

...
END_VAR

...

REPEAT
client.system.device.errors.RemoveError( );
IF client.system.device.errors.bOk THEN
ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG,
'IEC60870-5-104 master error: 0x%s',
DWORD_TO_HEXSTR( client.system.device.errors.getError.nErrId, 8, FALSE) );
END_IF
UNTIL NOT client.system.device.errors.bOk
END_REPEAT

...
```

You can download the corresponding PLC tutorial example on the Central Station Tutorial (Master) [▶ 747] page.

6.5.1.1.3.13 Extended diagnosis

Debug messages written to the application log facilitate debugging of the system. Currently, four levels of debug messages can be activated in an IEC application. The messages are activated via the dbgMode system parameter of the control station (ST_IEC870_5_101SystemParams [▶ 588]).

Debug messages that are logged when the TCP/IP connection is established or released, plus link layer error messages (dbgMode: IEC870_DEBUGMODE_LINKERROR);

Station status messages (dbgMode: IEC870_DEBUGMODE_DEVSTATE);

Hexadecimal output of the ASDUs (dbgMode: IEC870_DEBUGMODE_ASDU). 32 ASDU data bytes per row are output as hexadecimal numbers. Longer ASDUs are spread across several rows.

Hexadecimal output of the APDUs (TCP/IP telegrams, dbgMode: IEC870_DEBUGMODE_LINKLAYER). 32 APDU data bytes per row are output as hexadecimal numbers. Like in 3., longer APDUs are spread across several rows.

In order to view the activated debug messages, start the TwinCAT System Manager and activate the logger view. A debugging output is shown below. The first three different types of message are marked with corresponding numbers.

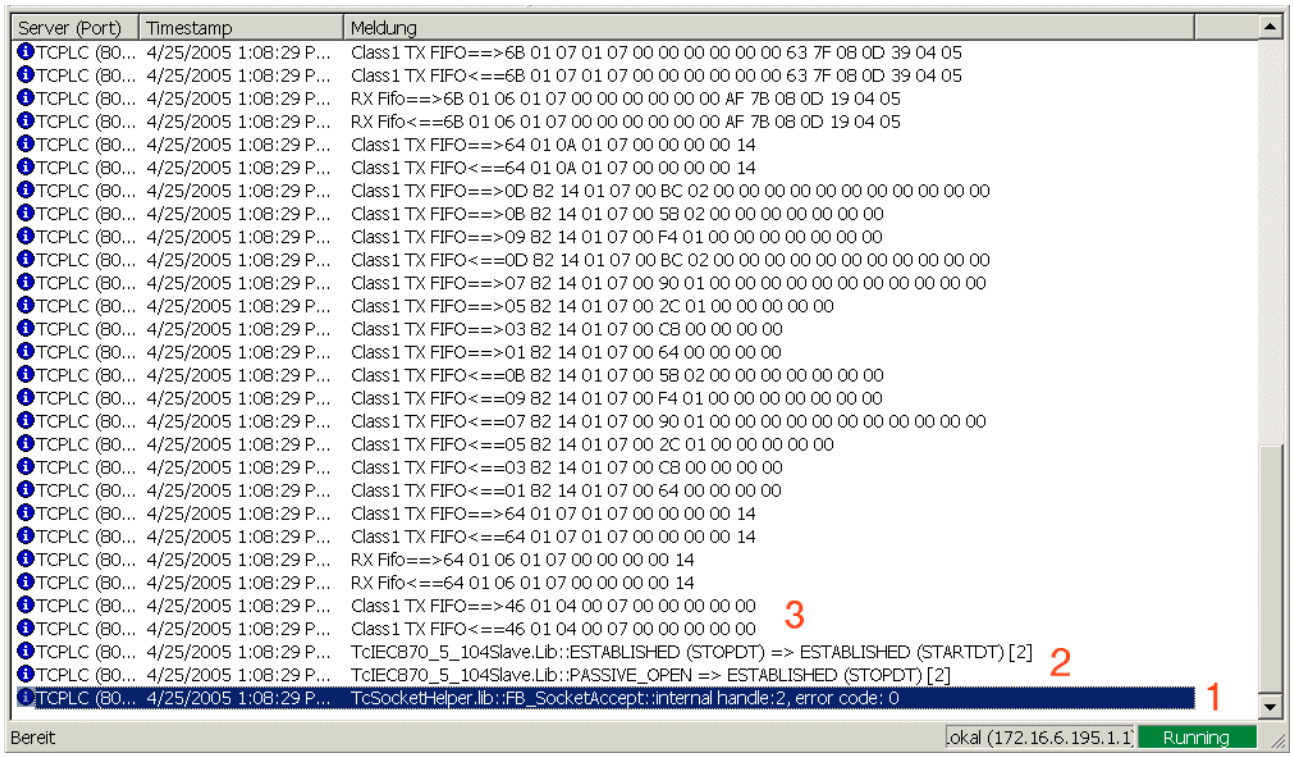


Fig. 1:

Further diagnostic tools are available:

TwinCAT ADS Monitor

Network monitor

Wireshark®

Ethereal

Diverse protocol test suite products

6.5.1.1.3.14 Glossary

Term	Description
substation, slave, server	Synonyms for a subordinate station (which is monitored)
Central station, control station, master, client	Synonyms for a higher-level station (control station, monitors others stations)
control direction	Data transfer direction from the central station to the substation
monitoring direction	Data transfer direction from the substation to the central station
application objects	IEC information objects in the TwinCAT PLC application (single points, double points, measured values, short floating point values etc.)
APDU	Protocol data unit of the application layer (application protocol data unit)
APCI	Protocol control information of the application layer (application protocol control information)
ASDU	Service data unit of the application layer (application service data unit)
IOA, address of the information object	Address of the single point, double point, etc. (information object address)

6.5.1.1.4 Control station tutorial (master)

A more complex central station application with different data points in both directions and a connection to the substation. The sample application uses the TwinCAT IEC 60870-5-104 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984633483.zip

In the sample project, the following station parameters are set for the central station:

- Server host address (IP address of the substation): **127.0.0.1** (You must at least adapt this parameter on your target platform!).
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Originator address: **1.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the substation):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_NA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200), M_DP_NA_1 (IOA := 201), M_DP_TB_1 (IOA := 202).

- Regulating step value: M_ST_NA_1 (IOA := 300), M_ST_NA_1 (IOA := 301), M_ST_TB_1 (IOA := 302).
- Bitstring value: M_BO_NA_1 (IOA := 400), M_BO_NA_1 (IOA := 401), M_BO_TB_1 (IOA := 402).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500), M_ME_NA_1 (IOA := 501), M_ME_TD_1 (IOA := 502).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600), M_ME_NB_1 (IOA := 601), M_ME_TD_1 (IOA := 602).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700), M_ME_NC_1 (IOA := 701), M_ME_TF_1 (IOA := 702).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_NA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10), C_SC_NA_1 (IOA := 11), C_SC_TA_1 (IOA := 12).
- Double command: C_DC_NA_1 (IOA := 20), C_DC_NA_1 (IOA := 21), C_DC_TA_1 (IOA := 22).
- Bitstring command: C_BO_NA_1 (IOA := 40), C_BO_NA_1 (IOA := 41), C_BO_TA_1 (IOA := 42).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50), C_SE_NA_1 (IOA := 51), C_SE_TA_1 (IOA := 52).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60), C_SE_NB_1 (IOA := 61), C_SE_TB_1 (IOA := 62).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70), C_SE_NC_1 (IOA := 71), C_SE_TC_1 (IOA := 72).

Initialization sequence that is executed once/automatically after the connection is established:

- No test command (C_TS_NA_1, deactivated).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- No counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, disabled). Counter interrogation Mode A (counters are stored locally in the substation and sent to the central station as spontaneous data).

Sample implementation of the following services:

- Test command (C_TS_NA_1) every 60 seconds.
- Clock synchronization command (C_CS_NA_1) every 60 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 60 seconds.
- No counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, disabled). Counter interrogation Mode A (counters are stored locally in the substation and sent to the central station as spontaneous data).
- Simulation and transmission of the commands every 10 seconds if the variable **bSendCmd** was set to TRUE (see upper data point configuration).
- Reception of spontaneous data.
- Reception of M_EI_NA_1 (end of initialization).

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol (slow task).
fbClient	FB_IEC870_5_104Master (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
hTable	T_HAODBTable (STRUCT)	Application object database handle. These variables are used to access the application object database and never directly the individual array elements.
F_iecCreateTableHnd	FUNCTION	In the initialization step, all array elements of AODB are initialized once with this function call. If successful, this function call returns the handle.
F_iecAddTableEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.
memory	ST_MemoryData (STRUCT)	This structure contains constantly updated process data – either received or to be sent.
fbHandler	FB_IEC870_5_101TableEventHandler (FB)	This function block informs the application when data point data has been changed or updated.

6.5.1.1.5 Scenarios

6.5.1.1.5.1 Spontaneous data communication and general query

The sample program presented here shows an IEC 60870-5-104 master with High Level interface and a simple communication of two data points. One data point is transferred spontaneously in control direction, the other in monitor direction. In addition, a general interrogation is configured for the data point in monitor direction.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243747211.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.1.1.5.2 General query and groups

The sample program presented here shows an IEC 60870-5-104 master with High Level interface. The sample illustrates the definition of groups of data types. A general interrogation and a separate group 1 are configured. The general interrogation takes place every 30 seconds, group 1 is interrogated every 10 seconds. The communicated parameters include single points, bitstrings and measured values.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243751819.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.1.1.5.3 Counting value query and time synchronization

The sample program presented here shows an IEC 60870-5-104 master with High Level interface. At the core of the sample is the time synchronization from the master to the slave and the counter interrogation. In the sample, the time is synchronized every 120 seconds. A data type M_IT_NA_1 is used to transfer the counting value via a general counter interrogation.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243756427.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8

- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.1.2 Low-Level Interface

6.5.1.2.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-104 Central station low-level sample [▶ 751].

A PLC application that is to communicate with a substation via the "low-level" interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_104TProtocol [▶ 514];
- An instance of the function block for establishing the TCP/IP connection: FB_ServerClientConnection or FB_ClientServerConnection
- An instance of the TX/RX data buffer: ST_IEC870_5_101TBuffer [▶ 583];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_101TBufferCtrl [▶ 506];

6.5.1.2.2 Control station (master)

Simple central station application that uses the TwinCAT IEC 60870-5-104 low-level interface. This sample application can only communicate with one substation.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984636427.zip

In the sample project, the following station parameters are set for the central station:

- Server host address (IP address of the substation): **127.0.0.1 (You must at least adapt this parameter on your target platform!).**
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**

- Information object address size: **3 octets.**
- Originator address: **0.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the substation):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_NA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200).
- Regulating step value: M_ST_NA_1 (IOA := 300).
- Bitstring value: M_BO_NA_1 (IOA := 400).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_NA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10).
- Double command: C_DC_NA_1 (IOA := 20).
- Regulating step command: C_RC_NA_1 (IOA := 30).
- Bitstring command: C_BO_NA_1 (IOA := 40).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70).

Sample implementation of the following services:

- Test command (C_TS_TA_1) every 10 seconds.
- Clock synchronization command (C_CS_NA_1) every 20 seconds.
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN) every 40 seconds.
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN) every 60 seconds. Counter interrogation Mode A.
- Simulation and transmission of the commands every 60 seconds if the variable **bSimulate** was set to TRUE (see upper data point configuration).
- Reception of spontaneous data.
- Reception of M_EI_NA_1 (end of initialization).
-

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol
fbClient	FB_IEC870_5_104M (FB)	Implements the central station. An instance of this function block can only communicate with one substation. For each additional substation to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_104MServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
stProcPara	ST_IEC870_5_104MProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_104MSysPara (STRUCT)	Configuration of the system parameters of the central station and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.5.1.2.3 Scenarios

Basic explanatory notes regarding the Low Level scenarios:

The initialization step begins with the configuration of the TCP/IP interface, during which the IP address and the communication port are set. Then, various protocol parameters and parameters of the send and receive buffers are set. This is followed by establishment of the TCP/IP connection with the aid of function blocks from the TCP/IP function.

Data are then received and loaded from the receive buffer with the action **fbBuffer.RxRemoveObj** until the buffer is empty. The received data are sorted by type ID and object address. After sorting, the respective data are stored in a variable with the function **F_iecCopyStreamToBuffer** and then split according to the components set out in the standard. If the command for a general interrogation or the interrogation command for a specific group of interrogations is received in the slave, the corresponding transfer procedure for interrogation commands is triggered immediately, and the data points defined for the respective group are written to the send buffer. Sending of data is described in more detail in the following paragraph.

Both in the master and in the slave this is followed by sending of the data to be transferred outside of or in addition to an interrogation group. First, common transfer details of the data points such as ASDU address, number of objects to be transferred, reason for the data transfer and priority class are written to the structure **stTxAsdu.ident**. Then the protocol frames of each data point, consisting of type ID, object address and object-specific data, are written to the structure **stTxAsdu**. This is followed by copying via the function **F_iecCopyBufferToStream** and writing to the send buffer via the action **fbBuffer.TxAddObj**. Finally, any errors are read from the error memory and listed in an error output.

The process of reading, writing and subsequent error handling is repeated cyclically.

6.5.1.2.3.1 Spontaneous data communication and general query

The sample program presented here shows an IEC 60870-5-104 master with Low Level interface and a simple communication of two data points. One data point is transferred spontaneously in control direction, the other in monitor direction. In addition, a general interrogation is configured for the data point in monitor direction.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244576139.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8

- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.1.2.3.2 General query and groups

The sample program presented here shows an IEC 60870-5-104 master with Low Level interface. The sample illustrates the definition of groups of data types. A general interrogation and a separate group 1 are configured. The general interrogation takes place every 30 seconds, group 1 is interrogated every 10 seconds. The communicated parameters include single points, bitstrings and measured values.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244593547.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

Important notes:

We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!

The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.1.2.3.3 Counting value query and time synchronization

The sample program presented here shows an IEC 60870-5-104 master with Low Level interface. At the core of the sample is the time synchronization from the master to the slave and the counter interrogation. In the sample, the time is synchronized every 120 seconds. A data type M_IT_NA_1 is used to transfer the counting value via a general counter interrogation.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244598155.zip

In the sample, the following parameters are set for the central station (master):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

Important notes:

We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!

The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2 Controlled station

6.5.2.1 High-Level Interface

6.5.2.1.1 Quick start

Simple project including complete sources is to be found here: IEC 60870-5-104 Controlled Station-High-Level example [▶ 756].

Interoperability list is to be found here: Interoperability check list [▶ 85]

Overview of error codes is to be found here: Error Error codes [▶ 805]

Brief instruction

1. Create a new PLC project and link the PLC library: **Tc2_IEC60870_5_10x**.
2. Apply an instance of the T_HSERVER variable (connection handle) at the main program and initialize it once at the program part by calling of the F_CreateServerHnd function with the fitting parameters.
3. Configuration of data points: Apply an array variable of ST_IEC870_5_101AODBEntry [▶ 584]. Every array element corresponds to a data point. Configure the data points via the function F_iecInitAOEntry [▶ 537] zat runtime (e.g. at an init step).

4. Apply an instance of the des protocol block FB_IEC870_5_104Slave [▶ 495] at the main program, configure it and call it.
   The system and protocol parameter are to be configured according to the parameter of the control station.

6.5.2.1.2 Substation Minimal (slave)

Simple substation application using the TwinCAT IEC 60870-5-104 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984639371.zip

In the sample project, the following station parameters are set for the substation:

- Server host address (IP address of the substation): **127.0.0.1 (You must at least adapt this parameter on your target platform!).**
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Originator address: **1.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100).
- Single command: C_SC_NA_1 (IOA := 10).

Sample implementation of the following services:

- End of Initialization (M_EI_NA_1) is sent once.
- Test command (C_TS_TA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_CI_NA_1) of the group: General (eIEC870_RQT_REQCOGEN). Counter interrogation Mode C.
- Single command processing (C_SC_NA_1) (direct command execution)
- Sending spontaneous data of the type (M_SP_NA_1, IOA := 100) when the value of the **bSPI** variable changes.

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol (slow task).
fbServer	FB_IEC870_5_104Slave (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
bSPI	BOOL	The ON/OFF value to be sent is mapped directly from the flag address **AT%MX1.5** of this allocated PLC variable.
bSCS	BOOL	The received ON/OFF value is mapped into this PLC variable directly to the allocated flag address: **AT%MX0.0**.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
F_iecInitAOEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.

6.5.2.1.3 Tutorial

6.5.2.1.3.1 Configuration of the programming environment

Proceed as follows for the basic configuration of the programming environment.

✓ Open Microsoft Visual Studio

1. Go to: File / New / Project
2. Select a TwinCAT project
3. Once the solution is open with the TwinCAT project, you can open the context menu by right-clicking on the PLC entry and select Add New Item
4. Select a standard PLC project
5. Expand the PLC project tree and select Add Library via the context menu of the References sub-point.
6. Add the following libraries as the basis for your IEC 60870-5-10x application.
   Tc2_Utilities
   Tc2_TcpIp
   Tc2_IEC60870_5_10x

⇨ You have thus completed the basic configuration of the programming environment.

6.5.2.1.3.2 Definition and configuration of application objects

Application objects = single points, double points, measured values, short floating point values, etc.

In this example the commands were configured such that the process data for the commands are located in the same memory area as the information data in monitoring direction, although in a different byte and bit offset. If required the commands may have the same byte/bit offset as the information in monitoring direction.

Example:

C_SC_NA_1 with IOA = 10 on the same byte/bit offset as M_SP_NA_1 with IOA = 100 (both byte offset = 100 and bit offset = 0). In this case a value change due to a command from the control station will result in a transmission of the M_SP_NA_1 with the object address 100 and cause of transmission <11> (returned by remote command).

The following application objects are taken as examples for the tutorial:

Array element	ASDU identifier	Object address IOA	Group configuration parameter	Base time multiplier	PLC process data area	Offset byte	Offset bit	Process data width in the TwinCAT PLC
0	M_SP_NA_1	100	General query	0	Flags	100	0	1 bit
1	M_SP_NA_1	101	General query	0	Flags	100	1	1 bit
2	M_SP_TB_1	102	General query	0	Flags	100	2	1 bit
3	M_DP_NA_1	200	General query	0	Flags	200	0	2 bits
4	M_DP_NA_1	201	General query	0	Flags	200	2	2 bits
5	M_DP_TB_1	202	General query	0	Flags	200	4	2 bits
6	M_ST_NA_1	300	General query	0	Flags	300	0	1 byte
7	M_ST_NA_1	301	General query	0	Flags	301	0	1 byte
8	M_ST_TB_1	302	General query	0	Flags	302	0	1 byte
9	M_BO_NA_1	400	General query	0	Flags	400	0	4 byte
10	M_BO_NA_1	401	General query	0	Flags	404	0	4 byte
11	M_BO_TB_1	402	General query	0	Flags	408	0	4 byte
12	M_ME_NA_1	500	General query	0	Flags	500	0	2 byte
13	M_ME_NA_1	501	General query	0	Flags	502	0	2 byte
14	M_ME_TD_1	502	General query	0	Flags	504	0	2 byte
15	M_ME_NB_1	600	General query	0	Flags	600	0	2 byte
16	M_ME_NB_1	601	General query	0	Flags	602	0	2 byte
17	M_ME_TE_1	602	General query	0	Flags	604	0	2 byte
18	M_ME_NC_1	700	General query	0	Flags	700	0	4 byte
19	M_ME_NC_1	701	General query	0	Flags	704	0	4 byte
20	M_ME_TF_1	702	General query	0	Flags	708	0	4 byte
21	M_IT_NA_1	800	General counter query and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	800	0	4 byte

22	M_IT_NA_1	801	General counter query and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	804	0	4 byte
23	M_IT_TB_1	802	General counter query and Mode A (local freeze with spontaneous transmission every 15 s)	0	Flags	808	0	4 byte
Commands								
24	C_SC_NA_1	10	-	0	Flags	2100	0	1 bit
25	C_SC_NA_1	11	-	0	Flags	2100	1	1 bit
26	C_SC_TA_1	12	-	0	Flags	2100	2	1 bit
27	C_DC_NA_1	20	-	0	Flags	2200	0	2 bit
28	C_DC_NA_1	21	-	0	Flags	2200	2	2 bit
29	C_DC_TA_1	22	-	0	Flags	2200	4	2 bit
30	C_RC_NA_1	30	-	0	Flags	2300	0	1 byte
31	C_RC_NA_1	31	-	0	Flags	2301	0	1 byte
32	C_RC_TA_1	32	-	0	Flags	2302	0	1 byte
33	C_BO_NA_1	40	-	0	Flags	2400	0	4 byte
34	C_BO_NA_1	41	-	0	Flags	2404	0	4 byte
35	C_BO_TA_1	42	-	0	Flags	2408	0	4 byte
36	C_SE_NA_1	50	-	0	Flags	2500	0	2 byte
37	C_SE_NA_1	51	-	0	Flags	2502	0	2 byte
38	C_SE_TA_1	52	-	0	Flags	2504	0	2 byte
39	C_SE_NB_1	60	-	0	Flags	2600	0	2 byte
40	C_SE_NB_1	61	-	0	Flags	2602	0	2 byte
41	C_SE_TB_1	62	-	0	Flags	2604	0	2 byte
42	C_SE_NC_1	70	-	0	Flags	2700	0	4 byte
43	C_SE_NC_1	71	-	0	Flags	2704	0	4 byte
44	C_SE_TC_1	72	-	0	Flags	2708	0	4 byte

Declaring a database variable

The application object database is an array variable of type ST_IEC870_5_101AODBEntry [▶ 584]. Each array element corresponds to an application object. The maximum number of application objects is freely selectable and is only limited by the available memory. During PLC programming you have to specify a constant maximum number. The maximum number of application objects cannot be changed at runtime.

In our example 50 application objects are declared. This number is sufficient for most applications. Please note that many application objects require adequate memory and runtime resources.

Define the following variable in MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry [▶ 584];
END_VAR
```

Configuring application objects

The object type (M_SP_NA_1, M_DP_NA_1, M_ST_NA_1 etc.), the object address and further object parameters are specified during configuration of the individual application objects.

The required application objects are configured during program runtime. Each application object (database array element) is configured by calling the F_iecInitAOEntry [▶ 537] function once. The array element to be configured is transferred to the function via VAR_IN_OUT. Configuration is usually carried out once during PLC program start-up via an Init routine. The F_iecInitAOEntry [▶ 537] function expects the following function parameters (from left to right):

```
FUNCTION F_iecInitAOEntry : UDINT
VAR_INPUT
eType : E_IEC870_5_101TcTypeID [▶ 601];
objAddr : DWORD := 0;
group : DWORD := 0;
multiplier : BYTE := 0;
ioMapType : E_IEC870_5_101IOMappingType [▶ 608];
byteOffs : UDINT := 0;
bitOffs : UDINT := 0;
END_VAR
VAR_IN_OUT
dbEntry : ST_IEC870_5_101AODBEntry;
END_VAR
```

eType: application object type (ASDU identifier, e.g.: M_SP_NA_1 for single point or M_DP_NA_1 for double point). Please note that only the ASDU types listed in the compatibility list can be used. Invalid types are ignored.

objAddr : object address, e.g. 100. Each application object should be configured with a unique address.

group: group configuration parameters. The available group parameters are defined as constants and can be combined with an OR operator. E.g.: IEC870_GRP_INROGEN OR IEC870_GRP_PERCYC.

A description of all group configuration parameters can be found here.

mutiplier: base time multiplier for cyclic/periodic data transfer (0=deactivated). The base time is configured via the system parameters. If the base time was set to T#10 s, and the multiplier to 2, for example, the periodic/cyclic data of the application object are sent every 20 seconds.

ioMapType: This parameter defines from or in which process data area of the TwinCAT PLC the IEC process data are to be mapped at runtime (inputs, outputs, memory, data).

byteOffs: process data area byte offset;

bitOffs: process data area bit offset;

dbEntry: application object to be configured (a database variable array element that is transferred to the function via VAR_IN_OUT).

In order to configure the application objects during program start-up, add the following PLC code in MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry [▶ 584];

init : BOOL := TRUE;
initError : UDINT;
END_VAR

IF init THEN
init := FALSE;

(* Monitored Single Points *)
initError := F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 0,
AODB[0] );
initError := F_iecInitAOEntry( M_SP_NA_1, 101, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 1,
AODB[1] );
initError := F_iecInitAOEntry( M_SP_TB_1, 102, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 100, 2,
AODB[2]);
(* Double Points*)
initError := F_iecInitAOEntry( M_DP_NA_1, 200, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 0,
AODB[3] );
initError := F_iecInitAOEntry( M_DP_NA_1, 201, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 2,
AODB[4] );
initError := F_iecInitAOEntry( M_DP_TB_1, 202, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 200, 4,
AODB[5] );
(* Regulating step value *)
initError := F_iecInitAOEntry( M_ST_NA_1, 300, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 300, 0,
AODB[6] );
initError := F_iecInitAOEntry( M_ST_NA_1, 301, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 301, 0,
AODB[7] );
initError := F_iecInitAOEntry( M_ST_TB_1, 302, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 302, 0,
AODB[8] );
(* 32 bit string*)
initError := F_iecInitAOEntry( M_BO_NA_1, 400, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 400, 0,
AODB[9] );
initError := F_iecInitAOEntry( M_BO_NA_1, 401, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 404, 0,
AODB[10] );
initError := F_iecInitAOEntry( M_BO_TB_1, 402, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 408, 0,
AODB[11] );
(* Measured value, normalized value *)
initError := F_iecInitAOEntry( M_ME_NA_1, 500, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 500, 0,
AODB[12] );
initError := F_iecInitAOEntry( M_ME_NA_1, 501, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 502, 0,
AODB[13] );
initError := F_iecInitAOEntry( M_ME_TD_1, 502, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 504, 0,
AODB[14] );
(* Mesured value, scaled value *)
initError := F_iecInitAOEntry( M_ME_NB_1, 600, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 600, 0,
AODB[15] );
initError := F_iecInitAOEntry( M_ME_NB_1, 601, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 602, 0,
AODB[16] );
initError := F_iecInitAOEntry( M_ME_TE_1, 602, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 604, 0,
AODB[17] );
(* Measured value , short floating point value *)
initError := F_iecInitAOEntry( M_ME_NC_1, 700, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 700, 0,
AODB[18] );
initError := F_iecInitAOEntry( M_ME_NC_1, 701, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 704, 0,
AODB[19] );
initError := F_iecInitAOEntry( M_ME_TF_1, 702, IEC870_GRP_INROGEN, 0, MAP_AREA_MEMORY, 708, 0,
AODB[20] );
(* Integrated totals *)
initError := F_iecInitAOEntry( M_IT_NA_1, 800, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0,
MAP_AREA_MEMORY, 800, 0, AODB[21] );
initError := F_iecInitAOEntry( M_IT_NA_1, 801, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0,
MAP_AREA_MEMORY, 804, 0, AODB[22] );
initError := F_iecInitAOEntry( M_IT_TB_1, 802, IEC870_GRP_REQCOGEN OR IEC870_GRP_LOCFREEZE, 0,
MAP_AREA_MEMORY, 808, 0, AODB[23] );

(* Single commands *)
initError := F_iecInitAOEntry( C_SC_NA_1, 10, 0, 0, MAP_AREA_MEMORY, 2100, 0, AODB[24] );
initError := F_iecInitAOEntry( C_SC_NA_1, 11, 0, 0, MAP_AREA_MEMORY, 2100, 1, AODB[25] );
initError := F_iecInitAOEntry( C_SC_TA_1, 12, 0, 0, MAP_AREA_MEMORY, 2100, 2, AODB[26] );
(* Double commands *)
initError := F_iecInitAOEntry( C_DC_NA_1, 20, 0, 0, MAP_AREA_MEMORY, 2200, 0, AODB[27] );
initError := F_iecInitAOEntry( C_DC_NA_1, 21, 0, 0, MAP_AREA_MEMORY, 2200, 2, AODB[28] );
initError := F_iecInitAOEntry( C_DC_TA_1, 22, 0, 0, MAP_AREA_MEMORY, 2200, 4, AODB[29] );
(* Regulating step commands *)
initError := F_iecInitAOEntry( C_RC_NA_1, 30, 0, 0, MAP_AREA_MEMORY, 2300, 0, AODB[30] );
initError := F_iecInitAOEntry( C_RC_NA_1, 31, 0, 0, MAP_AREA_MEMORY, 2301, 0, AODB[31] );
initError := F_iecInitAOEntry( C_RC_TA_1, 32, 0, 0, MAP_AREA_MEMORY, 2302, 0, AODB[32] );
```

```
(* 32 bit string commands *)
initError := F_iecInitAOEntry( C_BO_NA_1, 40, 0, 0, MAP_AREA_MEMORY, 2400, 0, AODB[33] );
initError := F_iecInitAOEntry( C_BO_NA_1, 41, 0, 0, MAP_AREA_MEMORY, 2404, 0, AODB[34] );
initError := F_iecInitAOEntry( C_BO_TA_1, 42, 0, 0, MAP_AREA_MEMORY, 2408, 0, AODB[35] );
(* Set point, normalized values*)
initError := F_iecInitAOEntry( C_SE_NA_1, 50, 0, 0, MAP_AREA_MEMORY, 2500, 0, AODB[36] );
initError := F_iecInitAOEntry( C_SE_NA_1, 51, 0, 0, MAP_AREA_MEMORY, 2502, 0, AODB[37] );
initError := F_iecInitAOEntry( C_SE_TA_1, 52, 0, 0, MAP_AREA_MEMORY, 2504, 0, AODB[38] );
(* Set point, scaled valuess *)
initError := F_iecInitAOEntry( C_SE_NB_1, 60, 0, 0, MAP_AREA_MEMORY, 2600, 0, AODB[39] );
initError := F_iecInitAOEntry( C_SE_NB_1, 61, 0, 0, MAP_AREA_MEMORY, 2602, 0, AODB[40] );
initError := F_iecInitAOEntry( C_SE_TB_1, 62, 0, 0, MAP_AREA_MEMORY, 2604, 0, AODB[41] );
(* Set point, short floating point values *)
initError := F_iecInitAOEntry( C_SE_NC_1, 70, 0, 0, MAP_AREA_MEMORY, 2700, 0, AODB[42] );
initError := F_iecInitAOEntry( C_SE_NC_1, 71, 0, 0, MAP_AREA_MEMORY, 2704, 0, AODB[43] );
initError := F_iecInitAOEntry( C_SE_TC_1, 72, 0, 0, MAP_AREA_MEMORY, 2708, 0, AODB[44] );

END_IF
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.3 Mapping the PLC and IEC process data

The TwinCAT PLC process data are cyclically mapped (copied) into the IEC process data (application objects) and vice versa at program runtime. Up to four process data ranges (IO inputs, IO outputs, flag range, data range) can be declared in the PLC program as buffer variables for the mapping of the IEC<->PLC process data. The byte size of the buffers is freely selectable and may be different for each area. Unused ranges need not necessarily be declared.

In our introductory example we declare 4 PLC process data areas with 3000 bytes each:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;

init : BOOL := TRUE;
initError : UDINT;

inputs AT%IB0 : ARRAY[0..2999] OF BYTE;
outputs AT%QB0 : ARRAY[0..2999] OF BYTE;
memory AT%MB0 : ARRAY[0..2999] OF BYTE;
data : ARRAY[0..2999] OF BYTE;

END_VAR
```

How the process data are to be mapped at runtime is specified during configuration of the application objects via the F_iecInitAOEntry function [▶ 537].
See also in: Definition and configuration of application objects [▶ 757].

The buffer variables were now declared as byte arrays. In order to improve access to the required data we define the individual variables a second time and allocate them to the corresponding byte/bit offset addresses. In case of a change in the byte array, the corresponding individual variable will be changed and inverted at the same time. However, this is not absolutely necessary. The bytes/bits of the byte array buffer variables can be accessed directly.

```
VAR_GLOBAL(* Memory offset 0..99 unused *)
(* Single points *)
msgSingle_0 AT%MX100.0 : BOOL;
msgSingle_1 AT%MX100.1 : BOOL;
msgSingle_2 AT%MX100.2 : BOOL;

(* Double points *)
(* Bit 0..1 = first double point,
Bit 2..3 = second double point,
Bit 4..5 = third double point,
Bit 6..7 = fourth double point *)
msgDouble_0 AT%MB200 : BYTE;

(* Regulating step values *)
msgStep_0 AT%MB300 : BYTE;
msgStep_1 AT%MB301 : BYTE;
msgStep_2 AT%MB302 : BYTE;

(* 32 bit strings *)
```

```
msgBitStr_0 AT%MD400 : DWORD := 2#10001000_10001000_10001000_10001000;
msgBitStr_1 AT%MD404 : DWORD := 2#10001000_10001000_10001000_10001000;
msgBitStr_2 AT%MD408 : DWORD := 2#10001000_10001000_10001000_10001000;

(* Measured values, normalized values *)
msgNormalized_0 AT%MW500 : WORD;
msgNormalized_1 AT%MW502 : WORD;
msgNormalized_2 AT%MW504 : WORD;

(* Measured values, scaled values *)
msgScaled_0 AT%MW600 : INT;
msgScaled_1 AT%MW602 : INT;
msgScaled_2 AT%MW604 : INT;

(* Measured values, short floating point values *)
msgFloating_0 AT%MD700 : REAL;
msgFloating_1 AT%MD704 : REAL;
msgFloating_2 AT%MD708 : REAL;

(* Integrated totals *)
msgTotal_0 AT%MD800 : UDINT;
msgTotal_1 AT%MD804 : UDINT;
msgTotal_2 AT%MD808 : UDINT;

(*####################################################################*)
(* Single commands *)
cmdSingle_0 AT%MX2100.0 : BOOL;
cmdSingle_1 AT%MX2100.1 : BOOL;
cmdSingle_2 AT%MX2100.2 : BOOL;

(* Double commands *)
(* Bit 0..1 = first double command,
Bit 2..3 = second double command,
Bit 4..5 = third double command,
Bit 6..7 = fourth double command *)
cmdDouble_0 AT%MB2200 : BYTE;

(* Regulating step commands *)
cmdStep_0 AT%MB2300 : BYTE;
cmdStep_1 AT%MB2301 : BYTE;
cmdStep_2 AT%MB2302 : BYTE;

(* 32 bit string commands *)
cmdBitStr_0 AT%MD2400 : DWORD;
cmdBitStr_1 AT%MD2404 : DWORD;
cmdBitStr_2 AT%MD2408 : DWORD;

(* Set point, normalized values *)
cmdNormalized_0 AT%MW2500 : WORD;
cmdNormalized_1 AT%MW2502 : WORD;
cmdNormalized_2 AT%MW2504 : WORD;

(* Set point, scaled values *)
cmdScaled_0 AT%MW2600 : INT;
cmdScaled_1 AT%MW2602 : INT;
cmdScaled_2 AT%MW2604 : INT;

(* Set point, short floating point values *)
cmdFloating_0 AT%MD2700 : REAL;
cmdFloating_1 AT%MD2704 : REAL;
cmdFloating_2 AT%MD2708 : REAL;
END_VAR
```

Mapping of the IEC<->PLC process data in the substation

Process data in the monitoring direction (Slave->Master information)

Example 1

Single point information (M_SP_NA_1) with the IOA 100, PLC flag range, byte offset = 100, bit offset = 0.

msgSingle_0 == memory[100].0 -> Substation FB -> ... -> Control Station

Example 2

Measured value, short floating point value (M_ME_NC_1) with the IOA 700, PLC flag range, byte offset = 700, bit offset = 0 (no meaning).

msgFloating_0 == memory[700..703] -> Substation FB -> ... -> Control Station

Process data in the control direction (Master->Slave commands)

Example 1

Single command state (C_SC_NA_1) with the IOA 10, PLC flag range, byte offset = 2100, bit offset = 0.

Control station -> ... -> Substation FB -> memory[2100].0 == cmdSingle_0

Example 2

Set point, short floating point value (C_SE_NC_1) with the IOA 70, PLC flag range, byte offset = 2700, bit offset = 0 (no meaning).

Control station -> ... -> Substation FB -> memory[2700 to 703] == cmdFloating_0

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.4 Call and declaration of the substation

The complete functionality of a substation is encapsulated in the FB_IEC870_5_104Slave function block. An instance can be used to establish a connection to the master. For a further connection, declare a further instance and transmit the same server handle (*hServer* variable) to this second instance, or use the FB_IEC870_5-104SlaveGrp [▶ 498] function block (recommended). You must set the IP address to match the IP address of your target system

Insert the following PLC code in the declaration part of MAIN:

```
PROGRAM MAIN
VAR
AODB : ARRAY[0..49] OF ST_IEC870_5_101AODBEntry;

init : BOOL := TRUE;
initError : UDINT;

inputs AT%IB0 : ARRAY[0..2999] OF BYTE;
outputs AT%QB0 : ARRAY[0..2999] OF BYTE;
memory AT%MB0 : ARRAY[0..2999] OF BYTE;
data : ARRAY[0..2999] OF BYTE;

bEnable : BOOL := TRUE;
hServer : T_HSERVER;
server : FB_IEC870_5_104Slave [▶ 495];

END_VAR
```

and the instance is called in the program part:

```
IF init THEN
init := FALSE;
...


F_CreateServerHnd( '', '127.0.0.1'(* change this! *), 2404, nMode := LISTEN_MODE_CLOSEALL OR
CONNECT_MODE_ENABLEDBG, bEnable, hServer );

ELSE
server( pInputs := ADR( inputs ),
cbInputs := SIZEOF( inputs ),
pOutputs := ADR( outputs ),
cbOutputs := SIZEOF( outputs ),
pMemory := ADR( memory ),
cbMemory := SIZEOF( memory ),
pData := ADR( data ),
cbData := SIZEOF( data ),
pAOEntries := ADR( AODB ),
cbAOEntries := SIZEOF( AODB ),
hServer := hServer,
bEnable := bEnable );
END_IF
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.5 Basic settings: Protocol parameters

The behavior of the substation can be adapted to the requirements of the master via the IEC60870-5-104 protocol parameters. Most parameters have preallocated default values that do not have to be changed.

The values of the iK and iW parameters are changed in our example:

```
IF init THEN
init := FALSE;
...
F_CreateServerHnd( '', '127.0.0.1'(* change this! *), 2404, nMode := LISTEN_MODE_CLOSEALL OR
CONNECT_MODE_ENABLEDBG, bEnable, hServer);

server.protPara.iK := 12;
server.protPara.iW := 8;

ELSE
server( pInputs := ADR( inputs ),
cbInputs := SIZEOF( inputs ),
pOutputs := ADR( outputs ),
...
END_IF
```

The documentation for all transfer protocol parameters can be found here: ST_IEC870_5_104PotocolParams [▶ 651].

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.6 Basic settings: System parameters

The common ASDU address and the user functions of the substation are configured via the system parameters.

In our introduction we configure the following system parameters:

The common ASDU address is set to 7 (*asduAddr*).

During initialization the system time of the substation is synchronized with the system time of the local TwinCAT PC (*bUsePCTime*).

Synchronization of the substation system time via the time synchronization command is activated (*bSyncTime*).

During the synchronization of the system time in the substation, the system time of the TwinCAT PC should not be synchronized (*bSyncPCTime*).

Sending of M_EI_NA_1 (End of init) to the central station is activated (*bEndOfInit*).

Sending of the periodic/cyclic data is deactivated (*bPerCyclic*). The base time for sending of these data is set to 5 s.

Background scan is deactivated (*bBackScan*). The background scan cycle time is set to 30 s.

Local freeze and resetting of the counter readings is activated (*bPerFRZ*), and the cycle time for freeze and resetting is set to 15 s.

The logging of the debug messages in the application log is activated (*dbgMode*). Changes in the device status are logged.

Add the following PLC code to your PLC project:

```
IF init THEN
init := FALSE;
...

server.sysPara.asduAddr := 7;
server.sysPara.bUsePCTime := TRUE;
server.sysPara.bSyncTime := TRUE;
server.sysPara.bSyncPCTime := FALSE;
```

```
server.sysPara.bEndOfInit := TRUE;
server.sysPara.bPerCyclic := FALSE;
server.sysPara.tPerCyclicBase := T#5s;
server.sysPara.bBackScan := FALSE;
server.sysPara.tBackScanCycle := T#30s;
server.sysPara.bPerFRZ := TRUE;
server.sysPara.tPerFRZCycle := T#15s;
server.sysPara.dbgMode := (*IEC870_DEBUGMODE_ASDU OR*) IEC870_DEBUGMODE_DEVSTATE;

...
ELSE
server( pInputs := ADR( inputs ),
cbInputs := SIZEOF( inputs ),
pOutputs := ADR( outputs ),
...

END_IF
```

The documentation for all system parameters can be found here: ST_IEC870_5_101SystemParams [▶ 588].

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.7 Station query / general query

The station interrogation command is initiated by the central station. The ID field of the command also contains the group (1 to 16 or general). The substation transfers the application objects associated with this group to the central station with cause of transmission <*20*> to <*36*>. Application objects with time tags are transferred without time tags.

Configuration of the system parameters

No special system parameters need to be set;

Configuration of the application objects

The data point must be assigned to one or more groups. The group parameter must be set. An overview of all available groups can be found here: Group configuration flags [▶ 655].

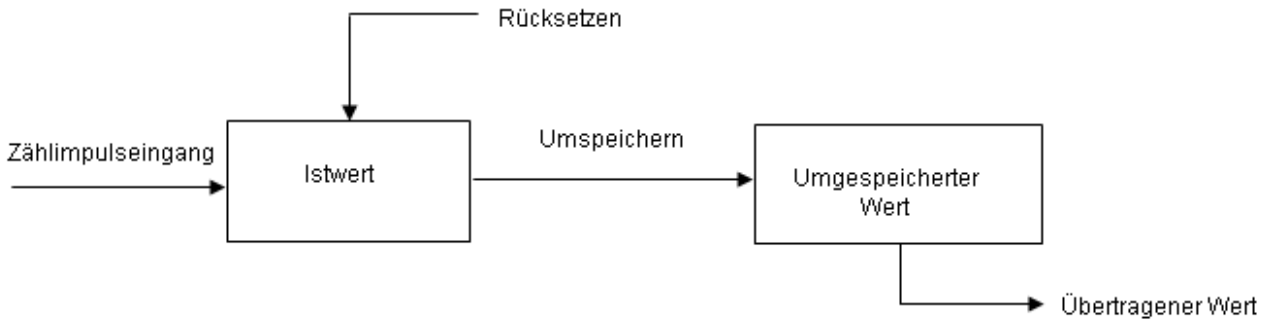
Example configuration for a data point that was assigned to Group: 1 and Group: General.

```
F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN OR IEC870_GRP_INRO1, 0, MAP_AREA_MEMORY, 100,
0, AODB[0] );
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.8 Count transfer

For explanation purposes, here is a general model for the transmission of integrated totals:



The actual values are added by counters. Via a re-store command that is either received by the central station or generated locally (in the substation), the actual values can be re-stored (copied) periodically into re-stored values. After freezing, the recorded value is either reset to zero (logging of incremental values), or the counter continues adding up (logging of counter readings).

Application objects with counts are assigned to groups. The groups are frozen individually, reset, or transferred. The central station sends count query commands to the substation. The task to be carried out (FRZ) and the group (RQT) are specified in an ID field of the command (QCC).

The allocation of the application objects to the individual groups (1 to 4 or general) is specified via the group flag parameter during configuration. There are four operating modes for recording counter readings and incremental values. Each mode includes notes about the configuration of the system parameters or the application objects.

Mode A: local freeze with spontaneous transfer

The substation internally initiates freeze or freeze with reset. The frozen counts are transferred spontaneously, once the freeze or freeze with reset function was executed. In this mode the central station does not issue any count query commands.

Configuration of the system parameters:

```
bPerFRZ := TRUE
tPerFRZCycle := T#60s
```

The first parameter activates local freeze or freeze with reset. The second parameter specifies the cycle time with which freeze or freeze with reset is carried out (e.g. every 60 seconds).

Configuration of the application objects:

The IEC870_GRP_SPONTOFF group parameter must not be set. This would prevent the spontaneous transmission of the counts.

The count is frozen if the IEC870_GRP_LOCFREEZE group parameter was set.

The count is reset if the IEC870_GRP_LOCRESET group parameter was set.

Local freeze or freeze with reset is carried out simultaneously for all groups (1 to 4 or general).

Operating mode B: Local freeze with counter interrogation

The substation internally initiates freeze or freeze with reset. The central station queries the frozen counts via count query commands. In this case the central station must not use freeze or freeze with reset in the command ID field (FRZ=0). The counts are queried generally or in groups 1 to 4.

Configuration of the system parameters:

```
bPerFRZ := TRUE
tPerFRZCycle := T#60s
```

The first parameter activates local freeze or (and) reset. The second parameter specifies the cycle time with which freeze or freeze with reset is carried out (e.g. every 60 seconds).

Configuration of the application objects:

The IEC870_GRP_SPONTOFF group parameter must be set. The counts are not to be transferred spontaneously to the central station.

The count is frozen if the IEC870_GRP_LOCFREEZE group parameter was set.

The count is reset if the IEC870_GRP_LOCRESET group parameter was set.

Local freeze or freeze with reset is carried out simultaneously for all groups (1 to 4 or general).

Operating mode C: The central station initiates freeze, freeze with reset, or reset

The central station periodically issues a count query command to the substation for controlling the freeze or (and) reset process. This command does not result in a count transfer. The central station sends a subsequent count query command for collecting the frozen counts. This is similar to operating mode B.

Configuration of the system parameters:

```
bPerFRZ := FALSE
tPerFRZCycle := T#60s
```

Local freeze or (and) reset must be deactivated. The second parameter is ignored.

Configuration of the application objects:

IEC870_GRP_SPONTOFF must be set. The counts are not to be transferred spontaneously to the central station.

The IEC870_GRP_LOCFREEZE and IEC870_GRP_LOCRESET group parameters must not be set. The central station initiates freeze or (and) reset.

The counts can be assigned to individual groups (1 to 4 or general) and queried (IEC870_GRP_REQCOGEN, IEC870_GRP_REQCO1, IEC870_GRP_REQCO2, IEC870_GRP_REQCO3, IEC870_GRP_REQCO4).

Operating mode D: The central station initiates freeze and (or) reset, and the frozen values are transferred spontaneously

This mode is a combination of the count command from the central station as in mode C and spontaneous transfer of the counts as in mode A.

Configuration of the system parameters:

```
bPerFRZ := FALSE
tPerFRZCycle := T#60s
```

Local freeze or (and) reset must be deactivated. The second parameter is ignored.

Configuration of the application objects:

The IEC870_GRP_SPONTOFF group parameter must not be set. This would prevent the spontaneous transmission of the counts.

The IEC870_GRP_LOCFREEZE and IEC870_GRP_LOCRESET group parameters must not be set. The central station initiates freeze or (and) reset.

The counts can be assigned to individual groups (1 to 4 or general) and queried (IEC870_GRP_REQCOGEN, IEC870_GRP_REQCO1, IEC870_GRP_REQCO2, IEC870_GRP_REQCO3, IEC870_GRP_REQCO4).

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.9 Clock synchronization

The substation can be synchronized with the time of the central station. The settings necessary for this are made in the System Parameters:

```
     fbSlave.sysPara.bSyncTime := TRUE; // sync IEC system time after reception of C_CS_NA_1
commmando
     fbSlave.sysPara.bSyncPCTime := TRUE; // sync local PC system time from master
     fbSlave.sysPara.tSyncToPCTime := T#30S;
```

A detailed description:

bSyncTime: if TRUE, activate IEC system time synchronization on receiving the C_CS_NA_1 command. Default: TRUE.

bSyncPCTime: If TRUE, synchronize the IEC system time and the system time of the TwinCAT PC (the Windows system time in the taskbar). Default: FALSE.

bUsePCTime: If TRUE, synchronize the IEC system time with the system time of the PC. Following the initialization of the substation, the IEC system time is first synchronized with the system time of the TwinCAT PC (Windows system time). Default: TRUE.

tSyncToPCTime: controls the cyclic synchronization of the IEC system time with the system time of the TwinCAT PC (in online and offline mode). The time is determined by the cycle time in which the synchronization is executed. A value of T#0s deactivates the cyclic synchronization. Default: T#0s.

sPCTimeNetID: TwinCAT network address of the PC whose system time is to be used for the synchronization. If it is to be run on the local computer, an empty string can be entered. Default: empty string = local PC.

bTimeCOT3: If TRUE, send the system time to the central station with the cause of transmission <*3*> *spontaneous* on the hour. Default: FALSE

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.10 Background scan:

The background scan is used for refreshing the process information sent from the substation to the central station as an additional safety contribution to the station interrogation and for spontaneous transfers. Application objects with the same type IDs as for the station interrogation may be transferred continuously with low priority, and with <2> background scan as the cause of transmission. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transmission). The background scan is initiated by the substation and is independent of the station interrogation commands.

Configuration of the system parameters

The transfer cycle is specified via system parameters [▶ 588] in the substation.

```
bBackScan := TRUE;
tBackScanCycle := T#30s;
```

Configuration of the application objects

Application objects whose process data are to be transferred as a background scan have to be configured via the IEC870_GRP_BACKGROUND group flag.

Example:

```
F_iecInitAOEntry( M_SP_NA_1, 100, IEC870_GRP_INROGEN OR IEC870_GRP_BACKGROUND, 0, MAP_AREA_MEMORY,
100, 0, AODB[0] );
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.11 Cyclic data transmission

Cyclic data transfer is initiated in a similar way as the background scan from the substation. It is independent of other commands from the central station. Cyclic data transfer continuously refreshes the process data of the central station. The process data are usually measured values that are recorded at regular intervals. Cyclic data transfer is often used for monitoring non-time-critical or relatively slowly changing process data (e.g. temperature sensor data). Cyclic/periodic data are transferred to the central station with cause of transmission <*1*> *periodic/cyclic*. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transmission). Cyclic data transfer can be configured via the system parameters and the configuration parameters of the application objects.

Configuration of the system parameters:

```
bPerCyclic : BOOL := TRUE;
tPerCyclicBase : TIME := T#60s;
```

The first parameter activates cyclic transfer. The second parameter is the base time for the cyclic/periodic data transfer (in this case 60 seconds).

Configuration of the application objects:

IEC870_GRP_PERCYC group parameter has to be set;

The multiplier parameter *(multiplier)* of the F_iecInitAOEntry function has to be set to a zero <> value. Example: With a multiplier = 2 and a base time of 60 seconds the process data of the application object are sent to the central station every 120 seconds;

Example configuration for a measured value that is to be transmitted cyclically every 120 seconds to the central station (measured value, normalized value without time tag, M_ME_NA_1).

```
F_iecInitAOEntry( M_ME_NA_1, 222, IEC870_GRP_INROGEN OR IEC870_GRP_PERCYC, 2, MAP_AREA_MEMORY, 6, 0,
AODB[2] );
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.12 Command transmission

Commands can be sent from the central station in control direction (to the substation).
A single command with type ID 45 (C_SC_NA_1) is used for controlling an application object that is transferred in monitoring direction as a single-point information (M_SP_NA_1, M_SP_TA_1 or M_SP_TB_1).
A dual command (C_DC_NA_1) is used for controlling an application object that is transferred in monitoring direction as a double-point information (M_DP_NA1, M_DP_TA_1 or M_DP_TB_1), etc.

Configuration of the system parameters:

No special system parameters need to be set;

Configuration of the application objects:

The application objects must be configured as commands (data types in the control direction);

The addresses of the information objects (IOAs) must correspond to the addresses in the control station;

Examples:

Single command with the IOA 10. The value received is copied into the flag range buffer, byte offset = 100, bit offset = 0.

```
F_iecInitAOEntry( C_SC_NA_1, 10, 0, 0, MAP_AREA_MEMORY, 100, 0, AODB[24] );
```

Double command with the IOA 20. The value received is copied into the flag range buffer, byte offset = 200, bit offset = 0 to 1.

```
F_iecInitAOEntry( C_DC_NA_1, 20, 0, 0, MAP_AREA_MEMORY, 200, 0, AODB[27] );
```

Set point, scaled value with the IOA 60. The value received is copied into the flag range buffer, byte offset = 600 to 601, bit offset = 0.

```
F_iecInitAOEntry( C_SE_NB_1, 60, 0, 0, MAP_AREA_MEMORY, 600, 0, AODB[39] )
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.13 Interrogation/read procedure

The central station sends interrogation commands to the substation.
The interrogation command contains the address of the application object to be interrogated. The data of this application object are to be sent to the central station. The substation sends the data with cause of transmission *<5> interrogation or interrogated*. The valid ASDU type IDs are listed in the compatibility list for the station (table type ID <-> cause of transmission).

Configuration of the system parameters:
No special system parameters need to be set.

Configuration of the application objects:
No special parameters need to be set.

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.14 Double transmission

All application objects (information objects) that are transferred with cause of transmission *<3> spontaneous* may be transferred twice, with or without time tag. This mode is referred to as "double transmission".
Double transmission is currently not supported by the substation.

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.15 Quality Flags

The quality flags (quality descriptor) provide additional information for the central station on the quality of an application object. The quality flags can be set/reset independent of each other from the PLC application via the F_iecSetAOQuality [▶ 539] function. The F_iecGetAOQuality [▶ 540] function can be used to interrogate the state of the quality flags. Any change in the quality flags leads to a spontaneous transfer of the data to the central station.

The following quality flags are internally analyzed by the substation at runtime:

IECQ_BL_ON (blocked). If the process data of the application object were blocked for the transfer, mapping of the PLC and IEC process data is not executed for this application;

The following quality flags are internally set/reset by the substation at runtime:

IECQ_IV_ON (Invalid). The substation sets the invalid flag if mapping of the PLC and IEC process data could not be carried out (e.g. due to faulty configuration of the application object). This behavior can be deactivated by setting group parameter IEC870_GRP_IV_OFF.

All other quality flags are sent to the central station without change.

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.16 Test of the communication

A simple simulation of the data points in the monitoring direction is activated by setting the *bChangeIO* variable to TRUE and deactivated by setting it to FALSE. If a connection is active, the values are transferred cyclically to the control station every 3 seconds.

```
PROGRAM MAIN
VAR
...

bChangeIO : BOOL; (* TRUE => simulate/modify plc process data *)
timer : TON;
i : INT;

...
END_VAR

...


(*modify plc process data *)
timer( IN := bChangeIO, PT := T#3s );
IF timer.Q THEN
timer( IN := FALSE );

msgSingle_0 := NOT msgSingle_0;
msgSingle_1 := NOT msgSingle_1;
msgSingle_2 := NOT msgSingle_2;

FOR i:= 0 TO 3 DO
IF F_iecGetDPI(msgDouble_0, i) = eIEC870_DPI_ON THEN (* the value of double point allready ON? *)
msgDouble_0 := F_iecSetDPI( msgDouble_0, i, eIEC870_DPI_OFF ); (* change ON => OFF *)
ELSE
msgDouble_0 := F_iecSetDPI( msgDouble_0, i, eIEC870_DPI_ON );(* change OFF => ON *)
END_IF
END_FOR

F_iecIncVTI( msgStep_0 );
F_iecDecVTI( msgStep_1 );

msgBitStr_0 := ROL( msgBitStr_0, 1 );
msgBitStr_1 := ROR( msgBitStr_1, 1 );

msgNormalized_0 := msgNormalized_0 + 1;
msgNormalized_1 := msgNormalized_1 + 2;

msgScaled_0 := msgScaled_0 + 3;
```

```
msgScaled_1 := msgScaled_1 - 3;

msgFloating_0 := msgFloating_0 + 0.1;
msgFloating_1 := msgFloating_1 + 1.5;

msgTotal_0 := msgTotal_0 + 1;
msgTotal_1 := msgTotal_1 + 2;
END_IF


...
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.17 Protocol and station errors

The station error messages are stored in a FIFO memory. Up to 10 error messages can be buffered. In the case of fatal communication errors (e.g. errors of the connection layer, the checksum of the frame is wrong) the connection is interrupted and must be re-established. Errors in the application layer (e.g. the ASDU transmit buffer has overflowed due to too many frames) are only logged and do not lead to interruption of the connection. Even with these errors it is still possible to interrupt the connection from the application. Apart from the error code, the error source is also stored in the error message. This makes it easier to locate the error.

Example

The accumulated error messages of an IEC 60870-5-104 substation can be read out using the following call:

```
PROGRAM MAIN
VAR
...

server : FB_IEC870_5_104Slave;

...
END_VAR

....


REPEAT
server.system.device.errors.RemoveError();
IF server.system.device.errors.bOk THEN
ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG,
'IEC60870-5-104 slave error: 0x%s',
DWORD_TO_HEXSTR( server.system.device.errors.getError.nErrId, 8, FALSE) );
END_IF
UNTIL NOT server.system.device.errors.bOk
END_REPEAT

...
```

You can download the corresponding PLC tutorial example from the Substation Tutorial (Slave) [▶ 774] page.

6.5.2.1.3.18 Extended diagnosis

Debugging messages written to the application log facilitate troubleshooting of the system. 5 stages of the debug messages can presently be activated in an IEC application. Four stages are activated by the dbgMode system parameters of the substation (ST_IEC870_5_101SystemParams [▶ 588]).

Debugging messages that are logged when the TCP/IP connection is established or released (TcSocketHelper.Lib messages). These messages can be activated by the nMode parameter: CONNECT_MODE_ENABLEDBG when calling the F_CreateServerHnd() function;

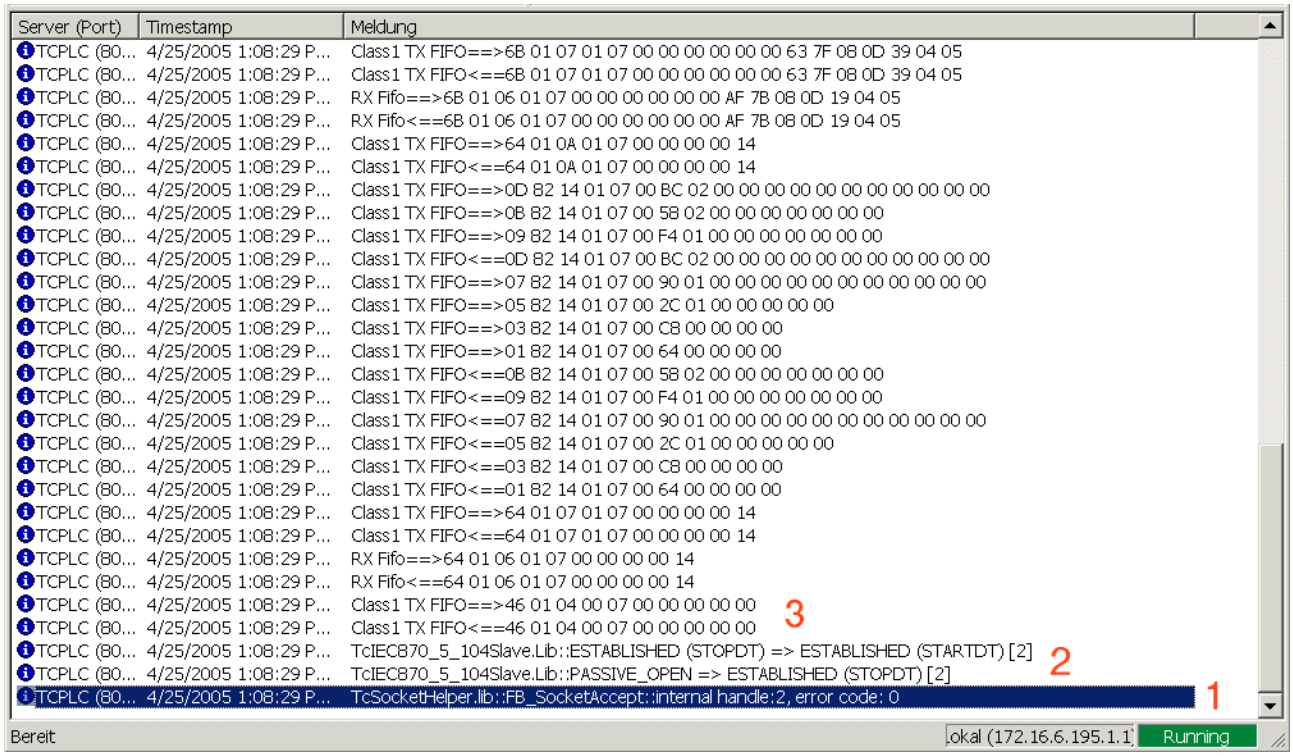
Station status messages (dbgMode: IEC870_DEBUGMODE_DEVSTATE);

Hexadecimal output of the ASDUs (dbgMode: IEC870_DEBUGMODE_ASDU). 32 ASDU data bytes per row are output as hexadecimal numbers. Longer ASDUs are spread across several rows;

Hexadecimal output of the APDUs (TCP/IP telegrams, dbgMode: IEC870_DEBUGMODE_LINKLAYER). 32 APDU data bytes per row are output as hexadecimal numbers. Like in 3., longer APDUs are spread across several rows;

Output of the link layer errors as messages (dbgMode: IEC870_DEBUGMODE_LINKERROR);

In order to view the activated debug messages start the TwinCAT System Manager and activate log view. A debugging output is shown below. The first three different types of message are marked with corresponding numbers.



6.5.2.1.3.19 Glossary

Term	Description
substation, slave, server	Synonyms for a subordinate station (which is monitored)
Central station, control station, master, client	Synonyms for a higher-level station (control station, monitors others stations)
control direction	Data transfer direction from the central station to the substation
monitoring direction	Data transfer direction from the substation to the central station
application objects	IEC information objects in the TwinCAT PLC application (single points, double points, measured values, short floating point values etc.)
APDU	Protocol data unit of the application layer (application protocol data unit)
APCI	Protocol control information of the application layer (application protocol control information)
ASDU	Service data unit of the application layer (application service data unit)
IOA, address of the information object	Address of the single point, double point, etc. (information object address)

6.5.2.1.4 Controlled station tutorial (slave)

A more complex substation application with different data points in both directions and a connection to the central station. The sample application uses the TwinCAT IEC 60870-5-104 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984642315.zip

In the sample project, the following station parameters are set for the substation:

- Server host address (IP address of the substation): **127.0.0.1 (You must at least adapt this parameter on your target platform!).**
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**
- Information object address size: **3 octets.**
- Originator address: **1.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_NA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200), M_DP_NA_1 (IOA := 201), M_DP_TB_1 (IOA := 202).
- Regulating step value: M_ST_NA_1 (IOA := 300), M_ST_NA_1 (IOA := 301), M_ST_TB_1 (IOA := 302).
- Bitstring value: M_BO_NA_1 (IOA := 400), M_BO_NA_1 (IOA := 401), M_BO_TB_1 (IOA := 402).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500), M_ME_NA_1 (IOA := 501), M_ME_TD_1 (IOA := 502).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600), M_ME_NB_1 (IOA := 601), M_ME_TD_1 (IOA := 602).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700), M_ME_NC_1 (IOA := 701), M_ME_TF_1 (IOA := 702).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_NA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10), C_SC_NA_1 (IOA := 11), C_SC_TA_1 (IOA := 12).
- Double command: C_DC_NA_1 (IOA := 20), C_DC_NA_1 (IOA := 21), C_DC_TA_1 (IOA := 22).
- Bitstring command: C_BO_NA_1 (IOA := 40), C_BO_NA_1 (IOA := 41), C_BO_TA_1 (IOA := 42).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50), C_SE_NA_1 (IOA := 51), C_SE_TA_1 (IOA := 52).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60), C_SE_NB_1 (IOA := 61), C_SE_TB_1 (IOA := 62).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70), C_SE_NC_1 (IOA := 71), C_SE_TC_1 (IOA := 72).

Sample implementation of the following services:

- End of initialization (M_EI_NA_1) is sent once.

- Test command (C_TS_TA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, Mode A). Counters are stored locally in the substation every 15 seconds and transmitted as spontaneous data to the central station.
- Command execution of some commands (direct commands, see upper data point configuration).
- Simulation and transmission of spontaneous data every 10 seconds if the variable **bSendSpont** was set to TRUE (see upper data point configuration).

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol.
hServer	T_HSERVER (STRUCT)	The server connection handle.
fbServer	FB_IEC870_5_104Slave (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
F_CreateServerHnd	FUNCTION	This function initializes the server connection handle.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
F_iecInitAOEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.
memory	ST_MemoryData (STRUCT)	This structure contains constantly updated process data – either received or to be sent.
F_iecSetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be set/reset.
F_iecGetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be queried.

6.5.2.1.5 Controlled station connection redundancy (slave)

A more complex substation application with different data points in both directions and two connections to the central station. The sample application uses the TwinCAT IEC 60870-5-104 high-level interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984645259.zip

In the sample project, the following station parameters are set for the substation:

- Server host address (IP address of the substation): **127.0.0.1 (You must at least adapt this parameter on your target platform!).**
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**

- Information object address size: **3 octets.**
- Originator address: **1.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_NA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200), M_DP_NA_1 (IOA := 201), M_DP_TB_1 (IOA := 202).
- Regulating step value: M_ST_NA_1 (IOA := 300), M_ST_NA_1 (IOA := 301), M_ST_TB_1 (IOA := 302).
- Bitstring value: M_BO_NA_1 (IOA := 400), M_BO_NA_1 (IOA := 401), M_BO_TB_1 (IOA := 402).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500), M_ME_NA_1 (IOA := 501), M_ME_TD_1 (IOA := 502).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600), M_ME_NB_1 (IOA := 601), M_ME_TD_1 (IOA := 602).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700), M_ME_NC_1 (IOA := 701), M_ME_TF_1 (IOA := 702).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_NA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10), C_SC_NA_1 (IOA := 11), C_SC_TA_1 (IOA := 12).
- Double command: C_DC_NA_1 (IOA := 20), C_DC_NA_1 (IOA := 21), C_DC_TA_1 (IOA := 22).
- Bitstring command: C_BO_NA_1 (IOA := 40), C_BO_NA_1 (IOA := 41), C_BO_TA_1 (IOA := 42).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50), C_SE_NA_1 (IOA := 51), C_SE_TA_1 (IOA := 52).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60), C_SE_NB_1 (IOA := 61), C_SE_TB_1 (IOA := 62).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70), C_SE_NC_1 (IOA := 71), C_SE_TC_1 (IOA := 72).

Sample implementation of the following services:

- End of initialization (M_EI_NA_1) is sent once.
- Test command (C_TS_TA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN, Mode A). Counters are stored locally in the substation every 15 seconds and transmitted as spontaneous data to the central station.
- Command execution of some commands (direct commands, see upper data point configuration).
- Simulation and transmission of spontaneous data every 10 seconds if the variable **bSendSpont** was set to TRUE (see upper data point configuration).

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol
hServer	T_HSERVER (STRUCT)	The server connection handle.
fbServer	FB_IEC870_5_104SlaveGrp (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
F_CreateServerHnd	FUNCTION	This function initializes the server connection handle.
AODB	ARRAY OF ST_IEC870_5_101AODBEntry (STRUCT)	Application object database. It's an array variable. An array element is required for each data point.
F_iecInitAOEntry	FUNCTION	In the initialization step, the individual data points of the station are configured once with this function call.
memory	ST_MemoryData (STRUCT)	This structure contains constantly updated process data – either received or to be sent.
F_iecSetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be set/reset.
F_iecGetAOQuality	FUNCTION	This function demonstrates how the quality flags of M_SP_NA_1(IOA := 100) can be queried.
aConnections	ARRAY OF ST_IEC870_5_104ServerConnection (STRUCT)	Each array element represents a connection to the central station.

6.5.2.1.6 Scenarios

6.5.2.1.6.1 Spontaneous data communication and general query

The sample program presented here shows an IEC 60870-5-104 slave with High Level interface and a simple communication of two data points. One data point is transferred spontaneously in control direction, the other in monitor direction. In addition, a general interrogation is configured for the data point in monitor direction.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243749515.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave/your own IP): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave/own port): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2.1.6.2 General query and groups

The sample program presented here shows an IEC 60870-5-104 slave with High Level interface. The sample illustrates the definition of groups of data types. A general interrogation and a separate group 1 are configured. The general interrogation takes place every 30 seconds, group 1 is interrogated every 10 seconds. The communicated parameters include single points, bitstrings and measured values.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243754123.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2.1.6.3 Counting value query and time synchronization

The sample program presented here shows an IEC 60870-5-104 slave with High Level interface. At the core of the sample is the time synchronization from the master to the slave and the counter interrogation. In the sample, the time is synchronized every 120 seconds. A data type M_IT_NA_1 is used to transfer the counting value via a general counter interrogation.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1243758731.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8

- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- The applications represent an exemplary implementation of the High Level interface. We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2.2 Low-Level Interface

6.5.2.2.1 Quick start

Simple projects with complete sources can be found here: IEC 60870-5-104 Substation low-level sample [▶ 779].

A PLC application that is to communicate with a central station via the "low-level" interface requires the following resources:

- An instance of the communication function block: FB_IEC870_5_104TProtocol [▶ 514];
- An instance of the function block for establishing the TCP/IP connection: FB_ServerClientConnection or FB_ClientServerConnection;
- An instance of the TX/RX data buffer: ST_IEC870_5_101TBuffer [▶ 583];
- An instance of the function block for manipulating the TX/RX data buffer: FB_IEC870_5_101TBufferCtrl [▶ 506];

6.5.2.2.2 IEC 60870-5-104 controlled station (slave)

Simple substation application using the TwinCAT IEC 60870-5-104 low-level interface. This sample application can communicate with a central station via a network interface.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/984648203.zip

In the sample project, the following station parameters are set for the substation:

- Server host address (IP address of the substation): **127.0.0.1 (You must at least adapt this parameter on your target platform!)**
- Server port address (port address of the substation): **2404.**
- k: **12.**
- w: **8.**
- t0: **30s.**
- t1: **15s.**
- t2: **10s.**
- t3: **20s.**
- Cause of transfer size: **2 octets (including originator address).**
- Common ASDU address size: **2 octets.**

- Information object address size: **3 octets.**
- Originator address: **0.**
- Common ASDU address: **7.**
- Max. APDU length: **253.**

Data point configuration (it should be the same as the data point configuration of the central station):

- Single point information: M_SP_NA_1 (IOA := 100), M_SP_NA_1(IOA := 101), M_SP_TB_1(IOA := 102).
- Double point information: M_DP_NA_1 (IOA := 200).
- Regulating step value: M_ST_NA_1 (IOA := 300).
- Bitstring value: M_BO_NA_1 (IOA := 400).
- Measured value, normalized value: M_ME_NA_1 (IOA := 500).
- Measured value, scaled value: M_ME_NB_1 (IOA := 600).
- Measured value, short floating point value: M_ME_NC_1 (IOA := 700).
- Binary counter value: M_IT_NA_1 (IOA := 800), M_IT_NA_1 (IOA := 801), M_IT_TB_1 (IOA := 802).
- Single command: C_SC_NA_1 (IOA := 10).
- Double command: C_DC_NA_1 (IOA := 20).
- Regulating step command: C_RC_NA_1 (IOA := 30).
- Bitstring command: C_BO_NA_1 (IOA := 40).
- Setpoint command, normalized value: C_SE_NA_1 (IOA := 50).
- Setpoint command, scaled value: C_SE_NB_1 (IOA := 60).
- Setpoint command, short floating point value: C_SE_NC_1 (IOA := 70).

Sample implementation of the following services:

- Test command (C_TS_TA_1).
- Clock synchronization command (C_CS_NA_1).
- General interrogation command (C_IC_NA_1) of the group: General (eIEC870_QOI_INROGEN).
- Counter interrogation command (C_IC_NA_1) of the group: General (eIEC870_RQT_REQCOGEN), Mode A. The counter values are frozen locally every 60 seconds.
- Command execution of some commands (see upper data point configuration).
- End of initialization (M_EI_NA_1) is sent once to the central station (after the connection has been established).
- Cyclic data transfer every 60 seconds.
- Simulation and transfer of spontaneous data every 60 seconds if the variable **bSimulate** is set to TRUE.

The following table briefly describes the main components of the project:

Component	Type	Description
MAIN	PRG	This program part implements the telecontrol protocol.
fbServer	FB_IEC870_5_104S (FB)	Implements the substation. An instance of this function block can only communicate with one central station. For each additional central station to communicate with, another instance of this function block is required.
fbServices	FB_IEC870_5_104SServices (FB)	Includes a simple implementation of some services such as general interrogation, clock synchronization, test commands, etc.
stProcPara	ST_IEC870_5_104SProcData (STRUCT)	Here you can found constantly updated process data – either received or to be sent.
sysPara	ST_IEC870_5_104SSysPara (STRUCT)	Configuration of the system parameters of the substation and the implemented services.
fbLog	FB_IEC870_DebugLogFifo (FB)	A simple log buffer for debugging/troubleshooting purposes.

6.5.2.2.3 Scenarios

Basic explanatory notes regarding the Low Level scenarios:

The initialization step begins with the configuration of the TCP/IP interface, during which the IP address and the communication port are set. Then, various protocol parameters and parameters of the send and receive buffers are set. This is followed by establishment of the TCP/IP connection with the aid of function blocks from the TCP/IP function.

Data are then received and loaded from the receive buffer with the action **fbBuffer.RxRemoveObj** until the buffer is empty. The received data are sorted by type ID and object address. After sorting, the respective data are stored in a variable with the function **F_iecCopyStreamToBuffer** and then split according to the components set out in the standard. If the command for a general interrogation or the interrogation command for a specific group of interrogations is received in the slave, the corresponding transfer procedure for interrogation commands is triggered immediately, and the data points defined for the respective group are written to the send buffer. Sending of data is described in more detail in the following paragraph.

Both in the master and in the slave this is followed by sending of the data to be transferred outside of or in addition to an interrogation group. First, common transfer details of the data points such as ASDU address, number of objects to be transferred, reason for the data transfer and priority class are written to the structure **stTxAsdu.ident**. Then the protocol frames of each data point, consisting of type ID, object address and object-specific data, are written to the structure **stTxAsdu**. This is followed by copying via the function **F_iecCopyBufferToStream** and writing to the send buffer via the action **fbBuffer.TxAddObj**. Finally, any errors are read from the error memory and listed in an error output.

The process of reading, writing and subsequent error handling is repeated cyclically.

6.5.2.2.3.1 Spontaneous data communication and general query

The sample program presented here shows an IEC 60870-5-104 slave with Low Level interface and a simple communication of two data points. One data point is transferred spontaneously in control direction, the other in monitor direction. In addition, a general interrogation is configured for the data point in monitor direction.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244578443.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8

- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2.2.3.2 General query and groups

The sample program presented here shows an IEC 60870-5-104 slave with Low Level interface. The sample illustrates the definition of groups of data types. A general interrogation and a separate group 1 are configured. The general interrogation takes place every 30 seconds, group 1 is interrogated every 10 seconds. The communicated parameters include single points, bitstrings and measured values.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244595851.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

6.5.2.2.3.3 Counting value query and time synchronization

The sample program presented here shows an IEC 60870-5-104 slave with Low Level interface. At the core of the sample is the time synchronization from the master to the slave and the counter interrogation. In the sample, the time is synchronized every 120 seconds. A data type M_IT_NA_1 is used to transfer the counting value via a general counter interrogation.

Download TwinCAT XAE Project (*.zip): https://infosys.beckhoff.com/content/1033/TF6500_TC3_IEC60870_5_10x/Resources/1244600459.zip

In the sample, the following parameters are set for the substation (slave):

- Server Host address (IP address of the slave): 127.0.0.1 (At least this parameter must be adjusted individually)
- Server port address (port address of the slave): 2404
- k: 12
- w: 8
- t0: 30 s
- t1: 15 s
- t2: 10 s
- t3: 20 s
- General ASDU address: 3
- Cause of transfer size: 2 octets
- Common ASDU address size: 2 octets
- Information object address size: 3 octets
- Max. APDU length: 253

- We make no claim to completeness and correctness. Set the IP address of the slave, both in the master part and in the slave part! Verify that port 2404 is not blocked by a firewall!
- The connection is established after the start of the program. The data points are managed in the Global Variable List (GVL).

7 Appendix

7.1 IEC 60870-5-101

7.1.1 TwinCAT IEC 60870-5-101 error codes

Requirements

Codes (hex)	Codes (dec)	Error source	Description
0x00000000-0x00002000	0-8192	TF6340 Serial-Communication error codes	TF6340 Serial-Communication errors (only if error source [▶ 610] = eIEC870_ESRC_IEC60870_5_101LINK).
0x00000000-0x00007800	0-30720	TwinCAT System error codes [▶ 784]	TwinCAT System errors (ADS error codes inclusive only if error error source [▶ 610] <> eIEC870_ESRC_IEC60870_5_101LINK).
0x00008100-0x000081FF	32768-33023	TwinCAT IEC 60870-5-10x error codes [▶ 658]	TwinCAT IEC 60870-5-10x protocol/ communication errors (e.g. timeout errors, telegram errors, configuration errors).

7.1.1.1 ADS Return Codes

Grouping of error codes:
Global error codes: 0x0000 [▶ 784]... (0x9811_0000 ...)
Router error codes: 0x500 [▶ 785]... (0x9811_0500 ...)
General ADS errors: 0x700 [▶ 785]... (0x9811_0700 ...)
RTime error codes: 0x1000 [▶ 787]... (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, not reachable or not installed.
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.
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. Several causes are possible. For example, an incorrect password was entered when creating routes.
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. Several causes are possible. For example, a unidirectional ADS route is used in the opposite direction.
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.
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.

Hex	Dec	HRESULT	Name	Description
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.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was canceled.

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 [▶ 809]			

7.1.2 Troubleshooting/diagnostics

General checks:

- Check the hardware and software requirements described in this documentation (TwinCAT version, CE image version etc.);
- Check the IO configuration and mapping of PLC variables in TwinCAT System Manager (configuration of serial interface [▶ 662], baudrate, parity, stoppbits settings). Compare them to the parameters set in the communication partner;
- Check the configured data points (type, information object address etc.);
- Check if the other communication partner issues an error code;
- Activate the debug output [▶ 789] during connection establishment and/or of ASDU data. Open the TwinCAT System Manager and activate the LogView window. Analyze/check the debug output strings;

High-Level interface:

- Especially by control station:
  - Check whether the FB_IEC870_5_101Master [▶ 488] function block issues an error code/source [▶ 784];
  - Compare the interoperability check list of control station [▶ 13] and controlled station;
  - Check the protocol parameters [▶ 640] that are transferred to the FB_IEC870_5_101Master function block (link address, link address octet size, FRAMELength etc.). Compare them to the parameters set in the controlled station;
  - Check the system parameter [▶ 588] that are transferred to the FB_IEC870_5_101Master function block (ASDU address, ASDU address octet size, information object address octet size, cause of transfer octet size, etc.). Compare them to the the parameters set in the controlled station;
  - Check the acquisition parameters [▶ 593] that are transferred to the FB_IEC870_5_101Master function block (initialisation sequence, cyclic general interrogation, cyclic counter interrogation, cyclic test commands, etc.);
- Especially by controlled station:

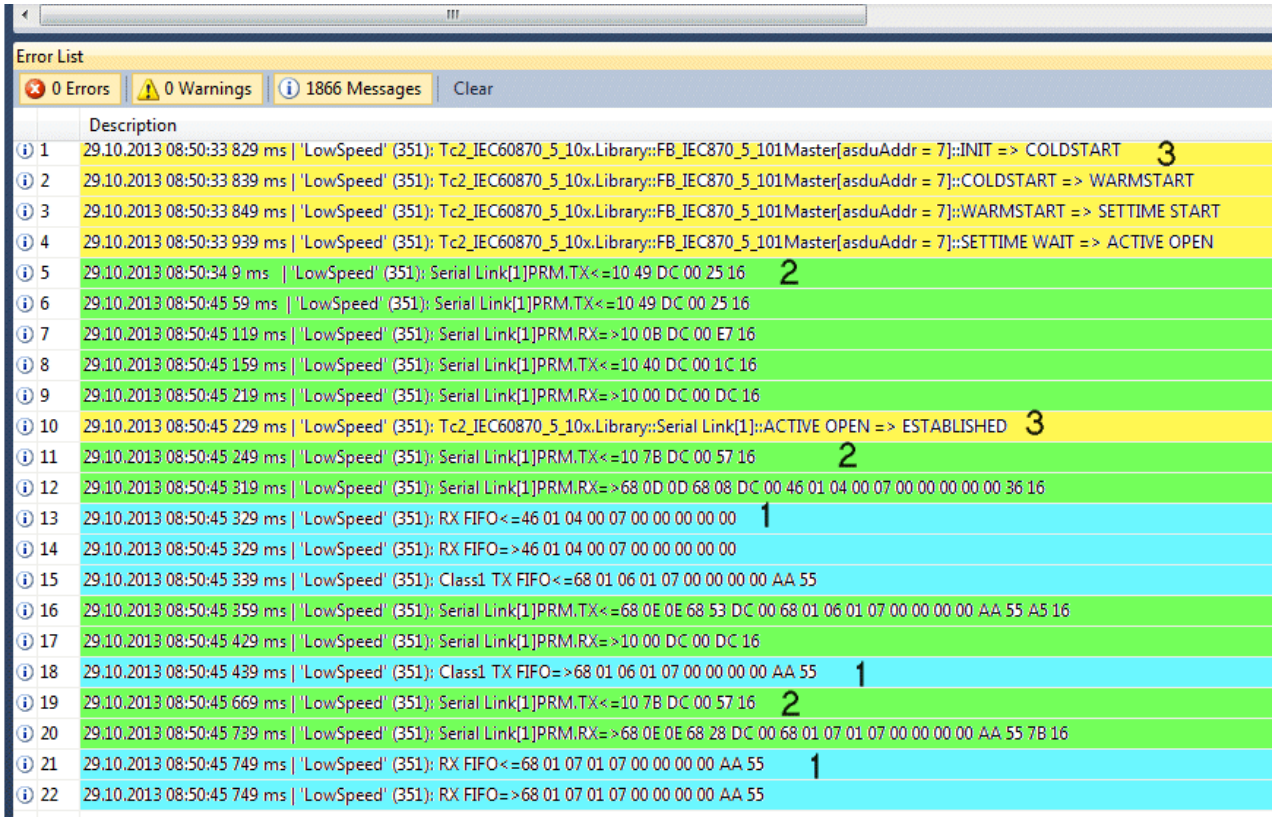
- Check whether the FB_IEC870_5_101Slave [▶ 490] function block issues an error code/source [▶ 784];
- Compare the interoperability check list of controlled station [▶ 30] and control station;
- Check the protocol parameters [▶ 640] that are transferred to the FB_IEC870_5_101Slave function block (link address, link address octet size, FRAMELength etc.). Compare them to the parameters set in the controlling station;
- Check the system parameter [▶ 588] that are transferred to the FB_IEC870_5_101Slave function block (ASDU address, ASDU address octet size, information object address octet size, cause of transfer octet size, etc.). Compare them to the the parameters set in the controlling station;

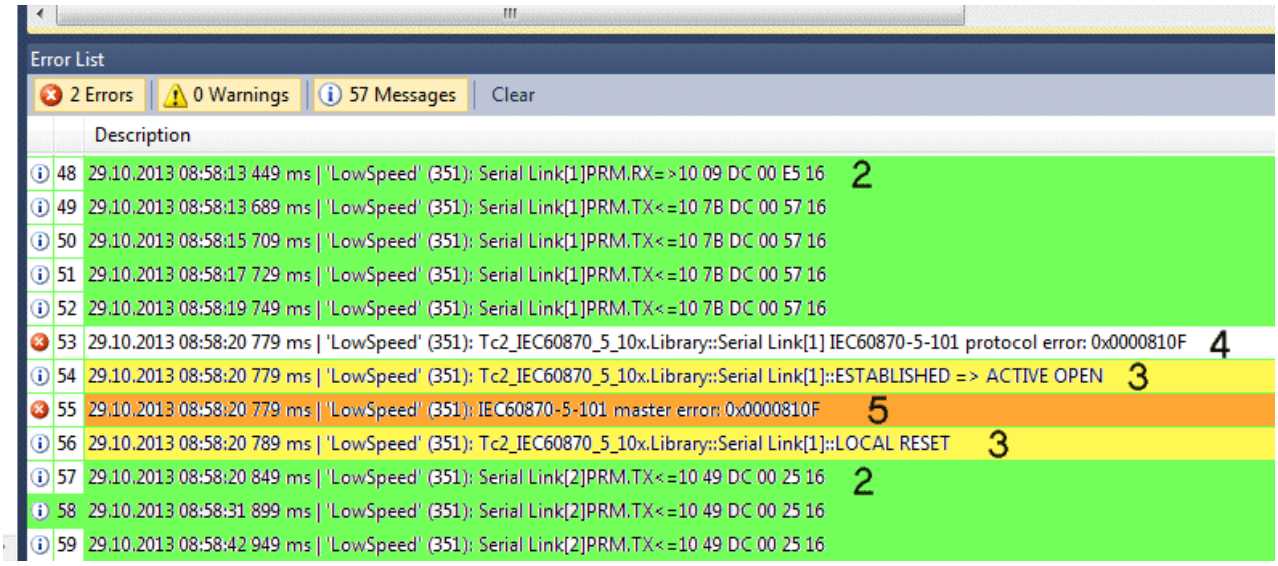
Low-Level interface:

- Check whether the FB_IEC870_5_101TProtocol [▶ 510] function block issues an error code [▶ 784];
- Compare the interoperability check list of control station and controlled station;
- Check the protocol parameters [▶ 640] that are transferred to the FB_IEC870_5_101TProtocol function block (link address, link address octet size, FRAMELength etc.). Compare them to the parameters set in the communication partner;
- Check the configured address lengths (instance of ST_IEC870_5_101TBuffer [▶ 583], TX/RX data buffer): ASDU address octet size, information object address octet size, cause of transfer octet size, max ASDU size. Compare them to the parameters set in the communication partner;

7.1.3 Debug messages in TwinCAT->Error List

Debugging messages written to the application log facilitate troubleshooting of the system. In order to view the activated debug messages start the TwinCAT XAE and select the View->Other Windows->Error List menue command. A debug output is shown below. The different message types are identified with the respective numbers.





The messages are activated in different ways (with different parameters) depending on whether you use the "Low-Level" or "High-Level" interface. Below the list of the different parameters.

1. Hexadecimal output of the ASDUs (without link layer control header). 32 ASDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF	Controlled station		Control station	
			Low-Level	High-Level	Low-Level	High-Level
ST_IEC870_5_101TBuffer [▶ 583].eDbg	eIEC870_FIFO_DBG_ALL	eIEC870_FIFO_DBG_OFF	Yes	-	Yes	-
FB_IEC870_5_101Slave [▶ 490].sysPara.dbgMode	IEC870_DEBUGMODE_ASDU	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_101Master [▶ 488].sysPara.dbgMode	IEC870_DEBUGMODE_ASDU	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

2. Hexadecimal output of the APDUs (TCP/IP frames). 32 APDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF	Controlled station		Control station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_101TProtocol [▶ 510].bOutDbg	TRUE	FALSE	Yes	-	Yes	-
FB_IEC870_5_101Slave [▶ 490].sysPara.dbgMode	IEC870_DEBUGMODE_LINKLAYER	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_101Master [▶ 488].sysPara.dbgMode	IEC870_DEBUGMODE_LINKLAYER	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes-

3. State change messages:

Name	ON	OFF	Controlled station		Control station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_101Slave [▶ 490].sysPara.dbgMode	IEC870_DEBUGMODE_DEVSTATE	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_101Master [▶ 488].sysPara.dbgMode	IEC870_DEBUGMODE_DEVSTATE	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

4: Link layer error messages:

Name	ON	OFF	Controlled station		Control station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_101Slave [▶ 490].sysPara.dbgMode	IEC870_DEBUGMODE_LINKERROR	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_101Master [▶ 488].sysPara.dbgMode	IEC870_DEBUGMODE_LINKERROR	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

5. Messages logged from the PLC application (sample).

Further diagnostic tools:

- Portmon for Windows (v3.02, Windows Sysinternals);
- Diverse protocoll test suite products;

7.2 IEC 60870-5-102

7.2.1 TwinCAT IEC 60870-5-102 error codes

Requirements

Codes (hex)	Codes (dec)	Error source	Description
0x00000000-0x00002000	0-8192	TF6340 Serial-Communication error codes	TF6340 Serial-Communication errors (only if error source [▶ 610] = eIEC870_ESRC_IEC60870_5_101LINK).
0x00000000-0x00007800	0-30720	TwinCAT System error codes [▶ 793]	TwinCAT System errors (ADS error codes inclusive only if error error source [▶ 610] <> eIEC870_ESRC_IEC60870_5_101LINK).
0x00008100-0x000081FF	32768-33023	TwinCAT IEC 60870-5-10x error codes [▶ 658]	TwinCAT IEC 60870-5-10x protocol/ communication errors (e.g. timeout errors, telegram errors, configuration errors).

7.2.1.1 ADS Return Codes

Grouping of error codes:
Global error codes: 0x0000 [▶ 793]... (0x9811_0000 ...)
Router error codes: 0x500 [▶ 794]... (0x9811_0500 ...)
General ADS errors: 0x700 [▶ 794]... (0x9811_0700 ...)
RTime error codes: 0x1000 [▶ 796]... (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, not reachable or not installed.
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.
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. Several causes are possible. For example, an incorrect password was entered when creating routes.
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. Several causes are possible. For example, a unidirectional ADS route is used in the opposite direction.
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.
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.

Hex	Dec	HRESULT	Name	Description
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.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was canceled.

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

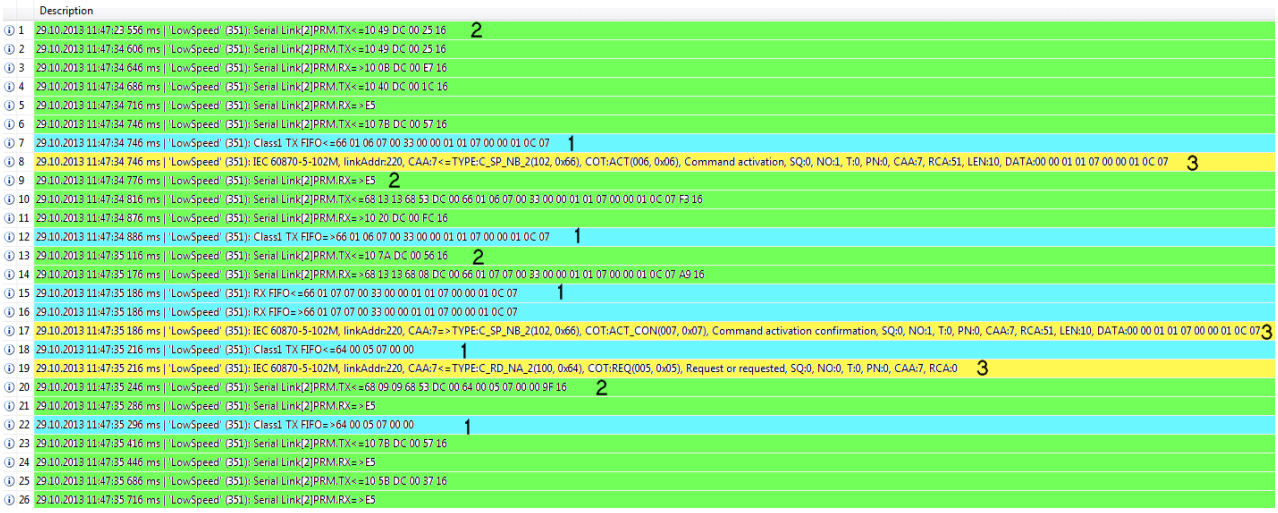
Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not 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 [▶ 809]			

7.2.2 Troubleshooting/diagnostics

- Check the hardware and software requirements described in this documentation (TwinCAT version, CE image version etc.);
- Compare the interoperability check list of control station [▶ 47] and substation;
- Check the IO configuration and mapping of PLC variables in TwinCAT System Manager (configuration of serial interface [▶ 662], baudrate, parity, stoppbits settings). Compare them to the parameters set in the substation;
- Check whether the FB_IEC870_5_102TProtocol [▶ 511] function block issues an error code [▶ 793];
- Check the protocol parameters [▶ 640] that are transferred to the FB_IEC870_5_102TProtocol function block (link address, FRAMELength etc.). Compare them to the parameters set in the substation;
- Check the configured address lengths (instance of ST_IEC870_5_102TBuffer [▶ 645], TX/RX data buffer): ASDU address octet size, max ASDU size. Compare them to the parameters set in the substation;
- Check the configured data points (type, information object address etc.);
- Check if the other communication partner issues an error code;
- Activate the debug output [▶ 797] during connection establishment and/or of ASDU data. Open the TwinCAT System Manager and activate the LogView window. Analyze/check the debug output strings;

7.2.3 Debug messages in TwinCAT->Error List

Debugging messages written to the TwinCAT XAE->Error List facilitate troubleshooting of the system. In order to view the activated debug messages start the TwinCAT XAE and select the View->Other Windows->Error List menue command. A debug output is shown below. The different message types are identified with the respective numbers.



The messages are activated in different ways (with different parameters). Below the list of the different parameter names.

1. Hexadecimal output of the ASDUs (without link layer control header). 32 ASDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF
ST_IEC870_5_102TBuffer [▶ 645].eDbg	eIEC870_FIFO_DBG_ALL	eIEC870_FIFO_DBG_OFF

2. Hexadecimal output of the APDUs (RS232/RS485). 32 APDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF
FB_IEC870_5_102TProtocol [▶ 511].bOutDbg	TRUE	FALSE

3. Messages from PLC application logged using the FB_IEC870_DebugLogFifo function block.

Name	ON	OFF
FB_IEC870_DebugLogFifo [▶ 521]	Aktionsaufruf: A_LogError, A_LogWarning, A_LogHint	-

Further diagnostic tools:

- Portmon for Windows (v3.02, Windows Sysinternals);
- Diverse protocoll test suite products;

7.3 IEC 60870-5-103

7.3.1 TwinCAT IEC 60870-5-103 error codes

Requirements

Codes (hex)	Codes (dec)	Error source	Description
0x00000000-0x00002000	0-8192	TF6340 Serial-Communication error codes	TF6340 Serial-Communication errors (only if error source [▶ 610] = eIEC870_ESRC_IEC60870_5_101LINK).
0x00000000-0x00007800	0-30720	TwinCAT System error codes [▶ 799]	TwinCAT System errors (ADS error codes inclusive only if error error source [▶ 610] <> eIEC870_ESRC_IEC60870_5_101LINK).
0x00008100-0x000081FF	32768-33023	TwinCAT IEC 60870-5-10x error codes [▶ 658]	TwinCAT IEC 60870-5-10x protocol/ communication errors (e.g. timeout errors, telegram errors, configuration errors).

7.3.1.1 ADS Return Codes

Grouping of error codes:
Global error codes: 0x0000 [▶ 799]... (0x9811_0000 ...)
Router error codes: 0x500 [▶ 800]... (0x9811_0500 ...)
General ADS errors: 0x700 [▶ 800]... (0x9811_0700 ...)
RTime error codes: 0x1000 [▶ 802]... (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, not reachable or not installed.
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.
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. Several causes are possible. For example, an incorrect password was entered when creating routes.
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. Several causes are possible. For example, a unidirectional ADS route is used in the opposite direction.
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.
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.

Hex	Dec	HRESULT	Name	Description
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.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was canceled.

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

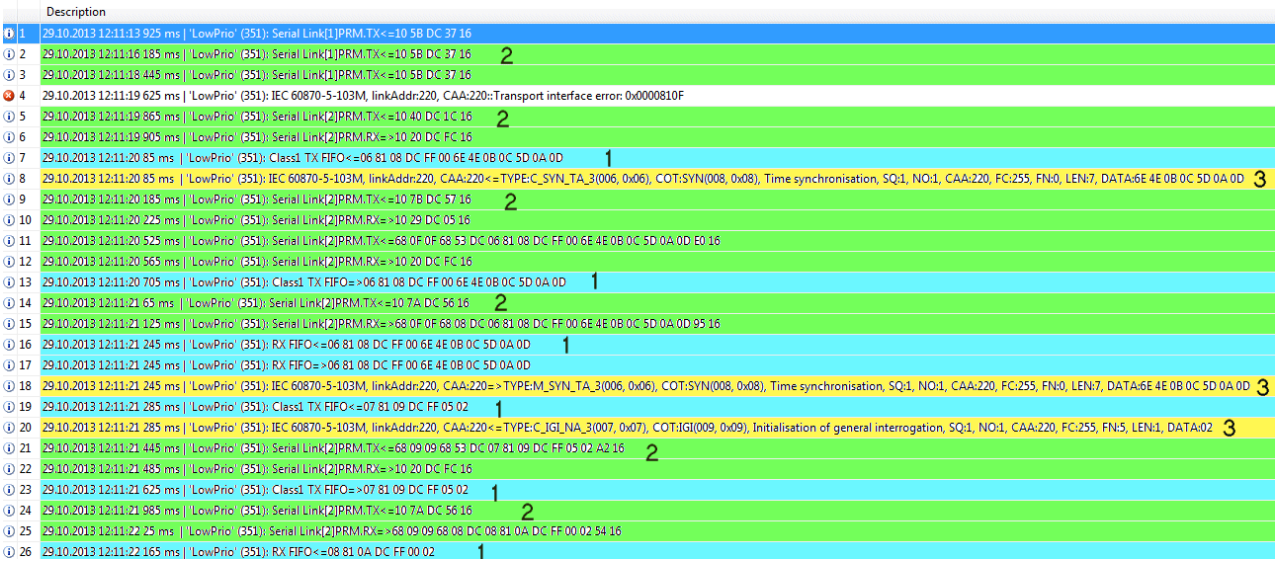
Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not 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 [▶ 809]			

7.3.2 Troubleshooting/diagnostics

- Check the hardware and software requirements described in this documentation (TwinCAT version, CE image version etc.);
- Compare the interoperability check list of control station [▶ 56] and substation;
- Check the IO configuration and mapping of PLC variables in TwinCAT System Manager (configuration of serial interface [▶ 662], baudrate, parity, stoppbits settings). Compare them to the parameters set in the substation;
- Check whether the FB_IEC870_5_103TProtocol [▶ 513] function block issues an error code [▶ 799];
- Check the protocol parameters [▶ 640] that are transferred to the FB_IEC870_5_103TProtocol function block (link address, link address octet size, FRAMELength etc.). Compare them to the parameters set in the substation;
- Check the max. configured ASDU length (instance of ST_IEC870_5_103TBuffer [▶ 648], TX/RX data buffer). Compare them to the parameters set in the substation;
- Check the configured data points (type, ASDU address, function number, information number etc.);
- Check if the other communication partner issues an error code;
- Activate the debug output [▶ 803] during connection establishment and/or of ASDU data. Open the TwinCAT System Manager and activate the LogView window. Analyze/check the debug output strings;

7.3.3 Debug messages in TwinCAT->Error List

Debugging messages written to the TwinCAT XAE->Error List facilitate troubleshooting of the system. In order to view the activated debugging messages start the TwinCATT XAE and select the View->Other Windows->Error List menue command. A debug output is shown below. The two different message types are identified with the respective numbers.



The messages are activated in different ways (with different parameters). Below the list of the different parameter names.

1. Hexadecimal output of the ASDUs (without link layer control header). 32 ASDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF
ST_IEC870_5_103TBuffer [▶ 648].eDbg	eIEC870_FIFO_DBG_ALL	eIEC870_FIFO_DBG_OFF

2. Hexadecimal output of the APDUs (RS232/RS485). 32 APDU data bytes per row are output as hexadecimal numbers. Bigger frames are split into multiple lines:

Name	ON	OFF
FB_IEC870_5_103TProtocol [▶ 513].bOutDbg	TRUE	FALSE

3. Messages from PLC application logged using the FB_IEC870_DebugLogFifo function block.

Name	ON	OFF
FB_IEC870_DebugLogFifo [▶ 521]	Action call: A_LogError, A_LogWarning, A_LogHint	-

Further diagnostic tools:

- Portmon for Windows (v3.02, Windows Sysinternals);
- Diverse protocoll test suite products;

7.4 IEC 60870-5-104

7.4.1 TwinCAT IEC 60870-5-104 error codes

Requirements

Codes (hex)	Codes (dec)	Error source	Description
0x00000000-0x00007800	0-30720	TwinCAT System error codes [▶ 805]	TwinCAT System error (ADS- error code inclusive).
0x00008000-0x000080FF	32768-33023	TF6310 TCP/IP error codes	TF6310 TCP/IP error codes.
0x00008100-0x000081FF	33024-33279	TwinCAT IEC 60870-5-10x error codes [▶ 658]	TwinCAT IEC 60870-5-10x protocol/ communication errors (e.g. timeout errors, telegram errors, configuration errors).
0x80070000-0x8007FFFF	2147942400-2148007935	Higher word = 0x8007 (fix) Lower word = Win32 system error codes [▶ 809]	Win32 System errors (Windows Sockets error codes inclusive).

7.4.1.1 ADS Return Codes

Grouping of error codes:
Global error codes: 0x0000 [▶ 805]... (0x9811_0000 ...)
Router error codes: 0x500 [▶ 806]... (0x9811_0500 ...)
General ADS errors: 0x700 [▶ 806]... (0x9811_0700 ...)
RTime error codes: 0x1000 [▶ 808]... (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, not reachable or not installed.
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.
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. Several causes are possible. For example, an incorrect password was entered when creating routes.
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. Several causes are possible. For example, a unidirectional ADS route is used in the opposite direction.
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.
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.

Hex	Dec	HRESULT	Name	Description
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.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was canceled.

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 [▶ 809]			

7.4.1.2 Win32 Error Codes

The following table provides a list of Win32 error codes.

0 [▶ 810], 100 [▶ 812], 200 [▶ 814], 1001 [▶ 816], 1100 [▶ 818], 1200 [▶ 820], 1400 [▶ 825], 1600 [▶ 827], 1800 [▶ 831], 2000 [▶ 833], 3000 [▶ 834], 5000 [▶ 837], 6000 [▶ 840], 8000 [▶ 842], 8500 [▶ 851], 9001 [▶ 855], 10004 [▶ 857], 12000 [▶ 860],

Error			Description
decimal	Hexadecimal	Name	
0	0x00000000	ERROR_SUCCESS	The operation completed successfully.
1	0x00000001	ERROR_INVALID_FUNCTION	Incorrect function.
2	0x00000002	ERROR_FILE_NOT_FOUND	The system cannot find the file specified.
3	0x00000003	ERROR_PATH_NOT_FOUND	The system cannot find the path specified.
4	0x00000004	ERROR_TOO_MANY_OPEN_FILES	The system cannot open the file.
5	0x00000005	ERROR_ACCESS_DENIED	Access is denied.
6	0x00000006	ERROR_INVALID_HANDLE	The handle is invalid.
7	0x00000007	ERROR_ARENA_TRASHED	The storage control blocks were destroyed.
8	0x00000008	ERROR_NOT_ENOUGH_MEMORY	Not enough storage is available to process this command.
9	0x00000009	ERROR_INVALID_BLOCK	The storage control block address is invalid.
10	0x0000000A	ERROR_BAD_ENVIRONMENT	The environment is incorrect.
11	0x0000000B	ERROR_BAD_FORMAT	An attempt was made to load a program with an incorrect format.
12	0x0000000C	ERROR_INVALID_ACCESS	The access code is invalid.
13	0x0000000D	ERROR_INVALID_DATA	The data is invalid.
14	0x0000000E	ERROR_OUTOFMEMORY	Not enough storage is available to complete this operation.
15	0x0000000F	ERROR_INVALID_DRIVE	The system cannot find the drive specified.
16	0x00000010	ERROR_CURRENT_DIRECTORY	The directory cannot be removed.
17	0x00000011	ERROR_NOT_SAME_DEVICE	The system cannot move the file to a different disk drive.
18	0x00000012	ERROR_NO_MORE_FILES	There are no more files.
19	0x00000013	ERROR_WRITE_PROTECT	The media is write protected.
20	0x00000014	ERROR_BAD_UNIT	The system cannot find the device specified.
21	0x00000015	ERROR_NOT_READY	The device is not ready.
22	0x00000016	ERROR_BAD_COMMAND	The device does not recognize the command.
23	0x00000017	ERROR_CRC	Data error (cyclic redundancy check).
24	0x00000018	ERROR_BAD_LENGTH	The program issued a command but the command length is incorrect.
25	0x00000019	ERROR_SEEK	The drive cannot locate a specific area or track on the disk.
26	0x0000001A	ERROR_NOT_DOS_DISK	The specified disk or diskette cannot be accessed.
27	0x0000001B	ERROR_SECTOR_NOT_FOUND	The drive cannot find the sector requested.
28	0x0000001C	ERROR_OUT_OF_PAPER	The printer is out of paper.
29	0x0000001D	ERROR_WRITE_FAULT	The system cannot write to the specified device.
30	0x0000001E	ERROR_READ_FAULT	The system cannot read from the specified device.
31	0x0000001F	ERROR_GEN_FAILURE	A device attached to the system is not functioning.
32	0x00000020	ERROR_SHARING_VIOLATION	The process cannot access the file because it is being used by another process.
33	0x00000021	ERROR_LOCK_VIOLATION	The process cannot access the file because another process has locked a portion of the file.
34	0x00000022	ERROR_WRONG_DISK	The wrong diskette is in the drive. Insert %2 (Volume Serial Number: %3) into drive %1.
36	0x00000024	ERROR_SHARING_BUFFER_EXCEEDED	Too many files opened for sharing.
38	0x00000026	ERROR_HANDLE_EOF	Reached the end of the file.
39	0x00000027	ERROR_HANDLE_DISK_FULL	The disk is full.
50	0x00000032	ERROR_NOT_SUPPORTED	The request is not supported.
51	0x00000033	ERROR_REM_NOT_LIST	The remote computer is not available.
52	0x00000034	ERROR_DUP_NAME	A duplicate name exists on the network.
53	0x00000035	ERROR_BAD_NETPATH	The network path was not found.
54	0x00000036	ERROR_NETWORK_BUSY	The network is busy.
55	0x00000037	ERROR_DEV_NOT_EXIST	The specified network resource or device is no longer available.
56	0x00000038	ERROR_TOO_MANY_CMDS	The network BIOS command limit has been reached.
57	0x00000039	ERROR_ADAP_HDW_ERR	A network adapter hardware error occurred.

Error			Description
decimal	Hexadecimal	Name	
58	0x0000003A	ERROR_BAD_NET_RESP	The specified server cannot perform the requested operation.
59	0x0000003B	ERROR_UNEXP_NET_ERR	An unexpected network error occurred.
60	0x0000003C	ERROR_BAD_REM_ADAP	The remote adapter is not compatible.
61	0x0000003D	ERROR_PRINTQ_FULL	The printer queue is full.
62	0x0000003E	ERROR_NO_SPOOL_SPACE	Space to store the file waiting to be printed is not available on the server.
63	0x0000003F	ERROR_PRINT_CANCELLED	Your file waiting to be printed was deleted.
64	0x00000040	ERROR_NETNAME_DELETED	The specified network name is no longer available.
65	0x00000041	ERROR_NETWORK_ACCESS_DENIED	Network access is denied.
66	0x00000042	ERROR_BAD_DEV_TYPE	The network resource type is not correct.
67	0x00000043	ERROR_BAD_NET_NAME	The network name cannot be found.
68	0x00000044	ERROR_TOO_MANY_NAMES	The name limit for the local computer network adapter card was exceeded.
69	0x00000045	ERROR_TOO_MANY_SESS	The network BIOS session limit was exceeded.
70	0x00000046	ERROR_SHARING_PAUSED	The remote server has been paused or is in the process of being started.
71	0x00000047	ERROR_REQ_NOT_ACCEP	No more connections can be made to this remote computer at this time because there are already as many connections as the computer can accept.
72	0x00000048	ERROR_REDIR_PAUSED	The specified printer or disk device has been paused.
80	0x00000050	ERROR_FILE_EXISTS	The file exists.
82	0x00000052	ERROR_CANNOT_MAKE	The directory or file cannot be created.
83	0x00000053	ERROR_FAIL_I24	Fail on INT 24.
84	0x00000054	ERROR_OUT_OF_STRUCTURES	Storage to process this request is not available.
85	0x00000055	ERROR_ALREADY_ASSIGNED	The local device name is already in use.
86	0x00000056	ERROR_INVALID_PASSWORD	The specified network password is not correct.
87	0x00000057	ERROR_INVALID_PARAMETER	The parameter is incorrect.
88	0x00000058	ERROR_NET_WRITE_FAULT	A write fault occurred on the network.
89	0x00000059	ERROR_NO_PROC_SLOTS	The system cannot start another process at this time.

Error			Description
decimal	Hexadecimal	Name	
100	0x00000064	ERROR_TOO_MANY_SEMAPHORES	Cannot create another system semaphore.
101	0x00000065	ERROR_EXCL_SEM_ALREADY_OWNED	The exclusive semaphore is owned by another process.
102	0x00000066	ERROR_SEM_IS_SET	The semaphore is set and cannot be closed.
103	0x00000067	ERROR_TOO_MANY_SEM_REQUESTS	The semaphore cannot be set again.
104	0x00000068	ERROR_INVALID_AT_INTERRUPT_TIME	Cannot request exclusive semaphores at interrupt time.
105	0x00000069	ERROR_SEM_OWNER_DIED	The previous ownership of this semaphore has ended.
106	0x0000006A	ERROR_SEM_USER_LIMIT	Insert the diskette for drive %1.
107	0x0000006B	ERROR_DISK_CHANGE	The program stopped because an alternate diskette was not inserted.
108	0x0000006C	ERROR_DRIVE_LOCKED	The disk is in use or locked by another process.
109	0x0000006D	ERROR_BROKEN_PIPE	The pipe has been ended.
110	0x0000006E	ERROR_OPEN_FAILED	The system cannot open the device or file specified.
111	0x0000006F	ERROR_BUFFER_OVERFLOW	The file name is too long.
112	0x00000070	ERROR_DISK_FULL	There is not enough space on the disk.
113	0x00000071	ERROR_NO_MORE_SEARCH_HANDLES	No more internal file identifiers available.
114	0x00000072	ERROR_INVALID_TARGET_HANDLE	The target internal file identifier is incorrect.
117	0x00000075	ERROR_INVALID_CATEGORY	The IOCTL call made by the application program is not correct.
118	0x00000076	ERROR_INVALID_VERIFY_SWITCH	The verify-on-write switch parameter value is not correct.
119	0x00000077	ERROR_BAD_DRIVER_LEVEL	The system does not support the command requested.
120	0x00000078	ERROR_CALL_NOT_IMPLEMENTED	This function is not supported on this system.
121	0x00000079	ERROR_SEM_TIMEOUT	The semaphore timeout period has expired.
122	0x0000007A	ERROR_INSUFFICIENT_BUFFER	The data area passed to a system call is too small.
123	0x0000007B	ERROR_INVALID_NAME	The filename, directory name, or volume label syntax is incorrect.
124	0x0000007C	ERROR_INVALID_LEVEL	The system call level is not correct.
125	0x0000007D	ERROR_NO_VOLUME_LABEL	The disk has no volume label.
126	0x0000007E	ERROR_MOD_NOT_FOUND	The specified module could not be found.
127	0x0000007F	ERROR_PROC_NOT_FOUND	The specified procedure could not be found.
128	0x00000080	ERROR_WAIT_NO_CHILDREN	There are no child processes to wait for.
129	0x00000081	ERROR_CHILD_NOT_COMPLETE	The %1 application cannot be run in Win32 mode.
130	0x00000082	ERROR_DIRECT_ACCESS_HANDLE	Attempt to use a file handle to an open disk partition for an operation other than raw disk I/O.
131	0x00000083	ERROR_NEGATIVE_SEEK	An attempt was made to move the file pointer before the beginning of the file.
132	0x00000084	ERROR_SEEK_ON_DEVICE	The file pointer cannot be set on the specified device or file.
133	0x00000085	ERROR_IS_JOIN_TARGET	A JOIN or SUBST command cannot be used for a drive that contains previously joined drives.
134	0x00000086	ERROR_IS_JOINED	An attempt was made to use a JOIN or SUBST command on a drive that has already been joined.
135	0x00000087	ERROR_IS_SUBSTED	An attempt was made to use a JOIN or SUBST command on a drive that has already been substituted.
136	0x00000088	ERROR_NOT_JOINED	The system tried to delete the JOIN of a drive that is not joined.
137	0x00000089	ERROR_NOT_SUBSTED	The system tried to delete the substitution of a drive that is not substituted.
138	0x0000008A	ERROR_JOIN_TO_JOIN	The system tried to join a drive to a directory on a joined drive.
139	0x0000008B	ERROR_SUBST_TO_SUBST	The system tried to substitute a drive to a directory on a substituted drive.
140	0x0000008C	ERROR_JOIN_TO_SUBST	The system tried to join a drive to a directory on a substituted drive.
141	0x0000008D	ERROR_SUBST_TO_JOIN	The system tried to SUBST a drive to a directory on a joined drive.

Error			Description
decimal	Hexadecimal	Name	
142	0x0000008E	ERROR_BUSY_DRIVE	The system cannot perform a JOIN or SUBST at this time.
143	0x0000008F	ERROR_SAME_DRIVE	The system cannot join or substitute a drive to or for a directory on the same drive.
144	0x00000090	ERROR_DIR_NOT_ROOT	The directory is not a subdirectory of the root directory.
145	0x00000091	ERROR_DIR_NOT_EMPTY	The directory is not empty.
146	0x00000092	ERROR_IS_SUBST_PATH	The path specified is being used in a substitute.
147	0x00000093	ERROR_IS_JOIN_PATH	Not enough resources are available to process this command.
148	0x00000094	ERROR_PATH_BUSY	The path specified cannot be used at this time.
149	0x00000095	ERROR_IS_SUBST_TARGET	An attempt was made to join or substitute a drive for which a directory on the drive is the target of a previous substitute.
150	0x00000096	ERROR_SYSTEM_TRACE	System trace information was not specified in your CONFIG.SYS file, or tracing is disallowed.
151	0x00000097	ERROR_INVALID_EVENT_COUNT	The number of specified semaphore events for DosMuxSemWait is not correct.
152	0x00000098	ERROR_TOO_MANY_MUXWAITERS	DosMuxSemWait did not execute; too many semaphores are already set.
153	0x00000099	ERROR_INVALID_LIST_FORMAT	The DosMuxSemWait list is not correct.
154	0x0000009A	ERROR_LABEL_TOO_LONG	The volume label you entered exceeds the label character limit of the target file system.
155	0x0000009B	ERROR_TOO_MANY_TCBS	Cannot create another thread.
156	0x0000009C	ERROR_SIGNAL_REFUSED	The recipient process has refused the signal.
157	0x0000009D	ERROR_DISCARDED	The segment is already discarded and cannot be locked.
158	0x0000009E	ERROR_NOT_LOCKED	The segment is already unlocked.
159	0x0000009F	ERROR_BAD_THREADID_ADDR	The address for the thread ID is not correct.
160	0x000000A0	ERROR_BAD_ARGUMENTS	The argument string passed to DosExecPgm is not correct.
161	0x000000A1	ERROR_BAD_PATHNAME	The specified path is invalid.
162	0x000000A2	ERROR_SIGNAL_PENDING	A signal is already pending.
164	0x000000A4	ERROR_MAX_THRDS_REACHED	No more threads can be created in the system.
167	0x000000A7	ERROR_LOCK_FAILED	Unable to lock a region of a file.
170	0x000000AA	ERROR_BUSY	The requested resource is in use.
173	0x000000AD	ERROR_CANCEL_VIOLATION	A lock request was not outstanding for the supplied cancel region.
174	0x000000AE	ERROR_ATOMIC_LOCKS_NOT_SUPPORTED	The file system does not support atomic changes to the lock type.
180	0x000000B4	ERROR_INVALID_SEGMENT_NUMBER	The system detected a segment number that was not correct.
182	0x000000B6	ERROR_INVALID_ORDINAL	The operating system cannot run %1.
183	0x000000B7	ERROR_ALREADY_EXISTS	Cannot create a file when that file already exists.
186	0x000000BA	ERROR_INVALID_FLAG_NUMBER	The flag passed is not correct.
187	0x000000BB	ERROR_SEM_NOT_FOUND	The specified system semaphore name was not found.
188	0x000000BC	ERROR_INVALID_STARTING_CODESEG	The operating system cannot run %1.
189	0x000000BD	ERROR_INVALID_STACKSEG	The operating system cannot run %1.
190	0x000000BE	ERROR_INVALID_MODULETYPE	The operating system cannot run %1.
191	0x000000BF	ERROR_INVALID_EXE_SIGNATURE	Cannot run %1 in Win32 mode.
192	0x000000C0	ERROR_EXE_MARKED_INVALID	The operating system cannot run %1.
193	0x000000C1	ERROR_BAD_EXE_FORMAT	%1 is not a valid Win32 application.
194	0x000000C2	ERROR_ITERATED_DATA_EXCEEDS_64k	The operating system cannot run %1.
195	0x000000C3	ERROR_INVALID_MINALLOCSIZE	The operating system cannot run %1.
196	0x000000C4	ERROR_DYNLINK_FROM_INVALID_RING	The operating system cannot run this application program.
197	0x000000C5	ERROR_IOPL_NOT_ENABLED	The operating system is not presently configured to run this application.
198	0x000000C6	ERROR_INVALID_SEGDPL	The operating system cannot run %1.
199	0x000000C7	ERROR_AUTODATASEG_EXCEEDS_64k	The operating system cannot run this application program.

Error			Description
decimal	Hexadecimal	Name	
200	0x000000C8	ERROR_RING2SEG_MUST_BE_MOVABLE	The code segment cannot be greater than or equal to 64K.
201	0x000000C9	ERROR_RELOC_CHAIN_XEEDS_SEGLIM	The operating system cannot run %1.
202	0x000000CA	ERROR_INFLOOP_IN_RELOC_CHAIN	The operating system cannot run %1.
203	0x000000CB	ERROR_ENVVAR_NOT_FOUND	The system could not find the environment option that was entered.
205	0x000000CD	ERROR_NO_SIGNAL_SENT	No process in the command subtree has a signal handler.
206	0x000000CE	ERROR_FILENAME_EXCED_RANGE	The filename or extension is too long.
207	0x000000CF	ERROR_RING2_STACK_IN_USE	The ring 2 stack is in use.
208	0x000000D0	ERROR_META_EXPANSION_TOO_LONG	The global filename characters, * or ?, are entered incorrectly or too many global filename characters are specified.
209	0x000000D1	ERROR_INVALID_SIGNAL_NUMBER	The signal being posted is not correct.
210	0x000000D2	ERROR_THREAD_1_INACTIVE	The signal handler cannot be set.
212	0x000000D4	ERROR_LOCKED	The segment is locked and cannot be reallocated.
214	0x000000D6	ERROR_TOO_MANY_MODULES	Too many dynamic-link modules are attached to this program or dynamic-link module.
215	0x000000D7	ERROR_NESTING_NOT_ALLOWED	Cannot nest calls to LoadModule.
216	0x000000D8	ERROR_EXE_MACHINE_TYPE_MISMATCH	The image file %1 is valid, but is for a machine type other than the current machine.
230	0x000000E6	ERROR_BAD_PIPE	The pipe state is invalid.
231	0x000000E7	ERROR_PIPE_BUSY	All pipe instances are busy.
232	0x000000E8	ERROR_NO_DATA	The pipe is being closed.
233	0x000000E9	ERROR_PIPE_NOT_CONNECTED	No process is on the other end of the pipe.
234	0x000000EA	ERROR_MORE_DATA	More data is available.
240	0x000000F0	ERROR_VC_DISCONNECTED	The session was canceled.
254	0x000000FE	ERROR_INVALID_EA_NAME	The specified extended attribute name was invalid.
255	0x000000FF	ERROR_EA_LIST_INCONSISTENT	The extended attributes are inconsistent.
258	0x00000102	WAIT_TIMEOUT	The wait operation timed out.
259	0x00000103	ERROR_NO_MORE_ITEMS	No more data is available.
266	0x0000010A	ERROR_CANNOT_COPY	The copy functions cannot be used.
267	0x0000010B	ERROR_DIRECTORY	The directory name is invalid.
275	0x00000113	ERROR_EAS_DIDNT_FIT	The extended attributes did not fit in the buffer.
276	0x00000114	ERROR_EA_FILE_CORRUPT	The extended attribute file on the mounted file system is corrupt.
277	0x00000115	ERROR_EA_TABLE_FULL	The extended attribute table file is full.
278	0x00000116	ERROR_INVALID_EA_HANDLE	The specified extended attribute handle is invalid.
282	0x0000011A	ERROR_EAS_NOT_SUPPORTED	The mounted file system does not support extended attributes.
288	0x00000120	ERROR_NOT_OWNER	Attempt to release mutex not owned by caller.
298	0x0000012A	ERROR_TOO_MANY_POSTS	Too many posts were made to a semaphore.
299	0x0000012B	ERROR_PARTIAL_COPY	Only part of a ReadProcessMemory or WriteProcessMemory request was completed.
300	0x0000012C	ERROR_OPLOCK_NOT_GRANTED	The oplock request is denied.
301	0x0000012D	ERROR_INVALID_OPLOCK_PROTOCOL	An invalid oplock acknowledgment was received by the system.
302	0x0000012E	ERROR_DISK_TOO_FRAGMENTED	The volume is too fragmented to complete this operation.
303	0x0000012F	ERROR_DELETE_PENDING	The file cannot be opened because it is in the process of being deleted.
317	0x0000013D	ERROR_MR_MID_NOT_FOUND	The system cannot find message text for message number 0x%1 in the message file for %2.
487	0x000001E7	ERROR_INVALID_ADDRESS	Attempt to access invalid address.
534	0x00000216	ERROR_ARITHMETIC_OVERFLOW	Arithmetic result exceeded 32 bits.
535	0x00000217	ERROR_PIPE_CONNECTED	There is a process on other end of the pipe.
536	0x00000218	ERROR_PIPE_LISTENING	Waiting for a process to open the other end of the pipe.
994	0x000003E2	ERROR_EA_ACCESS_DENIED	Access to the extended attribute was denied.

Error			Description
decimal	Hexadecimal	Name	
995	0x000003E3	ERROR_OPERATION_ABORTED	The I/O operation has been aborted because of either a thread exit or an application request.
996	0x000003E4	ERROR_IO_INCOMPLETE	Overlapped I/O event is not in a signaled state.
997	0x000003E5	ERROR_IO_PENDING	Overlapped I/O operation is in progress.
998	0x000003E6	ERROR_NOACCESS	Invalid access to memory location.
999	0x000003E7	ERROR_SWAPERROR	Error performing inpage operation.

Error			Description
decimal	Hexadecimal	Name	
1001	0x000003E9	ERROR_STACK_OVERFLOW	Recursion too deep; the stack overflowed.
1002	0x000003EA	ERROR_INVALID_MESSAGE	The window cannot act on the sent message.
1003	0x000003EB	ERROR_CAN_NOT_COMPLETE	Cannot complete this function.
1004	0x000003EC	ERROR_INVALID_FLAGS	Invalid flags.
1005	0x000003ED	ERROR_UNRECOGNIZED_VOLUME	The volume does not contain a recognized file system. Please make sure that all required file system drivers are loaded and that the volume is not corrupted.
1006	0x000003EE	ERROR_FILE_INVALID	The volume for a file has been externally altered so that the opened file is no longer valid.
1007	0x000003EF	ERROR_FULLSCREEN_MODE	The requested operation cannot be performed in full-screen mode.
1008	0x000003F0	ERROR_NO_TOKEN	An attempt was made to reference a token that does not exist.
1009	0x000003F1	ERROR_BADDB	The configuration registry database is corrupt.
1010	0x000003F2	ERROR_BADKEY	The configuration registry key is invalid.
1011	0x000003F3	ERROR_CANTOPEN	The configuration registry key could not be opened.
1012	0x000003F4	ERROR_CANTREAD	The configuration registry key could not be read.
1013	0x000003F5	ERROR_CANTWRITE	The configuration registry key could not be written.
1014	0x000003F6	ERROR_REGISTRY_RECOVERED	One of the files in the registry database had to be recovered by use of a log or alternate copy. The recovery was successful.
1015	0x000003F7	ERROR_REGISTRY_CORRUPT	The registry is corrupted. The structure of one of the files containing registry data is corrupted, or the system's memory image of the file is corrupted, or the file could not be recovered because the alternate copy or log was absent or corrupted.
1016	0x000003F8	ERROR_REGISTRY_IO_FAILED	An I/O operation initiated by the registry failed unrecoverably. The registry could not read in, or write out, or flush, one of the files that contain the system's image of the registry.
1017	0x000003F9	ERROR_NOT_REGISTRY_FILE	The system has attempted to load or restore a file into the registry, but the specified file is not in a registry file format.
1018	0x000003FA	ERROR_KEY_DELETED	Illegal operation attempted on a registry key that has been marked for deletion.
1019	0x000003FB	ERROR_NO_LOG_SPACE	System could not allocate the required space in a registry log.
1020	0x000003FC	ERROR_KEY_HAS_CHILDREN	Cannot create a symbolic link in a registry key that already has subkeys or values.
1021	0x000003FD	ERROR_CHILD_MUST_BE_VOLATILE	Cannot create a stable subkey under a volatile parent key.
1022	0x000003FE	ERROR_NOTIFY_ENUM_DIR	A notify change request is being completed and the information is not being returned in the caller's buffer. The caller now needs to enumerate the files to find the changes.
1051	0x0000041B	ERROR_DEPENDENT_SERVICES_RUNNING	A stop control has been sent to a service that other running services are dependent on.
1052	0x0000041C	ERROR_INVALID_SERVICE_CONTROL	The requested control is not valid for this service.
1053	0x0000041D	ERROR_SERVICE_REQUEST_TIMEOUT	The service did not respond to the start or control request in a timely fashion.
1054	0x0000041E	ERROR_SERVICE_NO_THREAD	A thread could not be created for the service.
1055	0x0000041F	ERROR_SERVICE_DATABASE_LOCKED	The service database is locked.
1056	0x00000420	ERROR_SERVICE_ALREADY_RUNNING	An instance of the service is already running.
1057	0x00000421	ERROR_INVALID_SERVICE_ACCOUNT	The account name is invalid or does not exist, or the password is invalid for the account name specified.
1058	0x00000422	ERROR_SERVICE_DISABLED	The service cannot be started, either because it is disabled or because it has no enabled devices associated with it.
1059	0x00000423	ERROR_CIRCULAR_DEPENDENCY	Circular service dependency was specified.

Error			Description
decimal	Hexadecimal	Name	
1060	0x00000424	ERROR_SERVICE_DOES_NOT_EXIST	The specified service does not exist as an installed service.
1061	0x00000425	ERROR_SERVICE_CANNOT_ACCEPT_CTRL	The service cannot accept control messages at this time.
1062	0x00000426	ERROR_SERVICE_NOT_ACTIVE	The service has not been started.
1063	0x00000427	ERROR_FAILED_SERVICE_CONTROLLER_CONNECT	The service process could not connect to the service controller.
1064	0x00000428	ERROR_EXCEPTION_IN_SERVICE	An exception occurred in the service when handling the control request.
1065	0x00000429	ERROR_DATABASE_DOES_NOT_EXIST	The database specified does not exist.
1066	0x0000042A	ERROR_SERVICE_SPECIFIC_ERROR	The service has returned a service-specific error code.
1067	0x0000042B	ERROR_PROCESS_ABORTED	The process terminated unexpectedly.
1068	0x0000042C	ERROR_SERVICE_DEPENDENCY_FAIL	The dependency service or group failed to start.
1069	0x0000042D	ERROR_SERVICE_LOGON_FAILED	The service did not start due to a logon failure.
1070	0x0000042E	ERROR_SERVICE_START_HANG	After starting, the service hung in a start-pending state.
1071	0x0000042F	ERROR_INVALID_SERVICE_LOCK	The specified service database lock is invalid.
1072	0x00000430	ERROR_SERVICE_MARKED_FOR_DELETE	The specified service has been marked for deletion.
1073	0x00000431	ERROR_SERVICE_EXISTS	The specified service already exists.
1074	0x00000432	ERROR_ALREADY_RUNNING_LKG	The system is currently running with the last-known-good configuration.
1075	0x00000433	ERROR_SERVICE_DEPENDENCY_DELETED	The dependency service does not exist or has been marked for deletion.
1076	0x00000434	ERROR_BOOT_ALREADY_ACCEPTED	The current boot has already been accepted for use as the last-known-good control set.
1077	0x00000435	ERROR_SERVICE_NEVER_STARTED	No attempts to start the service have been made since the last boot.
1078	0x00000436	ERROR_DUPLICATE_SERVICE_NAME	The name is already in use as either a service name or a service display name.
1079	0x00000437	ERROR_DIFFERENT_SERVICE_ACCOUNT	The account specified for this service is different from the account specified for other services running in the same process.
1080	0x00000438	ERROR_CANNOT_DETECT_DRIVER_FAILURE	Failure actions can only be set for Win32 services, not for drivers.
1081	0x00000439	ERROR_CANNOT_DETECT_PROCESS_ABORT	This service runs in the same process as the service control manager. Therefore, the service control manager cannot take action if this service's process terminates unexpectedly.
1082	0x0000043A	ERROR_NO_RECOVERY_PROGRAM	No recovery program has been configured for this service.
1083	0x0000043B	ERROR_SERVICE_NOT_IN_EXE	The executable program that this service is configured to run in does not implement the service.
1084	0x0000043C	ERROR_NOT_SAFEBOOT_SERVICE	This service cannot be started in Safe Mode.

Error			Description
decimal	Hexadecimal	Name	
1100	0x0000044C	ERROR_END_OF_MEDIA	The physical end of the tape has been reached.
1101	0x0000044D	ERROR_FILEMARK_DETECTED	A tape access reached a filemark.
1102	0x0000044E	ERROR_BEGINNING_OF_MEDIA	The beginning of the tape or a partition was encountered.
1103	0x0000044F	ERROR_SETMARK_DETECTED	A tape access reached the end of a set of files.
1104	0x00000450	ERROR_NO_DATA_DETECTED	No more data is on the tape.
1105	0x00000451	ERROR_PARTITION_FAILURE	Tape could not be partitioned.
1106	0x00000452	ERROR_INVALID_BLOCK_LENGTH	When accessing a new tape of a multivolume partition, the current block size is incorrect.
1107	0x00000453	ERROR_DEVICE_NOT_PARTITIONED	Tape partition information could not be found when loading a tape.
1108	0x00000454	ERROR_UNABLE_TO_LOCK_MEDIA	Unable to lock the media eject mechanism.
1109	0x00000455	ERROR_UNABLE_TO_UNLOAD_MEDIA	Unable to unload the media.
1110	0x00000456	ERROR_MEDIA_CHANGED	The media in the drive may have changed.
1111	0x00000457	ERROR_BUS_RESET	The I/O bus was reset.
1112	0x00000458	ERROR_NO_MEDIA_IN_DRIVE	No media in drive.
1113	0x00000459	ERROR_NO_UNICODE_TRANSLATION	No mapping for the Unicode character exists in the target multi-byte code page.
1114	0x0000045A	ERROR_DLL_INIT_FAILED	A dynamic link library (DLL) initialization routine failed.
1115	0x0000045B	ERROR_SHUTDOWN_IN_PROGRESS	A system shutdown is in progress.
1116	0x0000045C	ERROR_NO_SHUTDOWN_IN_PROGRESS	Unable to abort the system shutdown because no shutdown was in progress.
1117	0x0000045D	ERROR_IO_DEVICE	The request could not be performed because of an I/O device error.
1118	0x0000045E	ERROR_SERIAL_NO_DEVICE	No serial device was successfully initialized. The serial driver will unload.
1119	0x0000045F	ERROR_IRQ_BUSY	Unable to open a device that was sharing an interrupt request (IRQ) with other devices. At least one other device that uses that IRQ was already opened.
1120	0x00000460	ERROR_MORE_WRITES	A serial I/O operation was completed by another write to the serial port. (The IOCTL_SERIAL_XOFF_COUNTER reached zero.)
1121	0x00000461	ERROR_COUNTER_TIMEOUT	A serial I/O operation completed because the timeout period expired. (The IOCTL_SERIAL_XOFF_COUNTER did not reach zero.)
1122	0x00000462	ERROR_FLOPPY_ID_MARK_NOT_FOUND	No ID address mark was found on the floppy disk.
1123	0x00000463	ERROR_FLOPPY_WRONG_CYLINDER	Mismatch between the floppy disk sector ID field and the floppy disk controller track address.
1124	0x00000464	ERROR_FLOPPY_UNKNOWN_ERROR	The floppy disk controller reported an error that is not recognized by the floppy disk driver.
1125	0x00000465	ERROR_FLOPPY_BAD_REGISTERS	The floppy disk controller returned inconsistent results in its registers.
1126	0x00000466	ERROR_DISK_RECALIBRATE_FAILED	While accessing the hard disk, a recalibrate operation failed, even after retries.
1127	0x00000467	ERROR_DISK_OPERATION_FAILED	While accessing the hard disk, a disk operation failed even after retries.
1128	0x00000468	ERROR_DISK_RESET_FAILED	While accessing the hard disk, a disk controller reset was needed, but even that failed.
1129	0x00000469	ERROR_EOM_OVERFLOW	Physical end of tape encountered.
1130	0x0000046A	ERROR_NOT_ENOUGH_SERVER_MEMORY	Not enough server storage is available to process this command.
1131	0x0000046B	ERROR_POSSIBLE_DEADLOCK	A potential deadlock condition has been detected.
1132	0x0000046C	ERROR_MAPPED_ALIGNMENT	The base address or the file offset specified does not have the proper alignment.
1140	0x00000474	ERROR_SET_POWER_STATE_VETOED	An attempt to change the system power state was vetoed by another application or driver.
1141	0x00000475	ERROR_SET_POWER_STATE_FAILED	The system BIOS failed an attempt to change the system power state.
1142	0x00000476	ERROR_TOO_MANY_LINKS	An attempt was made to create more links on a file than the file system supports.

Error			Description
decimal	Hexadecimal	Name	
1150	0x0000047E	ERROR_OLD_WIN_VERSION	The specified program requires a newer version of Windows.
1151	0x0000047F	ERROR_APP_WRONG_OS	The specified program is not a Windows or MS-DOS program.
1152	0x00000480	ERROR_SINGLE_INSTANCE_APP	Cannot start more than one instance of the specified program.
1153	0x00000481	ERROR_RMODE_APP	The specified program was written for an earlier version of Windows.
1154	0x00000482	ERROR_INVALID_DLL	One of the library files needed to run this application is damaged.
1155	0x00000483	ERROR_NO_ASSOCIATION	No application is associated with the specified file for this operation.
1156	0x00000484	ERROR_DDE_FAIL	An error occurred in sending the command to the application.
1157	0x00000485	ERROR_DLL_NOT_FOUND	One of the library files needed to run this application cannot be found.
1158	0x00000486	ERROR_NO_MORE_USER_HANDLES	The current process has used all of its system allowance of handles for Window Manager objects.
1159	0x00000487	ERROR_MESSAGE_SYNC_ONLY	The message can be used only with synchronous operations.
1160	0x00000488	ERROR_SOURCE_ELEMENT_EMPTY	The indicated source element has no media.
1161	0x00000489	ERROR_DESTINATION_ELEMENT_FULL	The indicated destination element already contains media.
1162	0x0000048A	ERROR_ILLEGAL_ELEMENT_ADDRESS	The indicated element does not exist.
1163	0x0000048B	ERROR_MAGAZINE_NOT_PRESENT	The indicated element is part of a magazine that is not present.
1164	0x0000048C	ERROR_DEVICE_REINITIALIZATION_NEEDED	The indicated device requires reinitialization due to hardware errors.
1165	0x0000048D	ERROR_DEVICE_REQUIRES_CLEANING	The device has indicated that cleaning is required before further operations are attempted.
1166	0x0000048E	ERROR_DEVICE_DOOR_OPEN	The device has indicated that its door is open.
1167	0x0000048F	ERROR_DEVICE_NOT_CONNECTED	The device is not connected.
1168	0x00000490	ERROR_NOT_FOUND	Element not found.
1169	0x00000491	ERROR_NO_MATCH	There was no match for the specified key in the index.
1170	0x00000492	ERROR_SET_NOT_FOUND	The property set specified does not exist on the object.
1171	0x00000493	ERROR_POINT_NOT_FOUND	The point passed to GetMouseMovePointsEx is not in the buffer.
1172	0x00000494	ERROR_NO_TRACKING_SERVICE	The tracking (workstation) service is not running.
1173	0x00000495	ERROR_NO_VOLUME_ID	The Volume ID could not be found.
1175	0x00000497	ERROR_UNABLE_TO_REMOVE_REPLACED	Unable to remove the file to be replaced.
1176	0x00000498	ERROR_UNABLE_TO_MOVE_REPLACEMENT	Unable to move the replacement file to the file to be replaced. The file to be replaced has retained its original name.
1177	0x00000499	ERROR_UNABLE_TO_MOVE_REPLACEMENT_2	Unable to move the replacement file to the file to be replaced. The file to be replaced has been renamed using the backup name.
1178	0x0000049A	ERROR_JOURNAL_DELETE_IN_PROGRESS	The volume change journal is being deleted.
1179	0x0000049B	ERROR_JOURNAL_NOT_ACTIVE	The volume change journal is not active.
1180	0x0000049C	ERROR_POTENTIAL_FILE_FOUND	A file was found, but it may not be the correct file.
1181	0x0000049D	ERROR_JOURNAL_ENTRY_DELETED	The journal entry has been deleted from the journal.

Error			Description
decimal	Hexadecimal	Name	
1200	0x000004B0	ERROR_BAD_DEVICE	The specified device name is invalid.
1201	0x000004B1	ERROR_CONNECTION_UNAVAIL	The device is not currently connected but it is a remembered connection.
1202	0x000004B2	ERROR_DEVICE_ALREADY_REMEMBERED	The local device name has a remembered connection to another network resource.
1203	0x000004B3	ERROR_NO_NET_OR_BAD_PATH	No network provider accepted the given network path.
1204	0x000004B4	ERROR_BAD_PROVIDER	The specified network provider name is invalid.
1205	0x000004B5	ERROR_CANNOT_OPEN_PROFILE	Unable to open the network connection profile.
1206	0x000004B6	ERROR_BAD_PROFILE	The network connection profile is corrupted.
1207	0x000004B7	ERROR_NOT_CONTAINER	Cannot enumerate a noncontainer.
1208	0x000004B8	ERROR_EXTENDED_ERROR	An extended error has occurred.
1209	0x000004B9	ERROR_INVALID_GROUPNAME	The format of the specified group name is invalid.
1210	0x000004BA	ERROR_INVALID_COMPUTERNAME	The format of the specified computer name is invalid.
1211	0x000004BB	ERROR_INVALID_EVENTNAME	The format of the specified event name is invalid.
1212	0x000004BC	ERROR_INVALID_DOMAINNAME	The format of the specified domain name is invalid.
1213	0x000004BD	ERROR_INVALID_SERVICENAME	The format of the specified service name is invalid.
1214	0x000004BE	ERROR_INVALID_NETNAME	The format of the specified network name is invalid.
1215	0x000004BF	ERROR_INVALID_SHARENAME	The format of the specified share name is invalid.
1216	0x000004C0	ERROR_INVALID_PASSWORDNAME	The format of the specified password is invalid.
1217	0x000004C1	ERROR_INVALID_MESSAGENAME	The format of the specified message name is invalid.
1218	0x000004C2	ERROR_INVALID_MESSAGEDEST	The format of the specified message destination is invalid.
1219	0x000004C3	ERROR_SESSION_CREDENTIAL_CONFLICT	The credentials supplied conflict with an existing set of credentials.
1220	0x000004C4	ERROR_REMOTE_SESSION_LIMIT_EXCEEDED	An attempt was made to establish a session to a network server, but there are already too many sessions established to that server.
1221	0x000004C5	ERROR_DUP_DOMAINNAME	The workgroup or domain name is already in use by another computer on the network.
1222	0x000004C6	ERROR_NO_NETWORK	The network is not present or not started.
1223	0x000004C7	ERROR_CANCELLED	The operation was canceled by the user.
1224	0x000004C8	ERROR_USER_MAPPED_FILE	The requested operation cannot be performed on a file with a user-mapped section open.
1225	0x000004C9	ERROR_CONNECTION_REFUSED	The remote system refused the network connection.
1226	0x000004CA	ERROR_GRACEFUL_DISCONNECT	The network connection was gracefully closed.
1227	0x000004CB	ERROR_ADDRESS_ALREADY_ASSOCIATED	The network transport endpoint already has an address associated with it.
1228	0x000004CC	ERROR_ADDRESS_NOT_ASSOCIATED	An address has not yet been associated with the network endpoint.
1229	0x000004CD	ERROR_CONNECTION_INVALID	An operation was attempted on a nonexistent network connection.
1230	0x000004CE	ERROR_CONNECTION_ACTIVE	An invalid operation was attempted on an active network connection.
1231	0x000004CF	ERROR_NETWORK_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1232	0x000004D0	ERROR_HOST_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1233	0x000004D1	ERROR_PROTOCOL_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1234	0x000004D2	ERROR_PORT_UNREACHABLE	No service is operating at the destination network endpoint on the remote system.
1235	0x000004D3	ERROR_REQUEST_ABORTED	The request was aborted.
1236	0x000004D4	ERROR_CONNECTION_ABORTED	The network connection was aborted by the local system.

Error			Description
decimal	Hexadecimal	Name	
1237	0x000004D5	ERROR_RETRY	The operation could not be completed. A retry should be performed.
1238	0x000004D6	ERROR_CONNECTION_COUNT_LIMIT	A connection to the server could not be made because the limit on the number of concurrent connections for this account has been reached.
1239	0x000004D7	ERROR_LOGIN_TIME_RESTRICTION	Attempting to log in during an unauthorized time of day for this account.
1240	0x000004D8	ERROR_LOGIN_WKSTA_RESTRICTION	The account is not authorized to log in from this station.
1241	0x000004D9	ERROR_INCORRECT_ADDRESS	The network address could not be used for the operation requested.
1242	0x000004DA	ERROR_ALREADY_REGISTERED	The service is already registered.
1243	0x000004DB	ERROR_SERVICE_NOT_FOUND	The specified service does not exist.
1244	0x000004DC	ERROR_NOT_AUTHENTICATED	The operation being requested was not performed because the user has not been authenticated.
1245	0x000004DD	ERROR_NOT_LOGGED_ON	The operation being requested was not performed because the user has not logged on to the network. The specified service does not exist.
1246	0x000004DE	ERROR_CONTINUE	Continue with work in progress.
1247	0x000004DF	ERROR_ALREADY_INITIALIZED	An attempt was made to perform an initialization operation when initialization has already been completed.
1248	0x000004E0	ERROR_NO_MORE_DEVICES	No more local devices.
1249	0x000004E1	ERROR_NO_SUCH_SITE	The specified site does not exist.
1250	0x000004E2	ERROR_DOMAIN_CONTROLLER_EXISTS	A domain controller with the specified name already exists.
1251	0x000004E3	ERROR_ONLY_IF_CONNECTED	This operation is supported only when you are connected to the server.
1252	0x000004E4	ERROR_OVERRIDE_NOCHANGES	The group policy framework should call the extension even if there are no changes.
1253	0x000004E5	ERROR_BAD_USER_PROFILE	The specified user does not have a valid profile.
1254	0x000004E6	ERROR_NOT_SUPPORTED_ON_SBS	This operation is not supported on a Microsoft Small Business Server.
1255	0x000004E7	ERROR_SERVER_SHUTDOWN_IN_PROGRESS	The server machine is shutting down.
1256	0x000004E8	ERROR_HOST_DOWN	The remote system is not available. For information about network troubleshooting, see Windows Help.
1257	0x000004E9	ERROR_NON_ACCOUNT_SID	The security identifier provided is not from an account domain.
1258	0x000004EA	ERROR_NON_DOMAIN_SID	The security identifier provided does not have a domain component.
1259	0x000004EB	ERROR_APPHELP_BLOCK	AppHelp dialog canceled thus preventing the application from starting.
1260	0x000004EC	ERROR_ACCESS_DISABLED_BY_POLICY	Access to the requested resource has been disabled by your administrator.
1261	0x000004ED	ERROR_REG_NAT_CONSUMPTION	A program attempt to use an invalid register value. Normally caused by an uninitialized register. This error is Itanium specific.
1262	0x000004EE	ERROR_CSCSHARE_OFFLINE	The share is currently offline or does not exist.
1300	0x00000514	ERROR_NOT_ALL_ASSIGNED	Not all privileges referenced are assigned to the caller.
1301	0x00000515	ERROR_SOME_NOT_MAPPED	Some mapping between account names and security IDs was not done.
1302	0x00000516	ERROR_NO_QUOTAS_FOR_ACCOUNT	No system quota limits are specifically set for this account.
1303	0x00000517	ERROR_LOCAL_USER_SESSION_KEY	No encryption key is available. A well-known encryption key was returned.
1304	0x00000518	ERROR_NULL_LM_PASSWORD	The password is too complex to be converted to a LAN Manager password. The LAN Manager password returned is a NULL string.
1305	0x00000519	ERROR_UNKNOWN_REVISION	The revision level is unknown.
1306	0x0000051A	ERROR_REVISION_MISMATCH	Indicates two revision levels are incompatible.
1307	0x0000051B	ERROR_INVALID_OWNER	This security ID may not be assigned as the owner of this object.

Error			Description
decimal	Hexadecimal	Name	
1308	0x0000051C	ERROR_INVALID_PRIMARY_GROUP	This security ID may not be assigned as the primary group of an object.
1309	0x0000051D	ERROR_NO_IMPERSONATION_TOKEN	An attempt has been made to operate on an impersonation token by a thread that is not currently impersonating a client.
1310	0x0000051E	ERROR_CANT_DISABLE_MANDATORY	The group may not be disabled.
1311	0x0000051F	ERROR_NO_LOGON_SERVERS	There are currently no logon servers available to service the logon request.
1312	0x00000520	ERROR_NO_SUCH_LOGON_SESSION	A specified logon session does not exist. It may already have been terminated.
1313	0x00000521	ERROR_NO_SUCH_PRIVILEGE	A specified privilege does not exist.
1314	0x00000522	ERROR_PRIVILEGE_NOT_HELD	A required privilege is not held by the client.
1315	0x00000523	ERROR_INVALID_ACCOUNT_NAME	The name provided is not a properly formed account name.
1316	0x00000524	ERROR_USER_EXISTS	The specified user already exists.
1317	0x00000525	ERROR_NO_SUCH_USER	The specified user does not exist.
1318	0x00000526	ERROR_GROUP_EXISTS	The specified group already exists.
1319	0x00000527	ERROR_NO_SUCH_GROUP	The specified group does not exist.
1320	0x00000528	ERROR_MEMBER_IN_GROUP	Either the specified user account is already a member of the specified group, or the specified group cannot be deleted because it contains a member.
1321	0x00000529	ERROR_MEMBER_NOT_IN_GROUP	The specified user account is not a member of the specified group account.
1322	0x0000052A	ERROR_LAST_ADMIN	The last remaining administration account cannot be disabled or deleted.
1323	0x0000052B	ERROR_WRONG_PASSWORD	Unable to update the password. The value provided as the current password is incorrect.
1324	0x0000052C	ERROR_ILL_FORMED_PASSWORD	Unable to update the password. The value provided for the new password contains values that are not allowed in passwords.
1325	0x0000052D	ERROR_PASSWORD_RESTRICTION	Unable to update the password. The value provided for the new password does not meet the length, complexity, or history requirement of the domain.
1326	0x0000052E	ERROR_LOGON_FAILURE	Logon failure: unknown user name or bad password.
1327	0x0000052F	ERROR_ACCOUNT_RESTRICTION	Logon failure: user account restriction.
1328	0x00000530	ERROR_INVALID_LOGON_HOURS	Logon failure: account logon time restriction violation.
1329	0x00000531	ERROR_INVALID_WORKSTATION	Logon failure: user not allowed to log on to this computer.
1330	0x00000532	ERROR_PASSWORD_EXPIRED	Logon failure: the specified account password has expired.
1331	0x00000533	ERROR_ACCOUNT_DISABLED	Logon failure: account currently disabled.
1332	0x00000534	ERROR_NONE_MAPPED	No mapping between account names and security IDs was done.
1333	0x00000535	ERROR_TOO_MANY_LUIDS_REQUESTED	Too many local user identifiers (LUIDs) were requested at one time.
1334	0x00000536	ERROR_LUIDS_EXHAUSTED	No more local user identifiers (LUIDs) are available.
1335	0x00000537	ERROR_INVALID_SUB_AUTHORITY	The subauthority part of a security ID is invalid for this particular use.
1336	0x00000538	ERROR_INVALID_ACL	The access control list (ACL) structure is invalid.
1337	0x00000539	ERROR_INVALID_SID	The security ID structure is invalid.
1338	0x0000053A	ERROR_INVALID_SECURITY_DESCR	The security descriptor structure is invalid.
1340	0x0000053C	ERROR_BAD_INHERITANCE_ACL	The inherited access control list (ACL) or access control entry (ACE) could not be built.
1341	0x0000053D	ERROR_SERVER_DISABLED	The server is currently disabled.
1342	0x0000053E	ERROR_SERVER_NOT_DISABLED	The server is currently enabled.
1343	0x0000053F	ERROR_INVALID_ID_AUTHORITY	The value provided was an invalid value for an identifier authority.
1344	0x00000540	ERROR_ALLOTTED_SPACE_EXCEEDED	No more memory is available for security information updates.

Error			Description
decimal	Hexadecimal	Name	
1345	0x00000541	ERROR_INVALID_GROUP_ATTRIBUTES	The specified attributes are invalid, or incompatible with the attributes for the group as a whole.
1346	0x00000542	ERROR_BAD_IMPERSONATION_LEVEL	Either a required impersonation level was not provided, or the provided impersonation level is invalid.
1347	0x00000543	ERROR_CANT_OPEN_ANONYMOUS	Cannot open an anonymous level security token.
1348	0x00000544	ERROR_BAD_VALIDATION_CLASS	The validation information class requested was invalid.
1349	0x00000545	ERROR_BAD_TOKEN_TYPE	The type of the token is inappropriate for its attempted use.
1350	0x00000546	ERROR_NO_SECURITY_ON_OBJECT	Unable to perform a security operation on an object that has no associated security.
1351	0x00000547	ERROR_CANT_ACCESS_DOMAIN_INFO	Configuration information could not be read from the domain controller, either because the machine is unavailable, or access has been denied.
1352	0x00000548	ERROR_INVALID_SERVER_STATE	The security account manager (SAM) or local security authority (LSA) server was in the wrong state to perform the security operation.
1353	0x00000549	ERROR_INVALID_DOMAIN_STATE	The domain was in the wrong state to perform the security operation.
1354	0x0000054A	ERROR_INVALID_DOMAIN_ROLE	This operation is only allowed for the Primary Domain Controller of the domain.
1355	0x0000054B	ERROR_NO_SUCH_DOMAIN	The specified domain either does not exist or could not be contacted.
1356	0x0000054C	ERROR_DOMAIN_EXISTS	The specified domain already exists.
1357	0x0000054D	ERROR_DOMAIN_LIMIT_EXCEEDED	An attempt was made to exceed the limit on the number of domains per server.
1358	0x0000054E	ERROR_INTERNAL_DB_CORRUPTION	Unable to complete the requested operation because of either a catastrophic media failure or a data structure corruption on the disk.
1359	0x0000054F	ERROR_INTERNAL_ERROR	An internal error occurred.
1360	0x00000550	ERROR_GENERIC_NOT_MAPPED	Generic access types were contained in an access mask which should already be mapped to nongeneric types.
1361	0x00000551	ERROR_BAD_DESCRIPTOR_FORMAT	A security descriptor is not in the right format (absolute or self-relative).
1362	0x00000552	ERROR_NOT_LOGON_PROCESS	The requested action is restricted for use by logon processes only. The calling process has not registered as a logon process.
1363	0x00000553	ERROR_LOGON_SESSION_EXISTS	Cannot start a new logon session with an ID that is already in use.
1364	0x00000554	ERROR_NO_SUCH_PACKAGE	A specified authentication package is unknown.
1365	0x00000555	ERROR_BAD_LOGON_SESSION_STATE	The logon session is not in a state that is consistent with the requested operation.
1366	0x00000556	ERROR_LOGON_SESSION_COLLISION	The logon session ID is already in use.
1367	0x00000557	ERROR_INVALID_LOGON_TYPE	A logon request contained an invalid logon type value.
1368	0x00000558	ERROR_CANNOT_IMPERSONATE	Unable to impersonate using a named pipe until data has been read from that pipe.
1369	0x00000559	ERROR_RXACT_INVALID_STATE	The transaction state of a registry subtree is incompatible with the requested operation.
1370	0x0000055A	ERROR_RXACT_COMMIT_FAILURE	An internal security database corruption has been encountered.
1371	0x0000055B	ERROR_SPECIAL_ACCOUNT	Cannot perform this operation on built-in accounts.
1372	0x0000055C	ERROR_SPECIAL_GROUP	Cannot perform this operation on this built-in special group.
1373	0x0000055D	ERROR_SPECIAL_USER	Cannot perform this operation on this built-in special user.
1374	0x0000055E	ERROR_MEMBERS_PRIMARY_GROUP	The user cannot be removed from a group because the group is currently the user's primary group.
1375	0x0000055F	ERROR_TOKEN_ALREADY_IN_USE	The token is already in use as a primary token.
1376	0x00000560	ERROR_NO_SUCH_ALIAS	The specified local group does not exist.

Error			Description
decimal	Hexadecimal	Name	
1377	0x00000561	ERROR_MEMBER_NOT_IN_ALIAS	The specified account name is not a member of the local group.
1378	0x00000562	ERROR_MEMBER_IN_ALIAS	The specified account name is already a member of the local group.
1379	0x00000563	ERROR_ALIAS_EXISTS	The specified local group already exists.
1380	0x00000564	ERROR_LOGON_NOT_GRANTED	Logon failure: the user has not been granted the requested logon type at this computer.
1381	0x00000565	ERROR_TOO_MANY_SECRETS	The maximum number of secrets that may be stored in a single system has been exceeded.
1382	0x00000566	ERROR_SECRET_TOO_LONG	The length of a secret exceeds the maximum length allowed.
1383	0x00000567	ERROR_INTERNAL_DB_ERROR	The local security authority database contains an internal inconsistency.
1384	0x00000568	ERROR_TOO_MANY_CONTEXT_IDS	During a logon attempt, the user's security context accumulated too many security IDs.
1385	0x00000569	ERROR_LOGON_TYPE_NOT_GRANTED	Logon failure: the user has not been granted the requested logon type at this computer.
1386	0x0000056A	ERROR_NT_CROSS_ENCRYPTION_REQUIRED	A cross-encrypted password is necessary to change a user password.
1387	0x0000056B	ERROR_NO_SUCH_MEMBER	A new member could not be added to or removed from the local group because the member does not exist.
1388	0x0000056C	ERROR_INVALID_MEMBER	A new member could not be added to a local group because the member has the wrong account type.
1389	0x0000056D	ERROR_TOO_MANY_SIDS	Too many security IDs have been specified.
1390	0x0000056E	ERROR_LM_CROSS_ENCRYPTION_REQUIRED	A cross-encrypted password is necessary to change this user password.
1391	0x0000056F	ERROR_NO_INHERITANCE	Indicates an ACL contains no inheritable components.
1392	0x00000570	ERROR_FILE_CORRUPT	The file or directory is corrupted and unreadable.
1393	0x00000571	ERROR_DISK_CORRUPT	The disk structure is corrupted and unreadable.
1394	0x00000572	ERROR_NO_USER_SESSION_KEY	There is no user session key for the specified logon session.
1395	0x00000573	ERROR_LICENSE_QUOTA_EXCEEDED	The service being accessed is licensed for a particular number of connections. No more connections can be made to the service at this time because there are already as many connections as the service can accept.
1396	0x00000574	ERROR_WRONG_TARGET_NAME	Logon Failure: The target account name is incorrect.
1397	0x00000575	ERROR_MUTUAL_AUTH_FAILED	Mutual Authentication failed. The server's password is out of date at the domain controller.
1398	0x00000576	ERROR_TIME_SKEW	There is a time difference between the client and server.
1399	0x00000577	ERROR_CURRENT_DOMAIN_NOT_ALLOWED	This operation can not be performed on the current domain.

Error			Description
decimal	Hexadecimal	Name	
1400	0x00000578	ERROR_INVALID_WINDOW_HANDLE	Invalid window handle.
1401	0x00000579	ERROR_INVALID_MENU_HANDLE	Invalid menu handle.
1402	0x0000057A	ERROR_INVALID_CURSOR_HANDLE	Invalid cursor handle.
1403	0x0000057B	ERROR_INVALID_ACCEL_HANDLE	Invalid accelerator table handle.
1404	0x0000057C	ERROR_INVALID_HOOK_HANDLE	Invalid hook handle.
1405	0x0000057D	ERROR_INVALID_DWP_HANDLE	Invalid handle to a multiple-window position structure.
1406	0x0000057E	ERROR_TLW_WITH_WSCHILD	Cannot create a top-level child window.
1407	0x0000057F	ERROR_CANNOT_FIND_WND_CLASS	Cannot find window class.
1408	0x00000580	ERROR_WINDOW_OF_OTHER_THREAD	Invalid window; it belongs to other thread.
1409	0x00000581	ERROR_HOTKEY_ALREADY_REGISTERED	Hot key is already registered.
1410	0x00000582	ERROR_CLASS_ALREADY_EXISTS	Class already exists.
1411	0x00000583	ERROR_CLASS_DOES_NOT_EXIST	Class does not exist.
1412	0x00000584	ERROR_CLASS_HAS_WINDOWS	Class still has open windows.
1413	0x00000585	ERROR_INVALID_INDEX	Invalid index.
1414	0x00000586	ERROR_INVALID_ICON_HANDLE	Invalid icon handle.
1415	0x00000587	ERROR_PRIVATE_DIALOG_INDEX	Using private DIALOG window words.
1416	0x00000588	ERROR_LISTBOX_ID_NOT_FOUND	The list box identifier was not found.
1417	0x00000589	ERROR_NO_WILDCARD_CHARACTERS	No wildcards were found.
1418	0x0000058A	ERROR_CLIPBOARD_NOT_OPEN	Thread does not have a clipboard open.
1419	0x0000058B	ERROR_HOTKEY_NOT_REGISTERED	Hot key is not registered.
1420	0x0000058C	ERROR_WINDOW_NOT_DIALOG	The window is not a valid dialog window.
1421	0x0000058D	ERROR_CONTROL_ID_NOT_FOUND	Control ID not found.
1422	0x0000058E	ERROR_INVALID_COMBOBOX_MESSAGE	Invalid message for a combo box because it does not have an edit control.
1423	0x0000058F	ERROR_WINDOW_NOT_COMBOBOX	The window is not a combo box.
1424	0x00000590	ERROR_INVALID_EDIT_HEIGHT	Height must be less than 256.
1425	0x00000591	ERROR_DC_NOT_FOUND	Invalid device context (DC) handle.
1426	0x00000592	ERROR_INVALID_HOOK_FILTER	Invalid hook procedure type.
1427	0x00000593	ERROR_INVALID_FILTER_PROC	Invalid hook procedure.
1428	0x00000594	ERROR_HOOK_NEEDS_HMOD	Cannot set nonlocal hook without a module handle.
1429	0x00000595	ERROR_GLOBAL_ONLY_HOOK	This hook procedure can only be set globally.
1430	0x00000596	ERROR_JOURNAL_HOOK_SET	The journal hook procedure is already installed.
1431	0x00000597	ERROR_HOOK_NOT_INSTALLED	The hook procedure is not installed.
1432	0x00000598	ERROR_INVALID_LB_MESSAGE	Invalid message for single-selection list box.
1433	0x00000599	ERROR_SETCOUNT_ON_BAD_LB	LB_SETCOUNT sent to non-lazy list box.
1434	0x0000059A	ERROR_LB_WITHOUT_TABSTOPS	This list box does not support tab stops.
1435	0x0000059B	ERROR_DESTROY_OBJECT_OF_OTHER_THREAD	Cannot destroy object created by another thread.
1436	0x0000059C	ERROR_CHILD_WINDOW_MENU	Child windows cannot have menus.
1437	0x0000059D	ERROR_NO_SYSTEM_MENU	The window does not have a system menu.
1438	0x0000059E	ERROR_INVALID_MSGBOX_STYLE	Invalid message box style.
1439	0x0000059F	ERROR_INVALID_SPI_VALUE	Invalid system-wide (SPI_*) parameter.
1440	0x000005A0	ERROR_SCREEN_ALREADY_LOCKED	Screen already locked.
1441	0x000005A1	ERROR_HWNDS_HAVE_DIFF_PARENT	All handles to windows in a multiple-window position structure must have the same parent.
1442	0x000005A2	ERROR_NOT_CHILD_WINDOW	The window is not a child window.
1443	0x000005A3	ERROR_INVALID_GW_COMMAND	Invalid GW_* command.
1444	0x000005A4	ERROR_INVALID_THREAD_ID	Invalid thread identifier.
1445	0x000005A5	ERROR_NON_MDICHILD_WINDOW	Cannot process a message from a window that is not a multiple document interface (MDI) window.
1446	0x000005A6	ERROR_POPUP_ALREADY_ACTIVE	Popup menu already active.
1447	0x000005A7	ERROR_NO_SCROLLBARS	The window does not have scroll bars.
1448	0x000005A8	ERROR_INVALID_SCROLLBAR_RANGE	Scroll bar range cannot be greater than MAXLONG.
1449	0x000005A9	ERROR_INVALID_SHOWWIN_COMMAND	Cannot show or remove the window in the way specified.

Error			Description
decimal	Hexadecimal	Name	
1450	0x000005AA	ERROR_NO_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1451	0x000005AB	ERROR_NONPAGED_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1452	0x000005AC	ERROR_PAGED_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1453	0x000005AD	ERROR_WORKING_SET_QUOTA	Insufficient quota to complete the requested service.
1454	0x000005AE	ERROR_PAGEFILE_QUOTA	Insufficient quota to complete the requested service.
1455	0x000005AF	ERROR_COMMITMENT_LIMIT	The paging file is too small for this operation to complete.
1456	0x000005B0	ERROR_MENU_ITEM_NOT_FOUND	A menu item was not found.
1457	0x000005B1	ERROR_INVALID_KEYBOARD_HANDLE	Invalid keyboard layout handle.
1458	0x000005B2	ERROR_HOOK_TYPE_NOT_ALLOWED	Hook type not allowed.
1459	0x000005B3	ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION	This operation requires an interactive window station.
1460	0x000005B4	ERROR_TIMEOUT	This operation returned because the timeout period expired.
1461	0x000005B5	ERROR_INVALID_MONITOR_HANDLE	Invalid monitor handle.
1500	0x000005DC	ERROR_EVENTLOG_FILE_CORRUPT	The event log file is corrupted.
1501	0x000005DD	ERROR_EVENTLOG_CANT_START	No event log file could be opened, so the event logging service did not start.
1502	0x000005DE	ERROR_LOG_FILE_FULL	The event log file is full.
1503	0x000005DF	ERROR_EVENTLOG_FILE_CHANGED	The event log file has changed between read operations.

Error			Description
decimal	Hexadecimal	Name	
1601	0x00000641	ERROR_INSTALL_SERVICE_FAILURE	The Windows Installer service could not be accessed. Contact your support personnel to verify that the Windows Installer service is properly registered.
1602	0x00000642	ERROR_INSTALL_USEREXIT	User cancelled installation.
1603	0x00000643	ERROR_INSTALL_FAILURE	Fatal error during installation.
1604	0x00000644	ERROR_INSTALL_SUSPEND	Installation suspended, incomplete.
1605	0x00000645	ERROR_UNKNOWN_PRODUCT	This action is only valid for products that are currently installed.
1606	0x00000646	ERROR_UNKNOWN_FEATURE	Feature ID not registered.
1607	0x00000647	ERROR_UNKNOWN_COMPONENT	Component ID not registered.
1608	0x00000648	ERROR_UNKNOWN_PROPERTY	Unknown property.
1609	0x00000649	ERROR_INVALID_HANDLE_STATE	Handle is in an invalid state.
1610	0x0000064A	ERROR_BAD_CONFIGURATION	The configuration data for this product is corrupt. Contact your support personnel.
1611	0x0000064B	ERROR_INDEX_ABSENT	Component qualifier not present.
1612	0x0000064C	ERROR_INSTALL_SOURCE_ABSENT	The installation source for this product is not available. Verify that the source exists and that you can access it.
1613	0x0000064D	ERROR_INSTALL_PACKAGE_VERSION	This installation package cannot be installed by the Windows Installer service. You must install a Windows service pack that contains a newer version of the Windows Installer service.
1614	0x0000064E	ERROR_PRODUCT_UNINSTALLED	Product is uninstalled.
1615	0x0000064F	ERROR_BAD_QUERY_SYNTAX	SQL query syntax invalid or unsupported.
1616	0x00000650	ERROR_INVALID_FIELD	Record field does not exist.
1617	0x00000651	ERROR_DEVICE_REMOVED	The device has been removed.
1618	0x00000652	ERROR_INSTALL_ALREADY_RUNNING	Another installation is already in progress. Complete that installation before proceeding with this install.
1619	0x00000653	ERROR_INSTALL_PACKAGE_OPEN_FAILED	This installation package could not be opened. Verify that the package exists and that you can access it, or contact the application vendor to verify that this is a valid Windows Installer package.
1620	0x00000654	ERROR_INSTALL_PACKAGE_INVALID	This installation package could not be opened. Contact the application vendor to verify that this is a valid Windows Installer package.
1621	0x00000655	ERROR_INSTALL_UI_FAILURE	There was an error starting the Windows Installer service user interface. Contact your support personnel.
1622	0x00000656	ERROR_INSTALL_LOG_FAILURE	Error opening installation log file. Verify that the specified log file location exists and that you can write to it.
1623	0x00000657	ERROR_INSTALL_LANGUAGE_UNSUPPORTED	The language of this installation package is not supported by your system.
1624	0x00000658	ERROR_INSTALL_TRANSFORM_FAILURE	Error applying transforms. Verify that the specified transform paths are valid.
1625	0x00000659	ERROR_INSTALL_PACKAGE_REJECTED	This installation is forbidden by system policy. Contact your system administrator.
1626	0x0000065A	ERROR_FUNCTION_NOT_CALLED	Function could not be executed.
1627	0x0000065B	ERROR_FUNCTION_FAILED	Function failed during execution.
1628	0x0000065C	ERROR_INVALID_TABLE	Invalid or unknown table specified.
1629	0x0000065D	ERROR_DATATYPE_MISMATCH	Data supplied is of wrong type.
1630	0x0000065E	ERROR_UNSUPPORTED_TYPE	Data of this type is not supported.
1631	0x0000065F	ERROR_CREATE_FAILED	The Windows Installer service failed to start. Contact your support personnel.
1632	0x00000660	ERROR_INSTALL_TEMP_UNWRITABLE	The temp folder is either full or inaccessible. Verify that the temp folder exists and that you can write to it.
1633	0x00000661	ERROR_INSTALL_PLATFORM_UNSUPPORTED	This installation package is not supported by this processor type. Contact your product vendor.
1634	0x00000662	ERROR_INSTALL_NOTUSED	Component not used on this computer.

Error			Description
decimal	Hexadecimal	Name	
1635	0x00000663	ERROR_PATCH_PACKAGE_OPEN_FAILED	This patch package could not be opened. Verify that the patch package exists and that you can access it, or contact the application vendor to verify that this is a valid Windows Installer patch package.
1636	0x00000664	ERROR_PATCH_PACKAGE_INVALID	This patch package could not be opened. Contact the application vendor to verify that this is a valid Windows Installer patch package.
1637	0x00000665	ERROR_PATCH_PACKAGE_UNSUPPORTED.	This patch package cannot be processed by the Windows Installer service. You must install a Windows service pack that contains a newer version of the Windows Installer service.
1638	0x00000666	ERROR_PRODUCT_VERSION	Another version of this product is already installed. Installation of this version cannot continue. To configure or remove the existing version of this product, use Add/Remove Programs on the Control Panel.
1639	0x00000667	ERROR_INVALID_COMMAND_LINE	Invalid command line argument. Consult the Windows Installer SDK for detailed command line help.
1640	0x00000668	ERROR_INSTALL_REMOTE_DISALLOWED	Only administrators have permission to add, remove, or configure server software during a Terminal Services remote session. If you want to install or configure software on the server, contact your network administrator.
1641	0x00000669	ERROR_SUCCESS_REBOOT_INITIATED	The requested operation completed successfully. The system will be restarted so the changes can take effect.
1642	0x0000066A	ERROR_PATCH_TARGET_NOT_FOUND	The upgrade patch cannot be installed by the Windows Installer service because the program to be upgraded may be missing, or the upgrade patch may update a different version of the program. Verify that the program to be upgraded exists on your computer and that you have the correct upgrade patch.
1643	0x0000066B	ERROR_PATCH_PACKAGE_REJECTED	The patch package is not permitted by system policy. It is not signed with an appropriate certificate.
1644	0x0000066C	ERROR_INSTALL_TRANSFORM_REJECTED	One or more customizations are not permitted by system policy. They are not signed with an appropriate certificate.
1700	0x000006A4	RPC_S_INVALID_STRING_BINDING	The string binding is invalid.
1701	0x000006A5	RPC_S_WRONG_KIND_OF_BINDING	The binding handle is not the correct type.
1702	0x000006A6	RPC_S_INVALID_BINDING	The binding handle is invalid.
1703	0x000006A7	RPC_S_PROTSEQ_NOT_SUPPORTED	The RPC protocol sequence is not supported.
1704	0x000006A8	RPC_S_INVALID_RPC_PROTSEQ	The RPC protocol sequence is invalid.
1705	0x000006A9	RPC_S_INVALID_STRING_UUID	The string universal unique identifier (UUID) is invalid.
1706	0x000006AA	RPC_S_INVALID_ENDPOINT_FORMAT	The endpoint format is invalid.
1707	0x000006AB	RPC_S_INVALID_NET_ADDR	The network address is invalid.
1708	0x000006AC	RPC_S_NO_ENDPOINT_FOUND	No endpoint was found.
1709	0x000006AD	RPC_S_INVALID_TIMEOUT	The timeout value is invalid.
1710	0x000006AE	RPC_S_OBJECT_NOT_FOUND	The object universal unique identifier (UUID) was not found.
1711	0x000006AF	RPC_S_ALREADY_REGISTERED	The object universal unique identifier (UUID) has already been registered.
1712	0x000006B0	RPC_S_TYPE_ALREADY_REGISTERED	The type universal unique identifier (UUID) has already been registered.
1713	0x000006B1	RPC_S_ALREADY_LISTENING	The RPC server is already listening.
1714	0x000006B2	RPC_S_NO_PROTSEQS_REGISTERED	No protocol sequences have been registered.
1715	0x000006B3	RPC_S_NOT_LISTENING	The RPC server is not listening.
1716	0x000006B4	RPC_S_UNKNOWN_MGR_TYPE	The manager type is unknown.
1717	0x000006B5	RPC_S_UNKNOWN_IF	The interface is unknown.
1718	0x000006B6	RPC_S_NO_BINDINGS	There are no bindings.
1719	0x000006B7	RPC_S_NO_PROTSEQS	There are no protocol sequences.
1720	0x000006B8	RPC_S_CANT_CREATE_ENDPOINT	The endpoint cannot be created.

Error			Description
decimal	Hexadecimal	Name	
1721	0x000006B9	RPC_S_OUT_OF_RESOURCES	Not enough resources are available to complete this operation.
1722	0x000006BA	RPC_S_SERVER_UNAVAILABLE	The RPC server is unavailable.
1723	0x000006BB	RPC_S_SERVER_TOO_BUSY	The RPC server is too busy to complete this operation.
1724	0x000006BC	RPC_S_INVALID_NETWORK_OPTIONS	The network options are invalid.
1725	0x000006BD	RPC_S_NO_CALL_ACTIVE	There are no remote procedure calls active on this thread.
1726	0x000006BE	RPC_S_CALL_FAILED	The remote procedure call failed.
1727	0x000006BF	RPC_S_CALL_FAILED_DNE	The remote procedure call failed and did not execute.
1728	0x000006C0	RPC_S_PROTOCOL_ERROR	A remote procedure call (RPC) protocol error occurred.
1730	0x000006C2	RPC_S_UNSUPPORTED_TRANS_SYN	The transfer syntax is not supported by the RPC server.
1732	0x000006C4	RPC_S_UNSUPPORTED_TYPE	The universal unique identifier (UUID) type is not supported.
1733	0x000006C5	RPC_S_INVALID_TAG	The tag is invalid.
1734	0x000006C6	RPC_S_INVALID_BOUND	The array bounds are invalid.
1735	0x000006C7	RPC_S_NO_ENTRY_NAME	The binding does not contain an entry name.
1736	0x000006C8	RPC_S_INVALID_NAME_SYNTAX	The name syntax is invalid.
1737	0x000006C9	RPC_S_UNSUPPORTED_NAME_SYNTAX	The name syntax is not supported.
1739	0x000006CB	RPC_S_UUID_NO_ADDRESS	No network address is available to use to construct a universal unique identifier (UUID).
1740	0x000006CC	RPC_S_DUPLICATE_ENDPOINT	The endpoint is a duplicate.
1741	0x000006CD	RPC_S_UNKNOWN_AUTHN_TYPE	The authentication type is unknown.
1742	0x000006CE	RPC_S_MAX_CALLS_TOO_SMALL	The maximum number of calls is too small.
1743	0x000006CF	RPC_S_STRING_TOO_LONG	The string is too long.
1744	0x000006D0	RPC_S_PROTSEQ_NOT_FOUND	The RPC protocol sequence was not found.
1745	0x000006D1	RPC_S_PROCNUM_OUT_OF_RANGE	The procedure number is out of range.
1746	0x000006D2	RPC_S_BINDING_HAS_NO_AUTH	The binding does not contain any authentication information.
1747	0x000006D3	RPC_S_UNKNOWN_AUTHN_SERVICE	The authentication service is unknown.
1748	0x000006D4	RPC_S_UNKNOWN_AUTHN_LEVEL	The authentication level is unknown.
1749	0x000006D5	RPC_S_INVALID_AUTH_IDENTITY	The security context is invalid.
1750	0x000006D6	RPC_S_UNKNOWN_AUTHZ_SERVICE	The authorization service is unknown.
1751	0x000006D7	EPT_S_INVALID_ENTRY	The entry is invalid.
1752	0x000006D8	EPT_S_CANT_PERFORM_OP	The server endpoint cannot perform the operation.
1753	0x000006D9	EPT_S_NOT_REGISTERED	There are no more endpoints available from the endpoint mapper.
1754	0x000006DA	RPC_S_NOTHING_TO_EXPORT	No interfaces have been exported.
1755	0x000006DB	RPC_S_INCOMPLETE_NAME	The entry name is incomplete.
1756	0x000006DC	RPC_S_INVALID_VERS_OPTION	The version option is invalid.
1757	0x000006DD	RPC_S_NO_MORE_MEMBERS	There are no more members.
1758	0x000006DE	RPC_S_NOT_ALL_OBJS_UNEXPORTED	There is nothing to unexport.
1759	0x000006DF	RPC_S_INTERFACE_NOT_FOUND	The interface was not found.
1760	0x000006E0	RPC_S_ENTRY_ALREADY_EXISTS	The entry already exists.
1761	0x000006E1	RPC_S_ENTRY_NOT_FOUND	The entry is not found.
1762	0x000006E2	RPC_S_NAME_SERVICE_UNAVAILABLE	The name service is unavailable.
1763	0x000006E3	RPC_S_INVALID_NAF_ID	The network address family is invalid.
1764	0x000006E4	RPC_S_CANNOT_SUPPORT	The requested operation is not supported.
1765	0x000006E5	RPC_S_NO_CONTEXT_AVAILABLE	No security context is available to allow impersonation.
1766	0x000006E6	RPC_S_INTERNAL_ERROR	An internal error occurred in a remote procedure call (RPC).
1767	0x000006E7	RPC_S_ZERO_DIVIDE	The RPC server attempted an integer division by zero.
1768	0x000006E8	RPC_S_ADDRESS_ERROR	An addressing error occurred in the RPC server.
1769	0x000006E9	RPC_S_FP_DIV_ZERO	A floating-point operation at the RPC server caused a division by zero.

Error			Description
decimal	Hexadecimal	Name	
1770	0x000006EA	RPC_S_FP_UNDERFLOW	A floating-point underflow occurred at the RPC server.
1771	0x000006EB	RPC_S_FP_OVERFLOW	A floating-point overflow occurred at the RPC server.
1772	0x000006EC	RPC_X_NO_MORE_ENTRIES	The list of RPC servers available for the binding of auto handles has been exhausted.
1773	0x000006ED	RPC_X_SS_CHAR_TRANS_OPEN_FAIL	Unable to open the character translation table file.
1774	0x000006EE	RPC_X_SS_CHAR_TRANS_SHORT_FILE	The file containing the character translation table has fewer than 512 bytes.
1775	0x000006EF	RPC_X_SS_IN_NULL_CONTEXT	A null context handle was passed from the client to the host during a remote procedure call.
1777	0x000006F1	RPC_X_SS_CONTEXT_DAMAGED	The context handle changed during a remote procedure call.
1778	0x000006F2	RPC_X_SS_HANDLES_MISMATCH	The binding handles passed to a remote procedure call do not match.
1779	0x000006F3	RPC_X_SS_CANNOT_GET_CALL_HANDLE	The stub is unable to get the remote procedure call handle.
1780	0x000006F4	RPC_X_NULL_REF_POINTER	A null reference pointer was passed to the stub.
1781	0x000006F5	RPC_X_ENUM_VALUE_OUT_OF_RANGE	The enumeration value is out of range.
1782	0x000006F6	RPC_X_BYTE_COUNT_TOO_SMALL	The byte count is too small.
1783	0x000006F7	RPC_X_BAD_STUB_DATA	The stub received bad data.
1784	0x000006F8	ERROR_INVALID_USER_BUFFER	The supplied user buffer is not valid for the requested operation.
1785	0x000006F9	ERROR_UNRECOGNIZED_MEDIA	The disk media is not recognized. It may not be formatted.
1786	0x000006FA	ERROR_NO_TRUST_LSA_SECRET	The workstation does not have a trust secret.
1787	0x000006FB	ERROR_NO_TRUST_SAM_ACCOUNT	The security database on the server does not have a computer account for this workstation trust relationship.
1788	0x000006FC	ERROR_TRUSTED_DOMAIN_FAILURE	The trust relationship between the primary domain and the trusted domain failed.
1789	0x000006FD	ERROR_TRUSTED_RELATIONSHIP_FAILURE	The trust relationship between this workstation and the primary domain failed.
1790	0x000006FE	ERROR_TRUST_FAILURE	The network logon failed.
1791	0x000006FF	RPC_S_CALL_IN_PROGRESS	A remote procedure call is already in progress for this thread.
1792	0x00000700	ERROR_NETLOGON_NOT_STARTED	An attempt was made to logon, but the network logon service was not started.
1793	0x00000701	ERROR_ACCOUNT_EXPIRED	The user's account has expired.
1794	0x00000702	ERROR_REDIRECTOR_HAS_OPEN_HANDLES	The redirector is in use and cannot be unloaded.
1795	0x00000703	ERROR_PRINTER_DRIVER_ALREADY_INSTALLED	The specified printer driver is already installed.
1796	0x00000704	ERROR_UNKNOWN_PORT	The specified port is unknown.
1797	0x00000705	ERROR_UNKNOWN_PRINTER_DRIVER	The printer driver is unknown.
1798	0x00000706	ERROR_UNKNOWN_PRINTPROCESSOR	The print processor is unknown.
1799	0x00000707	ERROR_INVALID_SEPARATOR_FILE	The specified separator file is invalid.

Error			Description
decimal	Hexadecimal	Name	
1800	0x00000708	ERROR_INVALID_PRIORITY	The specified priority is invalid.
1801	0x00000709	ERROR_INVALID_PRINTER_NAME	The printer name is invalid.
1802	0x0000070A	ERROR_PRINTER_ALREADY_EXISTS	The printer already exists.
1803	0x0000070B	ERROR_INVALID_PRINTER_COMMAND	The printer command is invalid.
1804	0x0000070C	ERROR_INVALID_DATATYPE	The specified datatype is invalid.
1805	0x0000070D	ERROR_INVALID_ENVIRONMENT	The environment specified is invalid.
1806	0x0000070E	RPC_S_NO_MORE_BINDINGS	There are no more bindings.
1807	0x0000070F	ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT	The account used is an interdomain trust account. Use your global user account or local user account to access this server.
1808	0x00000710	ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT	The account used is a computer account. Use your global user account or local user account to access this server.
1809	0x00000711	ERROR_NOLOGON_SERVER_TRUST_ACCOUNT	The account used is a server trust account. Use your global user account or local user account to access this server.
1810	0x00000712	ERROR_DOMAIN_TRUST_INCONSISTENT	The name or security ID (SID) of the domain specified is inconsistent with the trust information for that domain.
1811	0x00000713	ERROR_SERVER_HAS_OPEN_HANDLES	The server is in use and cannot be unloaded.
1812	0x00000714	ERROR_RESOURCE_DATA_NOT_FOUND	The specified image file did not contain a resource section.
1813	0x00000715	ERROR_RESOURCE_TYPE_NOT_FOUND	The specified resource type cannot be found in the image file.
1814	0x00000716	ERROR_RESOURCE_NAME_NOT_FOUND	The specified resource name cannot be found in the image file.
1815	0x00000717	ERROR_RESOURCE_LANG_NOT_FOUND	The specified resource language ID cannot be found in the image file.
1816	0x00000718	ERROR_NOT_ENOUGH_QUOTA	Not enough quota is available to process this command.
1817	0x00000719	RPC_S_NO_INTERFACES	No interfaces have been registered.
1818	0x0000071A	RPC_S_CALL_CANCELLED	The remote procedure call was cancelled.
1819	0x0000071B	RPC_S_BINDING_INCOMPLETE	The binding handle does not contain all required information.
1820	0x0000071C	RPC_S_COMM_FAILURE	A communications failure occurred during a remote procedure call.
1821	0x0000071D	RPC_S_UNSUPPORTED_AUTHN_LEVEL	The requested authentication level is not supported.
1822	0x0000071E	RPC_S_NO_PRINC_NAME	No principal name registered.
1823	0x0000071F	RPC_S_NOT_RPC_ERROR	The error specified is not a valid Windows RPC error code.
1824	0x00000720	RPC_S_UUID_LOCAL_ONLY	A UUID that is valid only on this computer has been allocated.
1825	0x00000721	RPC_S_SEC_PKG_ERROR	A security package specific error occurred.
1826	0x00000722	RPC_S_NOT_CANCELLED	Thread is not canceled.
1827	0x00000723	RPC_X_INVALID_ES_ACTION	Invalid operation on the encoding/decoding handle.
1828	0x00000724	RPC_X_WRONG_ES_VERSION	Incompatible version of the serializing package.
1829	0x00000725	RPC_X_WRONG_STUB_VERSION	Incompatible version of the RPC stub.
1830	0x00000726	RPC_X_INVALID_PIPE_OBJECT	The RPC pipe object is invalid or corrupted.
1831	0x00000727	RPC_X_WRONG_PIPE_ORDER	An invalid operation was attempted on an RPC pipe object.
1832	0x00000728	RPC_X_WRONG_PIPE_VERSION	Unsupported RPC pipe version.
1898	0x0000076A	RPC_S_GROUP_MEMBER_NOT_FOUND	The group member was not found.
1899	0x0000076B	EPT_S_CANT_CREATE	The endpoint mapper database entry could not be created.
1900	0x0000076C	RPC_S_INVALID_OBJECT	The object universal unique identifier (UUID) is the nil UUID.
1901	0x0000076D	ERROR_INVALID_TIME	The specified time is invalid.
1902	0x0000076E	ERROR_INVALID_FORM_NAME	The specified form name is invalid.
1903	0x0000076F	ERROR_INVALID_FORM_SIZE	The specified form size is invalid.
1904	0x00000770	ERROR_ALREADY_WAITING	The specified printer handle is already being waited on

Error			Description
decimal	Hexadecimal	Name	
1905	0x00000771	ERROR_PRINTER_DELETED	The specified printer has been deleted.
1906	0x00000772	ERROR_INVALID_PRINTER_STATE	The state of the printer is invalid.
1907	0x00000773	ERROR_PASSWORD_MUST_CHANGE	The user's password must be changed before logging on the first time.
1908	0x00000774	ERROR_DOMAIN_CONTROLLER_NOT_FOUND	Could not find the domain controller for this domain.
1909	0x00000775	ERROR_ACCOUNT_LOCKED_OUT	The referenced account is currently locked out and may not be logged on to.
1910	0x00000776	OR_INVALID_OXID	The object exporter specified was not found.
1911	0x00000777	OR_INVALID_OID	The object specified was not found.
1912	0x00000778	OR_INVALID_SET	The object resolver set specified was not found.
1913	0x00000779	RPC_S_SEND_INCOMPLETE	Some data remains to be sent in the request buffer.
1914	0x0000077A	RPC_S_INVALID_ASYNC_HANDLE	Invalid asynchronous remote procedure call handle.
1915	0x0000077B	RPC_S_INVALID_ASYNC_CALL	Invalid asynchronous RPC call handle for this operation.
1916	0x0000077C	RPC_X_PIPE_CLOSED	The RPC pipe object has already been closed.
1917	0x0000077D	RPC_X_PIPE_DISCIPLINE_ERROR	The RPC call completed before all pipes were processed.
1918	0x0000077E	RPC_X_PIPE_EMPTY	No more data is available from the RPC pipe.
1919	0x0000077F	ERROR_NO_SITENAME	No site name is available for this machine.
1920	0x00000780	ERROR_CANT_ACCESS_FILE	The file can not be accessed by the system.
1921	0x00000781	ERROR_CANT_RESOLVE_FILENAME	The name of the file cannot be resolved by the system.
1922	0x00000782	RPC_S_ENTRY_TYPE_MISMATCH	The entry is not of the expected type.
1923	0x00000783	RPC_S_NOT_ALL_OBJS_EXPORTED	Not all object UUIDs could be exported to the specified entry.
1924	0x00000784	RPC_S_INTERFACE_NOT_EXPORTED	Interface could not be exported to the specified entry.
1925	0x00000785	RPC_S_PROFILE_NOT_ADDED	The specified profile entry could not be added.
1926	0x00000786	RPC_S_PRF_ELT_NOT_ADDED	The specified profile element could not be added.
1927	0x00000787	RPC_S_PRF_ELT_NOT_REMOVED	The specified profile element could not be removed.
1928	0x00000788	RPC_S_GRP_ELT_NOT_ADDED	The group element could not be added.
1929	0x00000789	RPC_S_GRP_ELT_NOT_REMOVED	The group element could not be removed.
1930	0x0000078A	ERROR_KM_DRIVER_BLOCKED	The printer driver is not compatible with a policy enabled on your computer that blocks NT 4.0 drivers.

Error			Description
decimal	Hexadecimal	Name	
2000	0x000007D0	ERROR_INVALID_PIXEL_FORMAT	The pixel format is invalid.
2001	0x000007D1	ERROR_BAD_DRIVER	The specified driver is invalid.
2002	0x000007D2	ERROR_INVALID_WINDOW_STYLE	The window style or class attribute is invalid for this operation.
2003	0x000007D3	ERROR_METAFILE_NOT_SUPPORTED	The requested metafile operation is not supported.
2004	0x000007D4	ERROR_TRANSFORM_NOT_SUPPORTED	The requested transformation operation is not supported.
2005	0x000007D5	ERROR_CLIPPING_NOT_SUPPORTED	The requested clipping operation is not supported.
2010	0x000007DA	ERROR_INVALID_CMM	The specified color management module is invalid.
2011	0x000007DB	ERROR_INVALID_PROFILE	The specified color profile is invalid.
2012	0x000007DC	ERROR_TAG_NOT_FOUND	The specified tag was not found.
2013	0x000007DD	ERROR_TAG_NOT_PRESENT	A required tag is not present.
2014	0x000007DE	ERROR_DUPLICATE_TAG	The specified tag is already present.
2015	0x000007DF	ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE	The specified color profile is not associated with any device.
2016	0x000007E0	ERROR_PROFILE_NOT_FOUND	The specified color profile was not found.
2017	0x000007E1	ERROR_INVALID_COLORSPACE	The specified color space is invalid.
2018	0x000007E2	ERROR_ICM_NOT_ENABLED	Image Color Management is not enabled.
2019	0x000007E3	ERROR_DELETING_ICM_XFORM	There was an error while deleting the color transform.
2020	0x000007E4	ERROR_INVALID_TRANSFORM	The specified color transform is invalid.
2021	0x000007E5	ERROR_COLORSPACE_MISMATCH	The specified transform does not match the bitmap's color space.
2022	0x000007E6	ERROR_INVALID_COLORINDEX	The specified named color index is not present in the profile.
2108	0x0000083C	ERROR_CONNECTED_OTHER_PASSWORD	The network connection was made successfully, but the user had to be prompted for a password other than the one originally specified.
2202	0x0000089A	ERROR_BAD_USERNAME	The specified username is invalid.
2250	0x000008CA	ERROR_NOT_CONNECTED	This network connection does not exist.
2401	0x00000961	ERROR_OPEN_FILES	This network connection has files open or requests pending.
2402	0x00000962	ERROR_ACTIVE_CONNECTIONS	Active connections still exist.
2404	0x00000964	ERROR_DEVICE_IN_USE	The device is in use by an active process and cannot be disconnected.
2500	0x000009C4	ERROR_PKINIT_FAILURE	The kerberos protocol encountered an error while validating the KDC certificate during smartcard logon.
2501	0x000009C5	ERROR_SMARTCARD_SUBSYSTEM_FAILURE	The kerberos protocol encountered an error while attempting to utilize the smartcard subsystem.

Error			Description
decimal	Hexadecimal	Name	
3000	0x00000BB8	ERROR_UNKNOWN_PRINT_MONITOR	The specified print monitor is unknown.
3001	0x00000BB9	ERROR_PRINTER_DRIVER_IN_USE	The specified printer driver is currently in use.
3002	0x00000BBA	ERROR_SPOOL_FILE_NOT_FOUND	The spool file was not found.
3003	0x00000BBB	ERROR_SPL_NO_STARTDOC	A StartDocPrinter call was not issued.
3004	0x00000BBC	ERROR_SPL_NO_ADDJOB	An AddJob call was not issued.
3005	0x00000BBD	ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED	The specified print processor has already been installed.
3006	0x00000BBE	ERROR_PRINT_MONITOR_ALREADY_INSTALLED	The specified print monitor has already been installed.
3007	0x00000BBF	ERROR_INVALID_PRINT_MONITOR	The specified print monitor does not have the required functions.
3008	0x00000BC0	ERROR_PRINT_MONITOR_IN_USE	The specified print monitor is currently in use.
3009	0x00000BC1	ERROR_PRINTER_HAS_JOBS_QUEUED	The requested operation is not allowed when there are jobs queued to the printer.
3010	0x00000BC2	ERROR_SUCCESS_REBOOT_REQUIRED	The requested operation is successful. Changes will not be effective until the system is rebooted.
3011	0x00000BC3	ERROR_SUCCESS_RESTART_REQUIRED	The requested operation is successful. Changes will not be effective until the service is restarted.
3012	0x00000BC4	ERROR_PRINTER_NOT_FOUND	No printers were found.
4000	0x00000FA0	ERROR_WINS_INTERNAL	WINS encountered an error while processing the command.
4001	0x00000FA1	ERROR_CAN_NOT_DEL_LOCAL_WINS	The local WINS can not be deleted.
4002	0x00000FA2	ERROR_STATIC_INIT	The importation from the file failed.
4003	0x00000FA3	ERROR_INC_BACKUP	The backup failed. Was a full backup done before?
4004	0x00000FA4	ERROR_FULL_BACKUP	The backup failed. Check the directory to which you are backing the database.
4005	0x00000FA5	ERROR_REC_NON_EXISTENT	The name does not exist in the WINS database.
4006	0x00000FA6	ERROR_RPL_NOT_ALLOWED	Replication with a nonconfigured partner is not allowed.
4100	0x00001004	ERROR_DHCP_ADDRESS_CONFLICT	The DHCP client has obtained an IP address that is already in use on the network. The local interface will be disabled until the DHCP client can obtain a new address.
4200	0x00001068	ERROR_WMI_GUID_NOT_FOUND	The GUID passed was not recognized as valid by a WMI data provider.
4201	0x00001069	ERROR_WMI_INSTANCE_NOT_FOUND	The instance name passed was not recognized as valid by a WMI data provider.
4202	0x0000106A	ERROR_WMI_ITEMID_NOT_FOUND	The data item ID passed was not recognized as valid by a WMI data provider.
4203	0x0000106B	ERROR_WMI_TRY_AGAIN	The WMI request could not be completed and should be retried.
4204	0x0000106C	ERROR_WMI_DP_NOT_FOUND	The WMI data provider could not be located.
4205	0x0000106D	ERROR_WMI_UNRESOLVED_INSTANCE_REF	The WMI data provider references an instance set that has not been registered.
4206	0x0000106E	ERROR_WMI_ALREADY_ENABLED	The WMI data block or event notification has already been enabled.
4207	0x0000106F	ERROR_WMI_GUID_DISCONNECTED	The WMI data block is no longer available.
4208	0x00001070	ERROR_WMI_SERVER_UNAVAILABLE	The WMI data service is not available.
4209	0x00001071	ERROR_WMI_DP_FAILED	The WMI data provider failed to carry out the request.
4210	0x00001072	ERROR_WMI_INVALID_MOF	The WMI MOF information is not valid.
4211	0x00001073	ERROR_WMI_INVALID_REGINFO	The WMI registration information is not valid.
4212	0x00001074	ERROR_WMI_ALREADY_DISABLED	The WMI data block or event notification has already been disabled.
4213	0x00001075	ERROR_WMI_READ_ONLY	The WMI data item or data block is read only.
4214	0x00001076	ERROR_WMI_SET_FAILURE	The WMI data item or data block could not be changed.
4300	0x000010CC	ERROR_INVALID_MEDIA	The media identifier does not represent a valid medium.
4301	0x000010CD	ERROR_INVALID_LIBRARY	The library identifier does not represent a valid library.

Error			Description
decimal	Hexadecimal	Name	
4302	0x000010CE	ERROR_INVALID_MEDIA_POOL	The media pool identifier does not represent a valid media pool.
4303	0x000010CF	ERROR_DRIVE_MEDIA_MISMATCH	The drive and medium are not compatible or exist in different libraries.
4304	0x000010D0	ERROR_MEDIA_OFFLINE	The medium currently exists in an offline library and must be online to perform this operation.
4305	0x000010D1	ERROR_LIBRARY_OFFLINE	The operation cannot be performed on an offline library.
4306	0x000010D2	ERROR_EMPTY	The library, drive, or media pool is empty.
4307	0x000010D3	ERROR_NOT_EMPTY	The library, drive, or media pool must be empty to perform this operation.
4308	0x000010D4	ERROR_MEDIA_UNAVAILABLE	No media is currently available in this media pool or library.
4309	0x000010D5	ERROR_RESOURCE_DISABLED	A resource required for this operation is disabled.
4310	0x000010D6	ERROR_INVALID_CLEANER	The media identifier does not represent a valid cleaner.
4311	0x000010D7	ERROR_UNABLE_TO_CLEAN	The drive cannot be cleaned or does not support cleaning.
4312	0x000010D8	ERROR_OBJECT_NOT_FOUND	The object identifier does not represent a valid object.
4313	0x000010D9	ERROR_DATABASE_FAILURE	Unable to read from or write to the database.
4314	0x000010DA	ERROR_DATABASE_FULL	The database is full.
4315	0x000010DB	ERROR_MEDIA_INCOMPATIBLE	The medium is not compatible with the device or media pool.
4316	0x000010DC	ERROR_RESOURCE_NOT_PRESENT	The resource required for this operation does not exist.
4317	0x000010DD	ERROR_INVALID_OPERATION	The operation identifier is not valid.
4318	0x000010DE	ERROR_MEDIA_NOT_AVAILABLE	The media is not mounted or ready for use.
4319	0x000010DF	ERROR_DEVICE_NOT_AVAILABLE	The device is not ready for use.
4320	0x000010E0	ERROR_REQUEST_REFUSED	The operator or administrator has refused the request.
4321	0x000010E1	ERROR_INVALID_DRIVE_OBJECT	The drive identifier does not represent a valid drive.
4322	0x000010E2	ERROR_LIBRARY_FULL	Library is full. No slot is available for use.
4323	0x000010E3	ERROR_MEDIUM_NOT_ACCESSIBLE	The transport cannot access the medium.
4324	0x000010E4	ERROR_UNABLE_TO_LOAD_MEDIUM	Unable to load the medium into the drive.
4325	0x000010E5	ERROR_UNABLE_TO_INVENTORY_DRIVE	Unable to retrieve status about the drive.
4326	0x000010E6	ERROR_UNABLE_TO_INVENTORY_SLOT	Unable to retrieve status about the slot.
4327	0x000010E7	ERROR_UNABLE_TO_INVENTORY_TRANSPORT	Unable to retrieve status about the transport.
4328	0x000010E8	ERROR_TRANSPORT_FULL	Cannot use the transport because it is already in use.
4329	0x000010E9	ERROR_CONTROLLING_IEPORT	Unable to open or close the inject/eject port.
4330	0x000010EA	ERROR_UNABLE_TO_EJECT_MOUNTED_MEDIA	Unable to eject the media because it is in a drive.
4331	0x000010EB	ERROR_CLEANER_SLOT_SET	A cleaner slot is already reserved.
4332	0x000010EC	ERROR_CLEANER_SLOT_NOT_SET	A cleaner slot is not reserved.
4333	0x000010ED	ERROR_CLEANER_CARTRIDGE_SPENT	The cleaner cartridge has performed the maximum number of drive cleanings.
4334	0x000010EE	ERROR_UNEXPECTED_OMID	Unexpected on-medium identifier.
4335	0x000010EF	ERROR_CANT_DELETE_LAST_ITEM	The last remaining item in this group or resource cannot be deleted.
4336	0x000010F0	ERROR_MESSAGE_EXCEEDS_MAX_SIZE	The message provided exceeds the maximum size allowed for this parameter.
4337	0x000010F1	ERROR_VOLUME_CONTAINS_SYS_FILES	The volume contains system or paging files.
4338	0x000010F2	ERROR_INDIGENOUS_TYPE	The media type cannot be removed from this library since at least one drive in the library reports it can support this media type.
4339	0x000010F3	ERROR_NO_SUPPORTING_DRIVES	This offline media cannot be mounted on this system since no enabled drives are present which can be used.
4340	0x000010F4	ERROR_CLEANER_CARTRIDGE_INSTALLED	A cleaner cartridge is present in the tape library.
4350	0x000010FE	ERROR_FILE_OFFLINE	The remote storage service was not able to recall the file.

Error			Description
decimal	Hexadecimal	Name	
4351	0x000010FF	ERROR_REMOTE_STORAGE_NOT_ACTIVE	The remote storage service is not operational at this time.
4352	0x00001100	ERROR_REMOTE_STORAGE_MEDIA_ERROR	The remote storage service encountered a media error.
4390	0x00001126	ERROR_NOT_A_REPARSE_POINT	The file or directory is not a reparse point.
4391	0x00001127	ERROR_REPARSE_ATTRIBUTE_CONFLICT	The reparse point attribute cannot be set because it conflicts with an existing attribute.
4392	0x00001128	ERROR_INVALID_REPARSE_DATA	The data present in the reparse point buffer is invalid.
4393	0x00001129	ERROR_REPARSE_TAG_INVALID	The tag present in the reparse point buffer is invalid.
4394	0x0000112A	ERROR_REPARSE_TAG_MISMATCH	There is a mismatch between the tag specified in the request and the tag present in the reparse point.
4500	0x00001194	ERROR_VOLUME_NOT_SIS_ENABLED	Single Instance Storage is not available on this volume.

Error			Description
decimal	Hexadecimal	Name	
5001	0x00001389	ERROR_DEPENDENT_RESOURCE_EXISTS	The cluster resource cannot be moved to another group because other resources are dependent on it.
5002	0x0000138A	ERROR_DEPENDENCY_NOT_FOUND	The cluster resource dependency cannot be found.
5003	0x0000138B	ERROR_DEPENDENCY_ALREADY_EXISTS	The cluster resource cannot be made dependent on the specified resource because it is already dependent.
5004	0x0000138C	ERROR_RESOURCE_NOT_ONLINE	The cluster resource is not online.
5005	0x0000138D	ERROR_HOST_NODE_NOT_AVAILABLE	A cluster node is not available for this operation.
5006	0x0000138E	ERROR_RESOURCE_NOT_AVAILABLE	The cluster resource is not available.
5007	0x0000138F	ERROR_RESOURCE_NOT_FOUND	The cluster resource could not be found.
5008	0x00001390	ERROR_SHUTDOWN_CLUSTER	The cluster is being shut down.
5009	0x00001391	ERROR_CANT_EVICT_ACTIVE_NODE	A cluster node cannot be evicted from the cluster unless the node is down.
5010	0x00001392	ERROR_OBJECT_ALREADY_EXISTS	The object already exists.
5011	0x00001393	ERROR_OBJECT_IN_LIST	The object is already in the list.
5012	0x00001394	ERROR_GROUP_NOT_AVAILABLE	The cluster group is not available for any new requests.
5013	0x00001395	ERROR_GROUP_NOT_FOUND	The cluster group could not be found.
5014	0x00001396	ERROR_GROUP_NOT_ONLINE	The operation could not be completed because the cluster group is not online.
5015	0x00001397	ERROR_HOST_NODE_NOT_RESOURCE_OWNER	The cluster node is not the owner of the resource.
5016	0x00001398	ERROR_HOST_NODE_NOT_GROUP_OWNER	The cluster node is not the owner of the group.
5017	0x00001399	ERROR_RESMON_CREATE_FAILED	The cluster resource could not be created in the specified resource monitor.
5018	0x0000139A	ERROR_RESMON_ONLINE_FAILED	The cluster resource could not be brought online by the resource monitor.
5019	0x0000139B	ERROR_RESOURCE_ONLINE	The operation could not be completed because the cluster resource is online.
5020	0x0000139C	ERROR_QUORUM_RESOURCE	The cluster resource could not be deleted or brought offline because it is the quorum resource.
5021	0x0000139D	ERROR_NOT_QUORUM_CAPABLE	The cluster could not make the specified resource a quorum resource because it is not capable of being a quorum resource.
5022	0x0000139E	ERROR_CLUSTER_SHUTTING_DOWN	The cluster software is shutting down.
5023	0x0000139F	ERROR_INVALID_STATE	The group or resource is not in the correct state to perform the requested operation.
5024	0x000013A0	ERROR_RESOURCE_PROPERTIES_STORED	The properties were stored but not all changes will take effect until the next time the resource is brought online.
5025	0x000013A1	ERROR_NOT_QUORUM_CLASS	The cluster could not make the specified resource a quorum resource because it does not belong to a shared storage class.
5026	0x000013A2	ERROR_CORE_RESOURCE	The cluster resource could not be deleted since it is a core resource.
5027	0x000013A3	ERROR_QUORUM_RESOURCE_ONLINE_FAILED	The quorum resource failed to come online.
5028	0x000013A4	ERROR_QUORUMLOG_OPEN_FAILED	The quorum log could not be created or mounted successfully.
5029	0x000013A5	ERROR_CLUSTERLOG_CORRUPT	The cluster log is corrupt.
5030	0x000013A6	ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE	The record could not be written to the cluster log since it exceeds the maximum size.
5031	0x000013A7	ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE	The cluster log exceeds its maximum size.
5032	0x000013A8	ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND	No checkpoint record was found in the cluster log.
5033	0x000013A9	ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE	The minimum required disk space needed for logging is not available.
5034	0x000013AA	ERROR_QUORUM_OWNER_ALIVE	The cluster node failed to take control of the quorum resource because the resource is owned by another active node.
5035	0x000013AB	ERROR_NETWORK_NOT_AVAILABLE	A cluster network is not available for this operation.
5036	0x000013AC	ERROR_NODE_NOT_AVAILABLE	A cluster node is not available for this operation.

Error			Description
decimal	Hexadecimal	Name	
5037	0x000013AD	ERROR_ALL_NODES_NOT_AVAILABLE	All cluster nodes must be running to perform this operation.
5038	0x000013AE	ERROR_RESOURCE_FAILED	A cluster resource failed.
5039	0x000013AF	ERROR_CLUSTER_INVALID_NODE	The cluster node is not valid.
5040	0x000013B0	ERROR_CLUSTER_NODE_EXISTS	The cluster node already exists.
5041	0x000013B1	ERROR_CLUSTER_JOIN_IN_PROGRESS	A node is in the process of joining the cluster.
5042	0x000013B2	ERROR_CLUSTER_NODE_NOT_FOUND	The cluster node was not found.
5043	0x000013B3	ERROR_CLUSTER_LOCAL_NODE_NOT_FOUND	The cluster local node information was not found.
5044	0x000013B4	ERROR_CLUSTER_NETWORK_EXISTS	The cluster network already exists.
5045	0x000013B5	ERROR_CLUSTER_NETWORK_NOT_FOUND	The cluster network was not found.
5046	0x000013B6	ERROR_CLUSTER_NETINTERFACE_EXISTS	The cluster network interface already exists.
5047	0x000013B7	ERROR_CLUSTER_NETINTERFACE_NOT_FOUND	The cluster network interface was not found.
5048	0x000013B8	ERROR_CLUSTER_INVALID_REQUEST	The cluster request is not valid for this object.
5049	0x000013B9	ERROR_CLUSTER_INVALID_NETWORK_PROVIDER	The cluster network provider is not valid.
5050	0x000013BA	ERROR_CLUSTER_NODE_DOWN	The cluster node is down.
5051	0x000013BB	ERROR_CLUSTER_NODE_UNREACHABLE	The cluster node is not reachable.
5052	0x000013BC	ERROR_CLUSTER_NODE_NOT_MEMBER	The cluster node is not a member of the cluster.
5053	0x000013BD	ERROR_CLUSTER_JOIN_NOT_IN_PROGRESS	A cluster join operation is not in progress.
5054	0x000013BE	ERROR_CLUSTER_INVALID_NETWORK	The cluster network is not valid.
5056	0x000013C0	ERROR_CLUSTER_NODE_UP	The cluster node is up.
5057	0x000013C1	ERROR_CLUSTER_IPADDR_IN_USE	The cluster IP address is already in use.
5058	0x000013C2	ERROR_CLUSTER_NODE_NOT_PAUSED	The cluster node is not paused.
5059	0x000013C3	ERROR_CLUSTER_NO_SECURITY_CONTEXT	No cluster security context is available.
5060	0x000013C4	ERROR_CLUSTER_NETWORK_NOT_INTERNAL	The cluster network is not configured for internal cluster communication.
5061	0x000013C5	ERROR_CLUSTER_NODE_ALREADY_UP	The cluster node is already up.
5062	0x000013C6	ERROR_CLUSTER_NODE_ALREADY_DOWN	The cluster node is already down.
5063	0x000013C7	ERROR_CLUSTER_NETWORK_ALREADY_ONLINE	The cluster network is already online.
5064	0x000013C8	ERROR_CLUSTER_NETWORK_ALREADY_OFFLINE	The cluster network is already offline.
5065	0x000013C9	ERROR_CLUSTER_NODE_ALREADY_MEMBER	The cluster node is already a member of the cluster.
5066	0x000013CA	ERROR_CLUSTER_LAST_INTERNAL_NETWORK	The cluster network is the only one configured for internal cluster communication between two or more active cluster nodes. The internal communication capability cannot be removed from the network.
5067	0x000013CB	ERROR_CLUSTER_NETWORK_HAS_DEPENDENTS	One or more cluster resources depend on the network to provide service to clients. The client access capability cannot be removed from the network.
5068	0x000013CC	ERROR_INVALID_OPERATION_ON_QUORUM	This operation cannot be performed on the cluster resource as it the quorum resource. You may not bring the quorum resource offline or modify its possible owners list.
5069	0x000013CD	ERROR_DEPENDENCY_NOT_ALLOWED	The cluster quorum resource is not allowed to have any dependencies.
5070	0x000013CE	ERROR_CLUSTER_NODE_PAUSED	The cluster node is paused.
5071	0x000013CF	ERROR_NODE_CANT_HOST_RESOURCE	The cluster resource cannot be brought online. The owner node cannot run this resource.
5072	0x000013D0	ERROR_CLUSTER_NODE_NOT_READY	The cluster node is not ready to perform the requested operation.
5073	0x000013D1	ERROR_CLUSTER_NODE_SHUTTING_DOWN	The cluster node is shutting down.
5074	0x000013D2	ERROR_CLUSTER_JOIN_ABORTED	The cluster join operation was aborted.
5075	0x000013D3	ERROR_CLUSTER_INCOMPATIBLE_VERSIONS	The cluster join operation failed due to incompatible software versions between the joining node and its sponsor.
5076	0x000013D4	ERROR_CLUSTER_MAXNUM_OF_RESOURCES_EXCEEDED	This resource cannot be created because the cluster has reached the limit on the number of resources it can monitor.

Error			Description
decimal	Hexadecimal	Name	
5077	0x000013D5	ERROR_CLUSTER_SYSTEM_CONFIG_CHANGED	The system configuration changed during the cluster join or form operation. The join or form operation was aborted.
5078	0x000013D6	ERROR_CLUSTER_RESOURCE_TYPE_NOT_FOUND	The specified resource type was not found.
5079	0x000013D7	ERROR_CLUSTER_RESTYPE_NOT_SUPPORTED	The specified node does not support a resource of this type. This may be due to version inconsistencies or due to the absence of the resource DLL on this node.
5080	0x000013D8	ERROR_CLUSTER_RESNAME_NOT_FOUND	The specified resource name is supported by this resource DLL. This may be due to a bad (or changed) name supplied to the resource DLL.
5081	0x000013D9	ERROR_CLUSTER_NO_RPC_PACKAGES_REGISTERED	No authentication package could be registered with the RPC server.
5082	0x000013DA	ERROR_CLUSTER_OWNER_NOT_IN_PREFLIST	You cannot bring the group online because the owner of the group is not in the preferred list for the group. To change the owner node for the group, move the group.
5083	0x000013DB	ERROR_CLUSTER_DATABASE_SEQMISMATCH	The join operation failed because the cluster database sequence number has changed or is incompatible with the locker node. This may happen during a join operation if the cluster database was changing during the join.
5084	0x000013DC	ERROR_RESMON_INVALID_STATE	The resource monitor will not allow the fail operation to be performed while the resource is in its current state. This may happen if the resource is in a pending state.
5085	0x000013DD	ERROR_CLUSTER_GUM_NOT_LOCKER	A non locker code got a request to reserve the lock for making global updates.
5086	0x000013DE	ERROR_QUORUM_DISK_NOT_FOUND	The quorum disk could not be located by the cluster service.
5087	0x000013DF	ERROR_DATABASE_BACKUP_CORRUPT	The backup up cluster database is possibly corrupt.
5088	0x000013E0	ERROR_CLUSTER_NODE_ALREADY_HAS_DFS_ROOT	A DFS root already exists in this cluster node.
5089	0x000013E1	ERROR_RESOURCE_PROPERTY_UNCHANGEABLE	An attempt to modify a resource property failed because it conflicts with another existing property.
5890	0x00001702	ERROR_CLUSTER_MEMBERSHIP_INVALID_STATE	An operation was attempted that is incompatible with the current membership state of the node.
5891	0x00001703	ERROR_CLUSTER_QUORUMLOG_NOT_FOUND	The quorum resource does not contain the quorum log.
5892	0x00001704	ERROR_CLUSTER_MEMBERSHIP_HALT	The membership engine requested shutdown of the cluster service on this node.
5893	0x00001705	ERROR_CLUSTER_INSTANCE_ID_MISMATCH	The join operation failed because the cluster instance ID of the joining node does not match the cluster instance ID of the sponsor node.
5894	0x00001706	ERROR_CLUSTER_NETWORK_NOT_FOUND_FOR_IP	A matching network for the specified IP address could not be found. Please also specify a subnet mask and a cluster network.
5895	0x00001707	ERROR_CLUSTER_PROPERTY_DATA_TYPE_MISMATCH	The actual data type of the property did not match the expected data type of the property.
5896	0x00001708	ERROR_CLUSTER_EVICT_WITHOUT_CLEANUP	The cluster node was evicted from the cluster successfully. The node was not cleaned up because it does not support the evict cleanup functionality.

Error			Description
decimal	Hexadecimal	Name	
6000	0x00001770	ERROR_ENCRYPTION_FAILED	The specified file could not be encrypted.
6001	0x00001771	ERROR_DECRYPTION_FAILED	The specified file could not be decrypted.
6002	0x00001772	ERROR_FILE_ENCRYPTED	The specified file is encrypted and the user does not have the ability to decrypt it.
6003	0x00001773	ERROR_NO_RECOVERY_POLICY	There is no valid encryption recovery policy configured for this system.
6004	0x00001774	ERROR_NO_EFS	The required encryption driver is not loaded for this system.
6005	0x00001775	ERROR_WRONG_EFS	The file was encrypted with a different encryption driver than is currently loaded.
6006	0x00001776	ERROR_NO_USER_KEYS	There are no EFS keys defined for the user.
6007	0x00001777	ERROR_FILE_NOT_ENCRYPTED	The specified file is not encrypted.
6008	0x00001778	ERROR_NOT_EXPORT_FORMAT	The specified file is not in the defined EFS export format.
6009	0x00001779	ERROR_FILE_READ_ONLY	The specified file is read only.
6010	0x0000177A	ERROR_DIR_EFS_DISALLOWED	The directory has been disabled for encryption.
6011	0x0000177B	ERROR_EFS_SERVER_NOT_TRUSTED	The server is not trusted for remote encryption operation.
6012	0x0000177C	ERROR_BAD_RECOVERY_POLICY	Recovery policy configured for this system contains invalid recovery certificate.
6013	0x0000177D	ERROR_EFS_ALG_BLOB_TOO_BIG	The encryption algorithm used on the source file needs a bigger key buffer than the one on the destination file.
6014	0x0000177E	ERROR_VOLUME_NOT_SUPPORT_EFS	The disk partition does not support file encryption.
6118	0x000017E6	ERROR_NO_BROWSER_SERVERS_FOUND	The list of servers for this workgroup is not currently available.
6200	0x00001838	SCHED_E_SERVICE_NOT_LOCALSYSTEM	The Task Scheduler service must be configured to run in the System account to function properly. Individual tasks may be configured to run in other accounts.
7001	0x00001B59	ERROR_CTX_WINSTATION_NAME_INVALID	The specified session name is invalid.
7002	0x00001B5A	ERROR_CTX_INVALID_PD	The specified protocol driver is invalid.
7003	0x00001B5B	ERROR_CTX_PD_NOT_FOUND	The specified protocol driver was not found in the system path.
7004	0x00001B5C	ERROR_CTX_WD_NOT_FOUND	The specified terminal connection driver was not found in the system path.
7005	0x00001B5D	ERROR_CTX_CANNOT_MAKE_EVENTLOG_ENTRY	A registry key for event logging could not be created for this session.
7006	0x00001B5E	ERROR_CTX_SERVICE_NAME_COLLISION	A service with the same name already exists on the system.
7007	0x00001B5F	ERROR_CTX_CLOSE_PENDING	A close operation is pending on the session.
7008	0x00001B60	ERROR_CTX_NO_OUTBUF	There are no free output buffers available.
7009	0x00001B61	ERROR_CTX_MODEM_INF_NOT_FOUND	The MODEM.INF file was not found.
7010	0x00001B62	ERROR_CTX_INVALID_MODEMNAME	The modem name was not found in MODEM.INF.
7011	0x00001B63	ERROR_CTX_MODEM_RESPONSE_ERROR	The modem did not accept the command sent to it. Verify that the configured modem name matches the attached modem.
7012	0x00001B64	ERROR_CTX_MODEM_RESPONSE_TIMEOUT	The modem did not respond to the command sent to it. Verify that the modem is properly cabled and powered on.
7013	0x00001B65	ERROR_CTX_MODEM_RESPONSE_NO_CARRIER	Carrier detect has failed or carrier has been dropped due to disconnect.
7014	0x00001B66	ERROR_CTX_MODEM_RESPONSE_NO_DIALTONE	Dial tone not detected within the required time. Verify that the phone cable is properly attached and functional.
7015	0x00001B67	ERROR_CTX_MODEM_RESPONSE_BUSY	Busy signal detected at remote site on callback.
7016	0x00001B68	ERROR_CTX_MODEM_RESPONSE_VOICE	Voice detected at remote site on callback.
7017	0x00001B69	ERROR_CTX_TD_ERROR	Transport driver error
7022	0x00001B6E	ERROR_CTX_WINSTATION_NOT_FOUND	The specified session cannot be found.
7023	0x00001B6F	ERROR_CTX_WINSTATION_ALREADY_EXISTS	The specified session name is already in use.
7024	0x00001B70	ERROR_CTX_WINSTATION_BUSY	The requested operation cannot be completed because the terminal connection is currently busy processing a connect, disconnect, reset, or delete operation.

Error			Description
decimal	Hexadecimal	Name	
7025	0x00001B71	ERROR_CTX_BAD_VIDEO_MODE	An attempt has been made to connect to a session whose video mode is not supported by the current client.
7035	0x00001B7B	ERROR_CTX_GRAPHICS_INVALID	The application attempted to enable DOS graphics mode. DOS graphics mode is not supported.
7037	0x00001B7D	ERROR_CTX_LOGON_DISABLED	Your interactive logon privilege has been disabled. Please contact your administrator.
7038	0x00001B7E	ERROR_CTX_NOT_CONSOLE	The requested operation can be performed only on the system console. This is most often the result of a driver or system DLL requiring direct console access.
7040	0x00001B80	ERROR_CTX_CLIENT_QUERY_TIMEOUT	The client failed to respond to the server connect message.
7041	0x00001B81	ERROR_CTX_CONSOLE_DISCONNECT	Disconnecting the console session is not supported.
7042	0x00001B82	ERROR_CTX_CONSOLE_CONNECT	Reconnecting a disconnected session to the console is not supported.
7044	0x00001B84	ERROR_CTX_SHADOW_DENIED	The request to control another session remotely was denied.
7045	0x00001B85	ERROR_CTX_WINSTATION_ACCESS_DENIED	The requested session access is denied.
7049	0x00001B89	ERROR_CTX_INVALID_WD	The specified terminal connection driver is invalid.
7050	0x00001B8A	ERROR_CTX_SHADOW_INVALID	The requested session cannot be controlled remotely. This may be because the session is disconnected or does not currently have a user logged on.
7051	0x00001B8B	ERROR_CTX_SHADOW_DISABLED	The requested session is not configured to allow remote control.
7052	0x00001B8C	ERROR_CTX_CLIENT_LICENSE_IN_USE	Your request to connect to this Terminal Server has been rejected. Your Terminal Server client license number is currently being used by another user. Please call your system administrator to obtain a unique license number.
7053	0x00001B8D	ERROR_CTX_CLIENT_LICENSE_NOT_SET	Your request to connect to this Terminal Server has been rejected. Your Terminal Server client license number has not been entered for this copy of the Terminal Server client. Please contact your system administrator.
7054	0x00001B8E	ERROR_CTX_LICENSE_NOT_AVAILABLE	The system has reached its licensed logon limit. Please try again later.
7055	0x00001B8F	ERROR_CTX_LICENSE_CLIENT_INVALID	The client you are using is not licensed to use this system. Your logon request is denied.
7056	0x00001B90	ERROR_CTX_LICENSE_EXPIRED	The system license has expired. Your logon request is denied.
7057	0x00001B91	ERROR_CTX_SHADOW_NOT_RUNNING	Remote control could not be terminated because the specified session is not currently being remotely controlled.

Error			Description
decimal	Hexadecimal	Name	
8001	0x00001F41	FRS_ERR_INVALID_API_SEQUENCE	The file replication service API was called incorrectly.
8002	0x00001F42	FRS_ERR_STARTING_SERVICE	The file replication service cannot be started.
8003	0x00001F43	FRS_ERR_STOPPING_SERVICE	The file replication service cannot be stopped.
8004	0x00001F44	FRS_ERR_INTERNAL_API	The file replication service API terminated the request. The event log may have more information.
8005	0x00001F45	FRS_ERR_INTERNAL	The file replication service terminated the request. The event log may have more information.
8006	0x00001F46	FRS_ERR_SERVICE_COMM	The file replication service cannot be contacted. The event log may have more information.
8007	0x00001F47	FRS_ERR_INSUFFICIENT_PRIV	The file replication service cannot satisfy the request because the user has insufficient privileges. The event log may have more information.
8008	0x00001F48	FRS_ERR_AUTHENTICATION	The file replication service cannot satisfy the request because authenticated RPC is not available. The event log may have more information.
8009	0x00001F49	FRS_ERR_PARENT_INSUFFICIENT_PRIV	The file replication service cannot satisfy the request because the user has insufficient privileges on the domain controller. The event log may have more information.
8010	0x00001F4A	FRS_ERR_PARENT_AUTHENTICATION	The file replication service cannot satisfy the request because authenticated RPC is not available on the domain controller. The event log may have more information.
8011	0x00001F4B	FRS_ERR_CHILD_TO_PARENT_COMM	The file replication service cannot communicate with the file replication service on the domain controller. The event log may have more information.
8012	0x00001F4C	FRS_ERR_PARENT_TO_CHILD_COMM	The file replication service on the domain controller cannot communicate with the file replication service on this computer. The event log may have more information.
8013	0x00001F4D	FRS_ERR_SYSVOL_POPULATE	The file replication service cannot populate the system volume because of an internal error. The event log may have more information.
8014	0x00001F4E	FRS_ERR_SYSVOL_POPULATE_TIMEOUT	The file replication service cannot populate the system volume because of an internal timeout. The event log may have more information.
8015	0x00001F4F	FRS_ERR_SYSVOL_IS_BUSY	The file replication service cannot process the request. The system volume is busy with a previous request.
8016	0x00001F50	FRS_ERR_SYSVOL_DEMOTE	The file replication service cannot stop replicating the system volume because of an internal error. The event log may have more information.
8017	0x00001F51	FRS_ERR_INVALID_SERVICE_PARAMETER	The file replication service detected an invalid parameter.
8200	0x00002008	ERROR_DS_NOT_INSTALLED	An error occurred while installing the directory service. For more information, see the event log.
8201	0x00002009	ERROR_DS_MEMBERSHIP_EVALUATED_LOCALLY	The directory service evaluated group memberships locally.
8202	0x0000200A	ERROR_DS_NO_ATTRIBUTE_OR_VALUE	The specified directory service attribute or value does not exist.
8203	0x0000200B	ERROR_DS_INVALID_ATTRIBUTE_SYNTAX	The attribute syntax specified to the directory service is invalid.
8204	0x0000200C	ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED	The attribute type specified to the directory service is not defined.
8205	0x0000200D	ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS	The specified directory service attribute or value already exists.
8206	0x0000200E	ERROR_DS_BUSY	The directory service is busy.
8207	0x0000200F	ERROR_DS_UNAVAILABLE	The directory service is unavailable.
8208	0x00002010	ERROR_DS_NO_RIDS_ALLOCATED	The directory service was unable to allocate a relative identifier.
8209	0x00002011	ERROR_DS_NO_MORE_RIDS	The directory service has exhausted the pool of relative identifiers.

Error			Description
decimal	Hexadecimal	Name	
8210	0x00002012	ERROR_DS_INCORRECT_ROLE_OWNER	The requested operation could not be performed because the directory service is not the master for that type of operation.
8211	0x00002013	ERROR_DS_RIDMGR_INIT_ERROR	The directory service was unable to initialize the subsystem that allocates relative identifiers.
8212	0x00002014	ERROR_DS_OBJ_CLASS_VIOLATION	The requested operation did not satisfy one or more constraints associated with the class of the object.
8213	0x00002015	ERROR_DS_CANT_ON_NON_LEAF	The directory service can perform the requested operation only on a leaf object.
8214	0x00002016	ERROR_DS_CANT_ON_RDN	The directory service cannot perform the requested operation on the RDN attribute of an object.
8215	0x00002017	ERROR_DS_CANT_MOD_OBJ_CLASS	The directory service detected an attempt to modify the object class of an object.
8216	0x00002018	ERROR_DS_CROSS_DOM_MOVE_ERROR	The requested cross-domain move operation could not be performed.
8217	0x00002019	ERROR_DS_GC_NOT_AVAILABLE	Unable to contact the global catalog server.
8218	0x0000201A	ERROR_SHARED_POLICY	The policy object is shared and can only be modified at the root.
8219	0x0000201B	ERROR_POLICY_OBJECT_NOT_FOUND	The policy object does not exist.
8220	0x0000201C	ERROR_POLICY_ONLY_IN_DS	The requested policy information is only in the directory service.
8221	0x0000201D	ERROR_PROMOTION_ACTIVE	A domain controller promotion is currently active.
8222	0x0000201E	ERROR_NO_PROMOTION_ACTIVE	A domain controller promotion is not currently active
8224	0x00002020	ERROR_DS_OPERATIONS_ERROR	An operations error occurred.
8225	0x00002021	ERROR_DS_PROTOCOL_ERROR	A protocol error occurred.
8226	0x00002022	ERROR_DS_TIMELIMIT_EXCEEDED	The time limit for this request was exceeded.
8227	0x00002023	ERROR_DS_SIZELIMIT_EXCEEDED	The size limit for this request was exceeded.
8228	0x00002024	ERROR_DS_ADMIN_LIMIT_EXCEEDED	The administrative limit for this request was exceeded.
8229	0x00002025	ERROR_DS_COMPARE_FALSE	The compare response was false.
8230	0x00002026	ERROR_DS_COMPARE_TRUE	The compare response was true.
8231	0x00002027	ERROR_DS_AUTH_METHOD_NOT_SUPPORTED	The requested authentication method is not supported by the server.
8232	0x00002028	ERROR_DS_STRONG_AUTH_REQUIRED	A more secure authentication method is required for this server.
8233	0x00002029	ERROR_DS_INAPPROPRIATE_AUTH	Inappropriate authentication.
8234	0x0000202A	ERROR_DS_AUTH_UNKNOWN	The authentication mechanism is unknown.
8235	0x0000202B	ERROR_DS_REFERRAL	A referral was returned from the server.
8236	0x0000202C	ERROR_DS_UNAVAILABLE_CRIT_EXTENSION	The server does not support the requested critical extension.
8237	0x0000202D	ERROR_DS_CONFIDENTIALITY_REQUIRED	This request requires a secure connection.
8238	0x0000202E	ERROR_DS_INAPPROPRIATE_MATCHING	Inappropriate matching.
8239	0x0000202F	ERROR_DS_CONSTRAINT_VIOLATION	A constraint violation occurred.
8240	0x00002030	ERROR_DS_NO_SUCH_OBJECT	There is no such object on the server.
8241	0x00002031	ERROR_DS_ALIAS_PROBLEM	There is an alias problem.
8242	0x00002032	ERROR_DS_INVALID_DN_SYNTAX	An invalid dn syntax has been specified.
8243	0x00002033	ERROR_DS_IS_LEAF	The object is a leaf object.
8244	0x00002034	ERROR_DS_ALIAS_DEREF_PROBLEM	There is an alias dereferencing problem.
8245	0x00002035	ERROR_DS_UNWILLING_TO_PERFORM	The server is unwilling to process the request.
8246	0x00002036	ERROR_DS_LOOP_DETECT	A loop has been detected.
8247	0x00002037	ERROR_DS_NAMING_VIOLATION	There is a naming violation.
8248	0x00002038	ERROR_DS_OBJECT_RESULTS_TOO_LARGE	The result set is too large.
8249	0x00002039	ERROR_DS_AFFECTS_MULTIPLE_DSAS	The operation affects multiple DSAs
8250	0x0000203A	ERROR_DS_SERVER_DOWN	The server is not operational.
8251	0x0000203B	ERROR_DS_LOCAL_ERROR	A local error has occurred.
8252	0x0000203C	ERROR_DS_ENCODING_ERROR	An encoding error has occurred.
8253	0x0000203D	ERROR_DS_DECODING_ERROR	A decoding error has occurred.
8254	0x0000203E	ERROR_DS_FILTER_UNKNOWN	The search filter cannot be recognized.

Error			Description
decimal	Hexadecimal	Name	
8255	0x0000203F	ERROR_DS_PARAM_ERROR	One or more parameters are illegal.
8256	0x00002040	ERROR_DS_NOT_SUPPORTED	The specified method is not supported.
8257	0x00002041	ERROR_DS_NO_RESULTS_RETURNED	No results were returned.
8258	0x00002042	ERROR_DS_CONTROL_NOT_FOUND	The specified control is not supported by the server.
8259	0x00002043	ERROR_DS_CLIENT_LOOP	A referral loop was detected by the client.
8260	0x00002044	ERROR_DS_REFERRAL_LIMIT_EXCEEDED	The preset referral limit was exceeded.
8261	0x00002045	ERROR_DS_SORT_CONTROL_MISSING	The search requires a SORT control.
8262	0x00002046	ERROR_DS_OFFSET_RANGE_ERROR	The search results exceed the offset range specified.
8301	0x0000206D	ERROR_DS_ROOT_MUST_BE_NC	The root object must be the head of a naming context. The root object cannot have an instantiated parent.
8302	0x0000206E	ERROR_DS_ADD_REPLICA_INHIBITED	The add replica operation cannot be performed. The naming context must be writable in order to create the replica.
8303	0x0000206F	ERROR_DS_ATT_NOT_DEF_IN_SCHEMA	A reference to an attribute that is not defined in the schema occurred.
8304	0x00002070	ERROR_DS_MAX_OBJ_SIZE_EXCEEDED	The maximum size of an object has been exceeded.
8305	0x00002071	ERROR_DS_OBJ_STRING_NAME_EXISTS	An attempt was made to add an object to the directory with a name that is already in use.
8306	0x00002072	ERROR_DS_NO_RDN_DEFINED_IN_SCHEMA	An attempt was made to add an object of a class that does not have an RDN defined in the schema.
8307	0x00002073	ERROR_DS_RDN_DOESNT_MATCH_SCHEMA	An attempt was made to add an object using an RDN that is not the RDN defined in the schema.
8308	0x00002074	ERROR_DS_NO_REQUESTED_ATTS_FOUND	None of the requested attributes were found on the objects.
8309	0x00002075	ERROR_DS_USER_BUFFER_TO_SMALL	The user buffer is too small.
8310	0x00002076	ERROR_DS_ATT_IS_NOT_ON_OBJ	The attribute specified in the operation is not present on the object.
8311	0x00002077	ERROR_DS_ILLEGAL_MOD_OPERATION	Illegal modify operation. Some aspect of the modification is not permitted.
8312	0x00002078	ERROR_DS_OBJ_TOO_LARGE	The specified object is too large.
8313	0x00002079	ERROR_DS_BAD_INSTANCE_TYPE	The specified instance type is not valid.
8314	0x0000207A	ERROR_DS_MASTERDSA_REQUIRED	The operation must be performed at a master DSA.
8315	0x0000207B	ERROR_DS_OBJECT_CLASS_REQUIRED	The object class attribute must be specified.
8316	0x0000207C	ERROR_DS_MISSING_REQUIRED_ATT	A required attribute is missing.
8317	0x0000207D	ERROR_DS_ATT_NOT_DEF_FOR_CLASS	An attempt was made to modify an object to include an attribute that is not legal for its class
8318	0x0000207E	ERROR_DS_ATT_ALREADY_EXISTS	The specified attribute is already present on the object.
8320	0x00002080	ERROR_DS_CANT_ADD_ATT_VALUES	The specified attribute is not present, or has no values.
8321	0x00002081	ERROR_DS_SINGLE_VALUE_CONSTRAINT	Multiple values were specified for an attribute that can have only one value.
8322	0x00002082	ERROR_DS_RANGE_CONSTRAINT	A value for the attribute was not in the acceptable range of values.
8323	0x00002083	ERROR_DS_ATT_VAL_ALREADY_EXISTS	The specified value already exists.
8324	0x00002084	ERROR_DS_CANT_REM_MISSING_ATT	The attribute cannot be removed because it is not present on the object.
8325	0x00002085	ERROR_DS_CANT_REM_MISSING_ATT_VAL	The attribute value cannot be removed because it is not present on the object.
8326	0x00002086	ERROR_DS_ROOT_CANT_BE_SUBREF	The specified root object cannot be a subref.
8327	0x00002087	ERROR_DS_NO_CHAINING	Chaining is not permitted.
8328	0x00002088	ERROR_DS_NO_CHAINED_EVAL	Chained evaluation is not permitted.
8329	0x00002089	ERROR_DS_NO_PARENT_OBJECT	The operation could not be performed because the object's parent is either uninstantiated or deleted.
8330	0x0000208A	ERROR_DS_PARENT_IS_AN_ALIAS	Having a parent that is an alias is not permitted. Aliases are leaf objects.

Error			Description
decimal	Hexadecimal	Name	
8331	0x0000208B	ERROR_DS_CANT_MIX_MASTER_AND_REPS	The object and parent must be of the same type, either both masters or both replicas.
8332	0x0000208C	ERROR_DS_CHILDREN_EXIST	The operation cannot be performed because child objects exist. This operation can only be performed on a leaf object.
8333	0x0000208D	ERROR_DS_OBJ_NOT_FOUND	Directory object not found.
8334	0x0000208E	ERROR_DS_ALIASED_OBJ_MISSING	The aliased object is missing.
8335	0x0000208F	ERROR_DS_BAD_NAME_SYNTAX	The object name has bad syntax.
8336	0x00002090	ERROR_DS_ALIAS_POINTS_TO_ALIAS	It is not permitted for an alias to refer to another alias.
8337	0x00002091	ERROR_DS_CANT_DEREF_ALIAS	The alias cannot be dereferenced.
8338	0x00002092	ERROR_DS_OUT_OF_SCOPE	The operation is out of scope.
8339	0x00002093	ERROR_DS_OBJECT_BEING_REMOVED	The operation cannot continue because the object is in the process of being removed.
8340	0x00002094	ERROR_DS_CANT_DELETE_DSA_OBJ	The DSA object cannot be deleted.
8341	0x00002095	ERROR_DS_GENERIC_ERROR	A directory service error has occurred.
8342	0x00002096	ERROR_DS_DSA_MUST_BE_INT_MASTER	The operation can only be performed on an internal master DSA object.
8343	0x00002097	ERROR_DS_CLASS_NOT_DSA	The object must be of class DSA.
8344	0x00002098	ERROR_DS_INSUFF_ACCESS_RIGHTS	Insufficient access rights to perform the operation.
8345	0x00002099	ERROR_DS_ILLEGAL_SUPERIOR	The object cannot be added because the parent is not on the list of possible superiors.
8346	0x0000209A	ERROR_DS_ATTRIBUTE_OWNED_BY_SAM	Access to the attribute is not permitted because the attribute is owned by the Security Accounts Manager (SAM).
8347	0x0000209B	ERROR_DS_NAME_TOO_MANY_PARTS	The name has too many parts.
8348	0x0000209C	ERROR_DS_NAME_TOO_LONG	The name is too long.
8349	0x0000209D	ERROR_DS_NAME_VALUE_TOO_LONG	The name value is too long.
8350	0x0000209E	ERROR_DS_NAME_UNPARSEABLE	The directory service encountered an error parsing a name.
8351	0x0000209F	ERROR_DS_NAME_TYPE_UNKNOWN	The directory service cannot get the attribute type for a name.
8352	0x000020A0	ERROR_DS_NOT_AN_OBJECT	The name does not identify an object; the name identifies a phantom.
8353	0x000020A1	ERROR_DS_SEC_DESC_TOO_SHORT	The security descriptor is too short.
8354	0x000020A2	ERROR_DS_SEC_DESC_INVALID	The security descriptor is invalid.
8355	0x000020A3	ERROR_DS_NO_DELETED_NAME	Failed to create name for deleted object.
8356	0x000020A4	ERROR_DS_SUBREF_MUST_HAVE_PARENT	The parent of a new subref must exist.
8357	0x000020A5	ERROR_DS_NCNAME_MUST_BE_NC	The object must be a naming context.
8358	0x000020A6	ERROR_DS_CANT_ADD_SYSTEM_ONLY	It is not permitted to add an attribute which is owned by the system.
8359	0x000020A7	ERROR_DS_CLASS_MUST_BE_CONCRETE	The class of the object must be structural; you cannot instantiate an abstract class.
8360	0x000020A8	ERROR_DS_INVALID_DMD	The schema object could not be found.
8361	0x000020A9	ERROR_DS_OBJ_GUID_EXISTS	A local object with this GUID (dead or alive) already exists.
8362	0x000020AA	ERROR_DS_NOT_ON_BACKLINK	The operation cannot be performed on a back link.
8363	0x000020AB	ERROR_DS_NO_CROSSREF_FOR_NC	The cross reference for the specified naming context could not be found.
8364	0x000020AC	ERROR_DS_SHUTTING_DOWN	The operation could not be performed because the directory service is shutting down.
8365	0x000020AD	ERROR_DS_UNKNOWN_OPERATION	The directory service request is invalid.
8366	0x000020AE	ERROR_DS_INVALID_ROLE_OWNER	The role owner attribute could not be read.
8367	0x000020AF	ERROR_DS_COULDNT_CONTACT_FSMO	The requested FSMO operation failed. The current FSMO holder could not be reached.
8368	0x000020B0	ERROR_DS_CROSS_NC_DN_RENAME	Modification of a DN across a naming context is not permitted.
8369	0x000020B1	ERROR_DS_CANT_MOD_SYSTEM_ONLY	The attribute cannot be modified because it is owned by the system.
8370	0x000020B2	ERROR_DS_REPLICATOR_ONLY	Only the replicator can perform this function.
8371	0x000020B3	ERROR_DS_OBJ_CLASS_NOT_DEFINED	The specified class is not defined.

Error			Description
decimal	Hexadecimal	Name	
8372	0x000020B4	ERROR_DS_OBJ_CLASS_NOT_SUBCLASS	The specified class is not a subclass.
8373	0x000020B5	ERROR_DS_NAME_REFERENCE_INVALID	The name reference is invalid.
8374	0x000020B6	ERROR_DS_CROSS_REF_EXISTS	A cross reference already exists.
8375	0x000020B7	ERROR_DS_CANT_DEL_MASTER_CROSSREF	It is not permitted to delete a master cross reference.
8376	0x000020B8	ERROR_DS_SUBTREE_NOTIFY_NOT_NC_HEAD	Subtree notifications are only supported on NC heads.
8377	0x000020B9	ERROR_DS_NOTIFY_FILTER_TOO_COMPLEX	Notification filter is too complex.
8378	0x000020BA	ERROR_DS_DUP_RDN	Schema update failed: duplicate RDN.
8379	0x000020BB	ERROR_DS_DUP_OID	Schema update failed: duplicate OID
8380	0x000020BC	ERROR_DS_DUP_MAPI_ID	Schema update failed: duplicate MAPI identifier.
8381	0x000020BD	ERROR_DS_DUP_SCHEMA_ID_GUID	Schema update failed: duplicate schema-id GUID.
8382	0x000020BE	ERROR_DS_DUP_LDAP_DISPLAY_NAME	Schema update failed: duplicate LDAP display name.
8383	0x000020BF	ERROR_DS_SEMANTIC_ATT_TEST	Schema update failed: range-lower less than range upper
8384	0x000020C0	ERROR_DS_SYNTAX_MISMATCH	Schema update failed: syntax mismatch
8385	0x000020C1	ERROR_DS_EXISTS_IN_MUST_HAVE	Schema deletion failed: attribute is used in must-contain
8386	0x000020C2	ERROR_DS_EXISTS_IN_MAY_HAVE	Schema deletion failed: attribute is used in may-contain
8387	0x000020C3	ERROR_DS_NONEXISTENT_MAY_HAVE	Schema update failed: attribute in may-contain does not exist
8388	0x000020C4	ERROR_DS_NONEXISTENT_MUST_HAVE	Schema update failed: attribute in must-contain does not exist
8389	0x000020C5	ERROR_DS_AUX_CLS_TEST_FAIL	Schema update failed: class in aux-class list does not exist or is not an auxiliary class
8390	0x000020C6	ERROR_DS_NONEXISTENT_POSS_SUP	Schema update failed: class in poss-superiors does not exist
8391	0x000020C7	ERROR_DS_SUB_CLS_TEST_FAIL	Schema update failed: class in subclassof list does not exist or does not satisfy hierarchy rules
8392	0x000020C8	ERROR_DS_BAD_RDN_ATT_ID_SYNTAX	Schema update failed: Rdn-Att-Id has wrong syntax
8393	0x000020C9	ERROR_DS_EXISTS_IN_AUX_CLS	Schema deletion failed: class is used as auxiliary class
8394	0x000020CA	ERROR_DS_EXISTS_IN_SUB_CLS	Schema deletion failed: class is used as sub class
8395	0x000020CB	ERROR_DS_EXISTS_IN_POSS_SUP	Schema deletion failed: class is used as poss superior
8396	0x000020CC	ERROR_DS_RECALCSCHEMA_FAILED	Schema update failed in recalculating validation cache.
8397	0x000020CD	ERROR_DS_TREE_DELETE_NOT_FINISHED	The tree deletion is not finished.
8398	0x000020CE	ERROR_DS_CANT_DELETE	The requested delete operation could not be performed.
8399	0x000020CF	ERROR_DS_ATT_SCHEMA_REQ_ID	Cannot read the governs class identifier for the schema record.
8400	0x000020D0	ERROR_DS_BAD_ATT_SCHEMA_SYNTAX	The attribute schema has bad syntax.
8401	0x000020D1	ERROR_DS_CANT_CACHE_ATT	The attribute could not be cached.
8402	0x000020D2	ERROR_DS_CANT_CACHE_CLASS	The class could not be cached.
8403	0x000020D3	ERROR_DS_CANT_REMOVE_ATT_CACHE	The attribute could not be removed from the cache.
8404	0x000020D4	ERROR_DS_CANT_REMOVE_CLASS_CACHE	The class could not be removed from the cache.
8405	0x000020D5	ERROR_DS_CANT_RETRIEVE_DN	The distinguished name attribute could not be read.
8406	0x000020D6	ERROR_DS_MISSING_SUPREF	A required subref is missing.
8407	0x000020D7	ERROR_DS_CANT_RETRIEVE_INSTANCE	The instance type attribute could not be retrieved.
8408	0x000020D8	ERROR_DS_CODE_INCONSISTENCY	An internal error has occurred.
8409	0x000020D9	ERROR_DS_DATABASE_ERROR	A database error has occurred.
8410	0x000020DA	ERROR_DS_GOVERNSID_MISSING	The attribute GOVERNSID is missing.
8411	0x000020DB	ERROR_DS_MISSING_EXPECTED_ATT	An expected attribute is missing.
8412	0x000020DC	ERROR_DS_NCNAME_MISSING_CR_REF	The specified naming context is missing a cross reference.
8413	0x000020DD	ERROR_DS_SECURITY_CHECKING_ERROR	A security checking error has occurred.

Error			Description
decimal	Hexadecimal	Name	
8414	0x000020DE	ERROR_DS_SCHEMA_NOT_LOADED	The schema is not loaded.
8415	0x000020DF	ERROR_DS_SCHEMA_ALLOC_FAILED	Schema allocation failed. Please check if the machine is running low on memory.
8416	0x000020E0	ERROR_DS_ATT_SCHEMA_REQ_SYNTAX	Failed to obtain the required syntax for the attribute schema.
8417	0x000020E1	ERROR_DS_GCVERIFY_ERROR	The global catalog verification failed. The global catalog is not available or does not support the operation. Some part of the directory is currently not available.
8418	0x000020E2	ERROR_DS_DRA_SCHEMA_MISMATCH	The replication operation failed because of a schema mismatch between the servers involved.
8419	0x000020E3	ERROR_DS_CANT_FIND_DSA_OBJ	The DSA object could not be found.
8420	0x000020E4	ERROR_DS_CANT_FIND_EXPECTED_NC	The naming context could not be found.
8421	0x000020E5	ERROR_DS_CANT_FIND_NC_IN_CACHE	The naming context could not be found in the cache.
8422	0x000020E6	ERROR_DS_CANT_RETRIEVE_CHILD	The child object could not be retrieved.
8423	0x000020E7	ERROR_DS_SECURITY_ILLEGAL_MODIFY	The modification was not permitted for security reasons.
8424	0x000020E8	ERROR_DS_CANT_REPLACE_HIDDEN_REC	The operation cannot replace the hidden record.
8425	0x000020E9	ERROR_DS_BAD_HIERARCHY_FILE	The hierarchy file is invalid.
8426	0x000020EA	ERROR_DS_BUILD_HIERARCHY_TABLE_FAILED	The attempt to build the hierarchy table failed.
8427	0x000020EB	ERROR_DS_CONFIG_PARAM_MISSING	The directory configuration parameter is missing from the registry.
8428	0x000020EC	ERROR_DS_COUNTING_AB_INDICES_FAILED	The attempt to count the address book indices failed.
8429	0x000020ED	ERROR_DS_HIERARCHY_TABLE_MALLOC_FAILED	The allocation of the hierarchy table failed.
8430	0x000020EE	ERROR_DS_INTERNAL_FAILURE	The directory service encountered an internal failure.
8431	0x000020EF	ERROR_DS_UNKNOWN_ERROR	The directory service encountered an unknown failure.
8432	0x000020F0	ERROR_DS_ROOT_REQUIRES_CLASS_TOP	A root object requires a class of 'top'.
8433	0x000020F1	ERROR_DS_REFUSING_FSMO_ROLES	This directory server is shutting down, and cannot take ownership of new floating single-master operation roles.
8434	0x000020F2	ERROR_DS_MISSING_FSMO_SETTINGS	The directory service is missing mandatory configuration information, and is unable to determine the ownership of floating single-master operation roles.
8435	0x000020F3	ERROR_DS_UNABLE_TO_SURRENDER_ROLES	The directory service was unable to transfer ownership of one or more floating single-master operation roles to other servers.
8436	0x000020F4	ERROR_DS_DRA_GENERIC	The replication operation failed.
8437	0x000020F5	ERROR_DS_DRA_INVALID_PARAMETER	An invalid parameter was specified for this replication operation.
8438	0x000020F6	ERROR_DS_DRA_BUSY	The directory service is too busy to complete the replication operation at this time.
8439	0x000020F7	ERROR_DS_DRA_BAD_DN	The distinguished name specified for this replication operation is invalid.
8440	0x000020F8	ERROR_DS_DRA_BAD_NC	The naming context specified for this replication operation is invalid.
8441	0x000020F9	ERROR_DS_DRA_DN_EXISTS	The distinguished name specified for this replication operation already exists.
8442	0x000020FA	ERROR_DS_DRA_INTERNAL_ERROR	The replication system encountered an internal error.
8443	0x000020FB	ERROR_DS_DRA_INCONSISTENT_DIT	The replication operation encountered a database inconsistency.
8444	0x000020FC	ERROR_DS_DRA_CONNECTION_FAILED	The server specified for this replication operation could not be contacted.
8445	0x000020FD	ERROR_DS_DRA_BAD_INSTANCE_TYPE	The replication operation encountered an object with an invalid instance type.
8446	0x000020FE	ERROR_DS_DRA_OUT_OF_MEM	The replication operation failed to allocate memory.
8447	0x000020FF	ERROR_DS_DRA_MAIL_PROBLEM	The replication operation encountered an error with the mail system.

Error			Description
decimal	Hexadecimal	Name	
8448	0x00002100	ERROR_DS_DRA_REF_ALREADY_EXISTS	The replication reference information for the target server already exists.
8449	0x00002101	ERROR_DS_DRA_REF_NOT_FOUND	The replication reference information for the target server does not exist.
8450	0x00002102	ERROR_DS_DRA_OBJ_IS_REP_SOURCE	The naming context cannot be removed because it is replicated to another server.
8451	0x00002103	ERROR_DS_DRA_DB_ERROR	The replication operation encountered a database error.
8452	0x00002104	ERROR_DS_DRA_NO_REPLICA	The naming context is in the process of being removed or is not replicated from the specified server.
8453	0x00002105	ERROR_DS_DRA_ACCESS_DENIED	Replication access was denied.
8454	0x00002106	ERROR_DS_DRA_NOT_SUPPORTED	The requested operation is not supported by this version of the directory service.
8455	0x00002107	ERROR_DS_DRA_RPC_CANCELLED	The replication remote procedure call was cancelled.
8456	0x00002108	ERROR_DS_DRA_SOURCE_DISABLED	The source server is currently rejecting replication requests.
8457	0x00002109	ERROR_DS_DRA_SINK_DISABLED	The destination server is currently rejecting replication requests.
8458	0x0000210A	ERROR_DS_DRA_NAME_COLLISION	The replication operation failed due to a collision of object names.
8459	0x0000210B	ERROR_DS_DRA_SOURCE_REINSTALLED	The replication source has been reinstalled.
8460	0x0000210C	ERROR_DS_DRA_MISSING_PARENT	The replication operation failed because a required parent object is missing.
8461	0x0000210D	ERROR_DS_DRA_PREEMPTED	The replication operation was preempted.
8462	0x0000210E	ERROR_DS_DRA_ABANDON_SYNC	The replication synchronization attempt was abandoned because of a lack of updates.
8463	0x0000210F	ERROR_DS_DRA_SHUTDOWN	The replication operation was terminated because the system is shutting down.
8464	0x00002110	ERROR_DS_DRA_INCOMPATIBLE_PARTIAL_SET	The replication synchronization attempt failed as the destination partial attribute set is not a subset of source partial attribute set.
8465	0x00002111	ERROR_DS_DRA_SOURCE_IS_PARTIAL_REPLICA	The replication synchronization attempt failed because a master replica attempted to sync from a partial replica.
8466	0x00002112	ERROR_DS_DRA_EXTN_CONNECTION_FAILED	The server specified for this replication operation was contacted, but that server was unable to contact an additional server needed to complete the operation.
8467	0x00002113	ERROR_DS_INSTALL_SCHEMA_MISMATCH	The version of the Active Directory schema of the source forest is not compatible with the version of Active Directory on this computer. You must upgrade the operating system on a domain controller in the source forest before this computer can be added as a domain controller to that forest.
8468	0x00002114	ERROR_DS_DUP_LINK_ID	Schema update failed: An attribute with the same link identifier already exists.
8469	0x00002115	ERROR_DS_NAME_ERROR_RESOLVING	Name translation: Generic processing error.
8470	0x00002116	ERROR_DS_NAME_ERROR_NOT_FOUND	Name translation: Could not find the name or insufficient right to see name.
8471	0x00002117	ERROR_DS_NAME_ERROR_NOT_UNIQUE	Name translation: Input name mapped to more than one output name.
8472	0x00002118	ERROR_DS_NAME_ERROR_NO_MAPPING	Name translation: Input name found, but not the associated output format.
8473	0x00002119	ERROR_DS_NAME_ERROR_DOMAIN_ONLY	Name translation: Unable to resolve completely, only the domain was found.
8474	0x0000211A	ERROR_DS_NAME_ERROR_NO_SYNTACTICAL_MAPPING	Name translation: Unable to perform purely syntactical mapping at the client without going out to the wire.
8475	0x0000211B	ERROR_DS_CONSTRUCTED_ATT_MOD	Modification of a constructed att is not allowed.
8476	0x0000211C	ERROR_DS_WRONG_OM_OBJ_CLASS	The OM-Object-Class specified is incorrect for an attribute with the specified syntax.
8477	0x0000211D	ERROR_DS_DRA_REPL_PENDING	The replication request has been posted; waiting for reply.

Error			Description
decimal	Hexadecimal	Name	
8478	0x0000211E	ERROR_DS_DS_REQUIRED	The requested operation requires a directory service, and none was available.
8479	0x0000211F	ERROR_DS_INVALID_LDAP_DISPLAY_NAME	The LDAP display name of the class or attribute contains non-ASCII characters.
8480	0x00002120	ERROR_DS_NON_BASE_SEARCH	The requested search operation is only supported for base searches.
8481	0x00002121	ERROR_DS_CANT_RETRIEVE_ATTS	The search failed to retrieve attributes from the database.
8482	0x00002122	ERROR_DS_BACKLINK_WITHOUT_LINK	The schema update operation tried to add a backward link attribute that has no corresponding forward link.
8483	0x00002123	ERROR_DS_EPOCH_MISMATCH	Source and destination of a cross domain move do not agree on the object's epoch number. Either source or destination does not have the latest version of the object.
8484	0x00002124	ERROR_DS_SRC_NAME_MISMATCH	Source and destination of a cross domain move do not agree on the object's current name. Either source or destination does not have the latest version of the object.
8485	0x00002125	ERROR_DS_SRC_AND_DST_NC_IDENTICAL	Source and destination of a cross domain move operation are identical. Caller should use local move operation instead of cross domain move operation.
8486	0x00002126	ERROR_DS_DST_NC_MISMATCH	Source and destination for a cross domain move are not in agreement on the naming contexts in the forest. Either source or destination does not have the latest version of the Partitions container.
8487	0x00002127	ERROR_DS_NOT_AUTHORITIVE_FOR_DST_NC	Destination of a cross domain move is not authoritative for the destination naming context.
8488	0x00002128	ERROR_DS_SRC_GUID_MISMATCH	Source and destination of a cross domain move do not agree on the identity of the source object. Either source or destination does not have the latest version of the source object.
8489	0x00002129	ERROR_DS_CANT_MOVE_DELETED_OBJECT	Object being moved across domains is already known to be deleted by the destination server. The source server does not have the latest version of the source object.
8490	0x0000212A	ERROR_DS_PDC_OPERATION_IN_PROGRESS	Another operation which requires exclusive access to the PDC PSMO is already in progress.
8491	0x0000212B	ERROR_DS_CROSS_DOMAIN_CLEANUP_REQD	A cross domain move operation failed such that the two versions of the moved object exist - one each in the source and destination domains. The destination object needs to be removed to restore the system to a consistent state.
8492	0x0000212C	ERROR_DS_ILLEGAL_XDOM_MOVE_OPERATION	This object may not be moved across domain boundaries either because cross domain moves for this class are disallowed, or the object has some special characteristics, eg: trust account or restricted RID, which prevent its move.
8493	0x0000212D	ERROR_DS_CANT_WITH_ACCT_GROUP_MEMBERSHPS	Can't move objects with memberships across domain boundaries as once moved, this would violate the membership conditions of the account group. Remove the object from any account group memberships and retry.
8494	0x0000212E	ERROR_DS_NC_MUST_HAVE_NC_PARENT	A naming context head must be the immediate child of another naming context head, not of an interior node.
8495	0x0000212F	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE	The directory cannot validate the proposed naming context name because it does not hold a replica of the naming context above the proposed naming context. Please ensure that the domain naming master role is held by a server that is configured as a global catalog server, and that the server is up to date with its replication partners. (Applies only to Windows 2000 Domain Naming masters)
8496	0x00002130	ERROR_DS_DST_DOMAIN_NOT_NATIVE	Destination domain must be in native mode.

Error			Description
decimal	Hexadecimal	Name	
8497	0x00002131	ERROR_DS_MISSING_INFRASTRUCTURE_CONTAINER	The operation can not be performed because the server does not have an infrastructure container in the domain of interest.
8498	0x00002132	ERROR_DS_CANT_MOVE_ACCOUNT_GROUP	Cross-domain move of non-empty account groups is not allowed.
8499	0x00002133	ERROR_DS_CANT_MOVE_RESOURCE_GROUP	Cross-domain move of non-empty resource groups is not allowed.

Error			Description
decimal	Hexadecimal	Name	
8500	0x00002134	ERROR_DS_INVALID_SEARCH_FLAG	The search flags for the attribute are invalid. The ANR bit is valid only on attributes of Unicode or Teletex strings.
8501	0x00002135	ERROR_DS_NO_TREE_DELETE_ABOVE_NC	Tree deletions starting at an object which has an NC head as a descendant are not allowed.
8502	0x00002136	ERROR_DS_COULDNT_LOCK_TREE_FOR_DELETE	The directory service failed to lock a tree in preparation for a tree deletion because the tree was in use.
8503	0x00002137	ERROR_DS_COULDNT_IDENTIFY_OBJECTS_FOR_TREE_DELETE	The directory service failed to identify the list of objects to delete while attempting a tree deletion.
8504	0x00002138	ERROR_DS_SAM_INIT_FAILURE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Click OK to shut down the system and reboot into Directory Services Restore Mode. Check the event log for detailed information.
8505	0x00002139	ERROR_DS_SENSITIVE_GROUP_VIOLATION	Only an administrator can modify the membership list of an administrative group.
8506	0x0000213A	ERROR_DS_CANT_MOD_PRIMARYGROUPID	Cannot change the primary group ID of a domain controller account.
8507	0x0000213B	ERROR_DS_ILLEGAL_BASE_SCHEMA_MOD	An attempt is made to modify the base schema.
8508	0x0000213C	ERROR_DS_NONSAFE_SCHEMA_CHANGE	Adding a new mandatory attribute to an existing class, deleting a mandatory attribute from an existing class, or adding an optional attribute to the special class Top that is not a backlink attribute (directly or through inheritance, for example, by adding or deleting an auxiliary class) is not allowed.
8509	0x0000213D	ERROR_DS_SCHEMA_UPDATE_DISALLOWED	Schema update is not allowed on this DC because the DC is not the schema FSMO Role Owner.
8510	0x0000213E	ERROR_DS_CANT_CREATE_UNDER_SCHEMA	An object of this class cannot be created under the schema container. You can only create attribute-schema and class-schema objects under the schema container.
8511	0x0000213F	ERROR_DS_INSTALL_NO_SRC_SCH_VERSION	The replica/child install failed to get the objectVersion attribute on the schema container on the source DC. Either the attribute is missing on the schema container or the credentials supplied do not have permission to read it.
8512	0x00002140	ERROR_DS_INSTALL_NO_SCH_VERSION_IN_INIFILE	The replica/child install failed to read the objectVersion attribute in the SCHEMA section of the file schema.ini in the system32 directory.
8513	0x00002141	ERROR_DS_INVALID_GROUP_TYPE	The specified group type is invalid.
8514	0x00002142	ERROR_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN	Cannot nest global groups in a mixed domain if the group is security-enabled.
8515	0x00002143	ERROR_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN	Cannot nest local groups in a mixed domain if the group is security-enabled.
8516	0x00002144	ERROR_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER	A global group cannot have a local group as a member.
8517	0x00002145	ERROR_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER	A global group cannot have a universal group as a member.
8518	0x00002146	ERROR_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER	A universal group cannot have a local group as a member.
8519	0x00002147	ERROR_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER	A global group cannot have a cross-domain member.
8520	0x00002148	ERROR_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER	A local group cannot have another cross-domain local group as a member.
8521	0x00002149	ERROR_DS_HAVE_PRIMARY_MEMBERS	A group with primary members cannot change to a security-disabled group.
8522	0x0000214A	ERROR_DS_STRING_SD_CONVERSION_FAILED	The schema cache load failed to convert the string default SD on a class-schema object.
8523	0x0000214B	ERROR_DS_NAMING_MASTER_GC	Only DSAs configured to be Global Catalog servers should be allowed to hold the Domain Naming Master FSMO role. (Applies only to Windows 2000 servers)
8524	0x0000214C	ERROR_DS_LOOKUP_FAILURE	The DSA operation is unable to proceed because of a DNS lookup failure.

Error			Description
decimal	Hexadecimal	Name	
8525	0x0000214D	ERROR_DS_COULDNT_UPDATE_SPNS	While processing a change to the DNS Host Name for an object, the Service Principal Name values could not be kept in sync.
8526	0x0000214E	ERROR_DS_CANT_RETRIEVE_SD	The Security Descriptor attribute could not be read.
8527	0x0000214F	ERROR_DS_KEY_NOT_UNIQUE.	The object requested was not found, but an object with that key was found.
8528	0x00002150	ERROR_DS_WRONG_LINKED_ATT_SYNTAX	The syntax of the linked attributed being added is incorrect. Forward links can only have syntax 2.5.5.1, 2.5.5.7, and 2.5.5.14, and backlinks can only have syntax 2.5.5.1.
8529	0x00002151	ERROR_DS_SAM_NEED_BOOTKEY_PASSWORD	Security Account Manager needs to get the boot password.
8530	0x00002152	ERROR_DS_SAM_NEED_BOOTKEY_FLOPPY	Security Account Manager needs to get the boot key from floppy disk.
8531	0x00002153	ERROR_DS_CANT_START	Directory Service cannot start.
8532	0x00002154	ERROR_DS_INIT_FAILURE	Directory Services could not start.
8533	0x00002155	ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTION	The connection between client and server requires packet privacy or better.
8534	0x00002156	ERROR_DS_SOURCE_DOMAIN_IN_FOREST	The source domain may not be in the same forest as destination.
8535	0x00002157	ERROR_DS_DESTINATION_DOMAIN_NOT_IN_FOREST	The destination domain must be in the forest.
8536	0x00002158	ERROR_DS_DESTINATION_AUDITING_NOT_ENABLED	The operation requires that destination domain auditing be enabled.
8537	0x00002159	ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAIN	The operation couldn't locate a DC for the source domain.
8538	0x0000215A	ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER	The source object must be a group or user.
8539	0x0000215B	ERROR_DS_SRC_SID_EXISTS_IN_FOREST	The source object's SID already exists in destination forest.
8540	0x0000215C	ERROR_DS_SRC_AND_DST_OBJECT_CLASS_MISMATCH	The source and destination object must be of the same type.
8541	0x0000215D	ERROR_SAM_INIT_FAILURE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Click OK to shut down the system and reboot into Safe Mode. Check the event log for detailed information.
8542	0x0000215E	ERROR_DS_DRA_SCHEMA_INFO_SHIP	Schema information could not be included in the replication request.
8543	0x0000215F	ERROR_DS_DRA_SCHEMA_CONFLICT	The replication operation could not be completed due to a schema incompatibility.
8544	0x00002160	ERROR_DS_DRA_EARLIER_SCHEMA_CONLICT	The replication operation could not be completed due to a previous schema incompatibility.
8545	0x00002161	ERROR_DS_DRA_OBJ_NC_MISMATCH	The replication update could not be applied because either the source or the destination has not yet received information regarding a recent cross-domain move operation.
8546	0x00002162	ERROR_DS_NC_STILL_HAS_DSAS	The requested domain could not be deleted because there exist domain controllers that still host this domain.
8547	0x00002163	ERROR_DS_GC_REQUIRED	The requested operation can be performed only on a global catalog server.
8548	0x00002164	ERROR_DS_LOCAL_MEMBER_OF_LOCAL_ONLY	A local group can only be a member of other local groups in the same domain.
8549	0x00002165	ERROR_DS_NO_FPO_IN_UNIVERSAL_GROUPS	Foreign security principals cannot be members of universal groups.
8550	0x00002166	ERROR_DS_CANT_ADD_TO_GC	The attribute is not allowed to be replicated to the GC because of security reasons.
8551	0x00002167	ERROR_DS_NO_CHECKPOINT_WITH_PDC	The checkpoint with the PDC could not be taken because there are too many modifications being processed currently.
8552	0x00002168	ERROR_DS_SOURCE_AUDITING_NOT_ENABLED	The operation requires that source domain auditing be enabled.
8553	0x00002169	ERROR_DS_CANT_CREATE_IN_NONDOMAIN_NC	Security principal objects can only be created inside domain naming contexts.

Error			Description
decimal	Hexadecimal	Name	
8554	0x0000216A	ERROR_DS_INVALID_NAME_FOR_SPN	A Service Principal Name (SPN) could not be constructed because the provided hostname is not in the necessary format.
8555	0x0000216B	ERROR_DS_FILTER_USES_CONTRUCTED_ATTRS	A Filter was passed that uses constructed attributes.
8556	0x0000216C	ERROR_DS_UNICODEPWD_NOT_IN_QUOTES	The unicodePwd attribute value must be enclosed in double quotes.
8557	0x0000216D	ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED	Your computer could not be joined to the domain. You have exceeded the maximum number of computer accounts you are allowed to create in this domain. Contact your system administrator to have this limit reset or increased.
8558	0x0000216E	ERROR_DS_MUST_BE_RUN_ON_DST_DC	For security reasons, the operation must be run on the destination DC.
8559	0x0000216F	ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER	For security reasons, the source DC must be NT4SP4 or greater.
8560	0x00002170	ERROR_DS_CANT_TREE_DELETE_CRITICAL_OBJ	Critical Directory Service System objects cannot be deleted during tree delete operations. The tree delete may have been partially performed.
8561	0x00002171	ERROR_DS_INIT_FAILURE_CONSOLE	Directory Services could not start because of the following error: %1. Error Status: 0x%2. Please click OK to shutdown the system. You can use the recovery console to diagnose the system further.
8562	0x00002172	ERROR_DS_SAM_INIT_FAILURE_CONSOLE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Please click OK to shutdown the system. You can use the recovery console to diagnose the system further.
8563	0x00002173	ERROR_DS_FOREST_VERSION_TOO_HIGH	This version of Windows is too old to support the current directory forest behavior. You must upgrade the operating system on this server before it can become a domain controller in this forest.
8564	0x00002174	ERROR_DS_DOMAIN_VERSION_TOO_HIGH	This version of Windows is too old to support the current domain behavior. You must upgrade the operating system on this server before it can become a domain controller in this domain.
8565	0x00002175	ERROR_DS_FOREST_VERSION_TOO_LOW	This version of Windows no longer supports the behavior version in use in this directory forest. You must advance the forest behavior version before this server can become a domain controller in the forest.
8566	0x00002176	ERROR_DS_DOMAIN_VERSION_TOO_LOW	This version of Windows no longer supports the behavior version in use in this domain. You must advance the domain behavior version before this server can become a domain controller in the domain.
8567	0x00002177	ERROR_DS_INCOMPATIBLE_VERSION	The version of Windows is incompatible with the behavior version of the domain or forest.
8568	0x00002178	ERROR_DS_LOW_DSA_VERSION	The behavior version cannot be increased to the requested value because Domain Controllers still exist with versions lower than the requested value.
8569	0x00002179	ERROR_DS_NO_BEHAVIOR_VERSION_IN_MIXEDDOMAIN	The behavior version value cannot be increased while the domain is still in mixed domain mode. You must first change the domain to native mode before increasing the behavior version.
8570	0x0000217A	ERROR_DS_NOT_SUPPORTED_SORT_ORDER	The sort order requested is not supported.
8571	0x0000217B	ERROR_DS_NAME_NOT_UNIQUE	Found an object with a non unique name.
8572	0x0000217C	ERROR_DS_MACHINE_ACCOUNT_CREATED_PRENT4	The machine account was created pre-NT4. The account needs to be recreated.
8573	0x0000217D	ERROR_DS_OUT_OF_VERSION_STORE	The database is out of version store.
8574	0x0000217E	ERROR_DS_INCOMPATIBLE_CONTROLS_USED	Unable to continue operation because multiple conflicting controls were used.
8575	0x0000217F	ERROR_DS_NO_REF_DOMAIN	Unable to find a valid security descriptor reference domain for this partition.

Error			Description
decimal	Hexadecimal	Name	
8576	0x00002180	ERROR_DS_RESERVED_LINK_ID	Schema update failed: The link identifier is reserved.
8577	0x00002181	ERROR_DS_LINK_ID_NOT_AVAILABLE	Schema update failed: There are no link identifiers available.
8578	0x00002182	ERROR_DS_AG_CANT_HAVE_UNIVERSAL_MEMBER	An account group can not have a universal group as a member.
8579	0x00002183	ERROR_DS_MODIFYDN_DISALLOWED_BY_INSTANCE_TYPE	Rename or move operations on naming context heads or read-only objects are not allowed.
8580	0x00002184	ERROR_DS_NO_OBJECT_MOVE_IN_SCHEMA_NC	Move operations on objects in the schema naming context are not allowed.
8581	0x00002185	ERROR_DS_MODIFYDN_DISALLOWED_BY_FLAG	A system flag has been set on the object and does not allow the object to be moved or renamed.
8582	0x00002186	ERROR_DS_MODIFYDN_WRONG_GRANDPARENT	This object is not allowed to change its grandparent container. Moves are not forbidden on this object, but are restricted to sibling containers.
8583	0x00002187	ERROR_DS_NAME_ERROR_TRUST_REFERRAL	Unable to resolve completely, a referral to another forest is generated.
8584	0x00002188	ERROR_NOT_SUPPORTED_ON_STANDARD_SERVER	The requested action is not supported on standard server.
8585	0x00002189	ERROR_DS_CANT_ACCESS_REMOTE_PART_OF_AD	Could not access a partition of the Active Directory located on a remote server. Make sure at least one server is running for the partition in question.
8586	0x0000218A	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE_V2	The directory cannot validate the proposed naming context (or partition) name because it does not hold a replica nor can it contact a replica of the naming context above the proposed naming context. Please ensure that the parent naming context is properly registered in DNS, and at least one replica of this naming context is reachable by the Domain Naming master.
8587	0x0000218B	ERROR_DS_THREAD_LIMIT_EXCEEDED	The thread limit for this request was exceeded.
8588	0x0000218C	ERROR_DS_NOT_CLOSEST	The Global catalog server is not in the closet site.

Error			Description
decimal	Hexadecimal	Name	
9001	0x00002329	DNS_ERROR_RCODE_FORMAT_ERROR	DNS server unable to interpret format.
9002	0x0000232A	DNS_ERROR_RCODE_SERVER_FAILURE	DNS server failure.
9003	0x0000232B	DNS_ERROR_RCODE_NAME_ERROR	DNS name does not exist.
9004	0x0000232C	DNS_ERROR_RCODE_NOT_IMPLEMENTED	DNS request not supported by name server.
9005	0x0000232D	DNS_ERROR_RCODE_REFUSED	DNS operation refused.
9006	0x0000232E	DNS_ERROR_RCODE_YXDOMAIN	DNS name that ought not exist, does exist.
9007	0x0000232F	DNS_ERROR_RCODE_YXRRSET	DNS RR set that ought not exist, does exist.
9008	0x00002330	DNS_ERROR_RCODE_NXRRSET	DNS RR set that ought to exist, does not exist.
9009	0x00002331	DNS_ERROR_RCODE_NOTAUTH	DNS server not authoritative for zone.
9010	0x00002332	DNS_ERROR_RCODE_NOTZONE	DNS name in update or prereq is not in zone.
9016	0x00002338	DNS_ERROR_RCODE_BADSIG	DNS signature failed to verify.
9017	0x00002339	DNS_ERROR_RCODE_BADKEY	DNS bad key.
9018	0x0000233A	DNS_ERROR_RCODE_BADTIME	DNS signature validity expired.
9501	0x0000251D	DNS_INFO_NO_RECORDS	No records found for given DNS query.
9502	0x0000251E	DNS_ERROR_BAD_PACKET	Bad DNS packet.
9503	0x0000251F	DNS_ERROR_NO_PACKET	No DNS packet.
9504	0x00002520	DNS_ERROR_RCODE	DNS error, check rcode.
9505	0x00002521	DNS_ERROR_UNSECURE_PACKET	Unsecured DNS packet.
9551	0x0000254F	DNS_ERROR_INVALID_TYPE	Invalid DNS type.
9552	0x00002550	DNS_ERROR_INVALID_IP_ADDRESS	Invalid IP address.
9553	0x00002551	DNS_ERROR_INVALID_PROPERTY	Invalid property.
9554	0x00002552	DNS_ERROR_TRY_AGAIN_LATER	Try DNS operation again later.
9555	0x00002553	DNS_ERROR_NOT_UNIQUE	Record for given name and type is not unique.
9556	0x00002554	DNS_ERROR_NON_RFC_NAME	DNS name does not comply with RFC specifications.
9557	0x00002555	DNS_STATUS_FQDN	DNS name is a fully-qualified DNS name.
9558	0x00002556	DNS_STATUS_DOTTED_NAME	DNS name is dotted (multi-label).
9559	0x00002557	DNS_STATUS_SINGLE_PART_NAME	DNS name is a single-part name.
9560	0x00002558	DNS_ERROR_INVALID_NAME_CHAR	DSN name contains an invalid character.
9561	0x00002559	DNS_ERROR_NUMERIC_NAME	DNS name is entirely numeric.
9601	0x00002581	DNS_ERROR_ZONE_DOES_NOT_EXIST	DNS zone does not exist.
9602	0x00002582	DNS_ERROR_NO_ZONE_INFO	DNS zone information not available.
9603	0x00002583	DNS_ERROR_INVALID_ZONE_OPERATION	Invalid operation for DNS zone.
9604	0x00002584	DNS_ERROR_ZONE_CONFIGURATION_ERROR	Invalid DNS zone configuration.
9605	0x00002585	DNS_ERROR_ZONE_HAS_NO_SOA_RECORD	DNS zone has no start of authority (SOA) record.
9606	0x00002586	DNS_ERROR_ZONE_HAS_NO_NS_RECORDS	DNS zone has no name server (NS) record.
9607	0x00002587	DNS_ERROR_ZONE_LOCKED	DNS zone is locked.
9608	0x00002588	DNS_ERROR_ZONE_CREATION_FAILED	DNS zone creation failed.
9609	0x00002589	DNS_ERROR_ZONE_ALREADY_EXISTS	DNS zone already exists.
9610	0x0000258A	DNS_ERROR_AUTOZONE_ALREADY_EXISTS	DNS automatic zone already exists.
9611	0x0000258B	DNS_ERROR_INVALID_ZONE_TYPE	Invalid DNS zone type.
9612	0x0000258C	DNS_ERROR_SECONDARY_REQUIRES_MASTER_IP	Secondary DNS zone requires master IP address.
9613	0x0000258D	DNS_ERROR_ZONE_NOT_SECONDARY	DNS zone not secondary.
9614	0x0000258E	DNS_ERROR_NEED_SECONDARY_ADDRESSES	Need secondary IP address.
9615	0x0000258F	DNS_ERROR_WINS_INIT_FAILED	WINS initialization failed.
9616	0x00002590	DNS_ERROR_NEED_WINS_SERVERS	Need WINS servers.
9617	0x00002591	DNS_ERROR_NBSTAT_INIT_FAILED	NBTSTAT initialization call failed.
9618	0x00002592	DNS_ERROR_SOA_DELETE_INVALID	Invalid delete of start of authority (SOA)
9619	0x00002593	DNS_ERROR_FORWARDER_ALREADY_EXISTS	A conditional forwarding zone already exists for that name.
9651	0x000025B3	DNS_ERROR_PRIMARY_REQUIRES_DATAFILE	Primary DNS zone requires datafile.
9652	0x000025B4	DNS_ERROR_INVALID_DATAFILE_NAME	Invalid datafile name for DNS zone.
9653	0x000025B5	DNS_ERROR_DATAFILE_OPEN_FAILURE	Failed to open datafile for DNS zone.
9654	0x000025B6	DNS_ERROR_FILE_WRITEBACK_FAILED	Failed to write datafile for DNS zone.
9655	0x000025B7	DNS_ERROR_DATAFILE_PARSING	Failure while reading datafile for DNS zone.
9701	0x000025E5	DNS_ERROR_RECORD_DOES_NOT_EXIST	DNS record does not exist.
9702	0x000025E6	DNS_ERROR_RECORD_FORMAT	DNS record format error.

Error			Description
decimal	Hexadecimal	Name	
9703	0x000025E7	DNS_ERROR_NODE_CREATION_FAILED	Node creation failure in DNS.
9704	0x000025E8	DNS_ERROR_UNKNOWN_RECORD_TYPE	Unknown DNS record type.
9705	0x000025E9	DNS_ERROR_RECORD_TIMED_OUT	DNS record timed out.
9706	0x000025EA	DNS_ERROR_NAME_NOT_IN_ZONE	Name not in DNS zone.
9707	0x000025EB	DNS_ERROR_CNAME_LOOP	CNAME loop detected.
9708	0x000025EC	DNS_ERROR_NODE_IS_CNAME	Node is a CNAME DNS record.
9709	0x000025ED	DNS_ERROR_CNAME_COLLISION	A CNAME record already exists for given name.
9710	0x000025EE	DNS_ERROR_RECORD_ONLY_AT_ZONE_ROOT	Record only at DNS zone root.
9711	0x000025EF	DNS_ERROR_RECORD_ALREADY_EXISTS	DNS record already exists.
9712	0x000025F0	DNS_ERROR_SECONDARY_DATA	Secondary DNS zone data error.
9713	0x000025F1	DNS_ERROR_NO_CREATE_CACHE_DATA	Could not create DNS cache data.
9714	0x000025F2	DNS_ERROR_NAME_DOES_NOT_EXIST	DNS name does not exist.
9715	0x000025F3	DNS_WARNING_PTR_CREATE_FAILED	Could not create pointer (PTR) record.
9716	0x000025F4	DNS_WARNING_DOMAIN_UNDELETED	DNS domain was undeleted.
9717	0x000025F5	DNS_ERROR_DS_UNAVAILABLE	The directory service is unavailable.
9718	0x000025F6	DNS_ERROR_DS_ZONE_ALREADY_EXISTS	DNS zone already exists in the directory service.
9719	0x000025F7	DNS_ERROR_NO_BOOTFILE_IF_DS_ZONE	DNS server not creating or reading the boot file for the directory service integrated DNS zone.
9751	0x00002617	DNS_INFO_AXFR_COMPLETE	DNS AXFR (zone transfer) complete.
9752	0x00002618	DNS_ERROR_AXFR	DNS zone transfer failed.
9753	0x00002619	DNS_INFO_ADDED_LOCAL_WINS	Added local WINS server.
9801	0x00002649	DNS_STATUS_CONTINUE_NEEDED	Secure update call needs to continue update request.
9851	0x0000267B	DNS_ERROR_NO_TCPIP	TCP/IP network protocol not installed.
9852	0x0000267C	DNS_ERROR_NO_DNS_SERVERS	No DNS servers configured for local system.
9901	0x000026AD	DNS_ERROR_DP_DOES_NOT_EXIST	The specified directory partition does not exist.
9902	0x000026AE	DNS_ERROR_DP_ALREADY_EXISTS	The specified directory partition already exists.
9903	0x000026AF	DNS_ERROR_DP_NOT_ENLISTED	The DS is not enlisted in the specified directory partition.
9904	0x000026B0	DNS_ERROR_DP_ALREADY_ENLISTED	The DS is already enlisted in the specified directory partition.

Error			Description
decimal	Hexadecimal	Name	
10004	0x00002714	WSAEINTR	A blocking operation was interrupted by a call to WSACancelBlockingCall.
10009	0x00002719	WSAEBADF	The file handle supplied is not valid.
10013	0x0000271D	WSAEACCES	An attempt was made to access a socket in a way forbidden by its access permissions.
10014	0x0000271E	WSAEFAULT	The system detected an invalid pointer address in attempting to use a pointer argument in a call.
10022	0x00002726	WSAEINVAL	An invalid argument was supplied.
10024	0x00002728	WSAEMFILE	Too many open sockets.
10035	0x00002733	WSAEWOULDBLOCK	A non-blocking socket operation could not be completed immediately.
10036	0x00002734	WSAEINPROGRESS	A blocking operation is currently executing.
10037	0x00002735	WSAEALREADY	An operation was attempted on a non-blocking socket that already had an operation in progress.
10038	0x00002736	WSAENOTSOCK	An operation was attempted on something that is not a socket.
10039	0x00002737	WSAEDESTADDRREQ	A required address was omitted from an operation on a socket.
10040	0x00002738	WSAEMSGSIZE	A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram into was smaller than the datagram itself.
10041	0x00002739	WSAEPROTOTYPE	A protocol was specified in the socket function call that does not support the semantics of the socket type requested.
10042	0x0000273A	WSAENOPROTOOPT	An unknown, invalid, or unsupported option or level was specified in a getsockopt or setsockopt call.
10043	0x0000273B	WSAEPROTONOSUPPORT	The requested protocol has not been configured into the system, or no implementation for it exists.
10044	0x0000273C	WSAESOCKTNOSUPPORT	The support for the specified socket type does not exist in this address family.
10045	0x0000273D	WSAEOPNOTSUPP	The attempted operation is not supported for the type of object referenced.
10046	0x0000273E	WSAEPFNOSUPPORT	The protocol family has not been configured into the system or no implementation for it exists.
10047	0x0000273F	WSAEAFNOSUPPORT	An address incompatible with the requested protocol was used.
10048	0x00002740	WSAEADDRINUSE	Only one usage of each socket address (protocol/ network address/port) is normally permitted.
10049	0x00002741	WSAEADDRNOTAVAIL	The requested address is not valid in its context.
10050	0x00002742	WSAENETDOWN	A socket operation encountered a dead network.
10051	0x00002743	WSAENETUNREACH	A socket operation was attempted to an unreachable network.
10052	0x00002744	WSAENETRESET	The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress.
10053	0x00002745	WSAECONNABORTED	An established connection was aborted by the software in your host machine.
10054	0x00002746	WSAECONNRESET	An existing connection was forcibly closed by the remote host.
10055	0x00002747	WSAENOBUFS	An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.
10056	0x00002748	WSAEISCONN	A connect request was made on an already connected socket.
10057	0x00002749	WSAENOTCONN	A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using a sendto call) no address was supplied.
10058	0x0000274A	WSAESHUTDOWN	A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous shutdown call.
10059	0x0000274B	WSAETOOMANYREFS	Too many references to some kernel object.

Error			Description
decimal	Hexadecimal	Name	
10060	0x0000274C	WSAETIMEDOUT	A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.
10061	0x0000274D	WSAECONNREFUSED	No connection could be made because the target machine actively refused it.
10062	0x0000274E	WSAELOOP	Cannot translate name.
10063	0x0000274F	WSAENAMETOOLONG	Name component or name was too long.
10064	0x00002750	WSAEHOSTDOWN	A socket operation failed because the destination host was down.
10065	0x00002751	WSAEHOSTUNREACH	A socket operation was attempted to an unreachable host.
10066	0x00002752	WSAENOTEMPTY	Cannot remove a directory that is not empty.
10067	0x00002753	WSAEPROCLIM	A Windows Sockets implementation may have a limit on the number of applications that may use it simultaneously.
10068	0x00002754	WSAEUSERS	Ran out of quota.
10069	0x00002755	WSAEDQUOT	Ran out of disk quota.
10070	0x00002756	WSAESTALE	File handle reference is no longer available.
10071	0x00002757	WSAEREMOTE	Item is not available locally.
10091	0x0000276B	WSASYSNOTREADY	WSAStartup cannot function at this time because the underlying system it uses to provide network services is currently unavailable.
10092	0x0000276C	WSAVERNOTSUPPORTED	The Windows Sockets version requested is not supported.
10093	0x0000276D	WSANOTINITIALISED	Either the application has not called WSAStartup, or WSAStartup failed.
10101	0x00002775	WSAEDISCON	Returned by WSARecv or WSARecvFrom to indicate the remote party has initiated a graceful shutdown sequence.
10102	0x00002776	WSAENOMORE	No more results can be returned by WSALookupServiceNext.
10103	0x00002777	WSAECANCELLED	A call to WSALookupServiceEnd was made while this call was still processing. The call has been canceled.
10104	0x00002778	WSAEINVALIDPROCTABLE	The procedure call table is invalid.
10105	0x00002779	WSAEINVALIDPROVIDER	The requested service provider is invalid.
10106	0x0000277A	WSAEPROVIDERFAILEDINIT	The requested service provider could not be loaded or initialized.
10107	0x0000277B	WSASYSCALLFAILURE	A system call that should never fail has failed.
10108	0x0000277C	WSASERVICE_NOT_FOUND	No such service is known. The service cannot be found in the specified name space.
10109	0x0000277D	WSATYPE_NOT_FOUND	The specified class was not found.
10110	0x0000277E	WSA_E_NO_MORE	No more results can be returned by WSALookupServiceNext.
10111	0x0000277F	WSA_E_CANCELLED	A call to WSALookupServiceEnd was made while this call was still processing. The call has been canceled.
10112	0x00002780	WSAEREFUSED	A database query failed because it was actively refused.
11001	0x00002AF9	WSAHOST_NOT_FOUND	No such host is known.
11002	0x00002AFA	WSATRY_AGAIN	This is usually a temporary error during hostname resolution and means that the local server did not receive a response from an authoritative server.
11003	0x00002AFB	WSANO_RECOVERY	A non-recoverable error occurred during a database lookup.
11004	0x00002AFC	WSANO_DATA	The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for.
11005	0x00002AFD	WSA_QOS_RECEIVERS	At least one reserve has arrived.
11006	0x00002AFE	WSA_QOS_SENDERS	At least one path has arrived.
11007	0x00002AFF	WSA_QOS_NO_SENDERS	There are no senders.
11008	0x00002B00	WSA_QOS_NO_RECEIVERS	There are no receivers.
11009	0x00002B01	WSA_QOS_REQUEST_CONFIRMED	Reserve has been confirmed.

Error			Description
decimal	Hexadecimal	Name	
11010	0x00002B02	WSA_QOS_ADMISSION_FAILURE	Error due to lack of resources.
11011	0x00002B03	WSA_QOS_POLICY_FAILURE	Rejected for administrative reasons - bad credentials.
11012	0x00002B04	WSA_QOS_BAD_STYLE	Unknown or conflicting style.
11013	0x00002B05	WSA_QOS_BAD_OBJECT	Problem with some part of the filterspec or providerspecific buffer in general.
11014	0x00002B06	WSA_QOS_TRAFFIC_CTRL_ERROR	Problem with some part of the flowspec.
11015	0x00002B07	WSA_QOS_GENERIC_ERROR	General QOS error.
11016	0x00002B08	WSA_QOS_ESERVICETYPE	An invalid or unrecognized service type was found in the flowspec.
11017	0x00002B09	WSA_QOS_EFLOWSPEC	An invalid or inconsistent flowspec was found in the QOS structure.
11018	0x00002B0A	WSA_QOS_EPROVSPECBUF	Invalid QOS provider-specific buffer.
11019	0x00002B0B	WSA_QOS_EFILTERSTYLE	An invalid QOS filter style was used.
11020	0x00002B0C	WSA_QOS_EFILTERTYPE	An invalid QOS filter type was used.
11021	0x00002B0D	WSA_QOS_EFILTERCOUNT	An incorrect number of QOS FILTERSPECs were specified in the FLOWDESCRIPTOR.
11022	0x00002B0E	WSA_QOS_EOBJLENGTH	An object with an invalid ObjectLength field was specified in the QOS provider-specific buffer.
11023	0x00002B0F	WSA_QOS_EFLOWCOUNT	An incorrect number of flow descriptors was specified in the QOS structure.
11024	0x00002B10	WSA_QOS_EUNKNOWNPSOBJ	An unrecognized object was found in the QOS provider-specific buffer.
11025	0x00002B11	WSA_QOS_EPOLICYOBJ	An invalid policy object was found in the QOS provider-specific buffer.
11026	0x00002B12	WSA_QOS_EFLOWDESC	An invalid QOS flow descriptor was found in the flow descriptor list.
11027	0x00002B13	WSA_QOS_EPSFLOWSPEC	An invalid or inconsistent flowspec was found in the QOS provider-specific buffer.
11028	0x00002B14	WSA_QOS_EPSFILTERSPEC	An invalid FILTERSPEC was found in the QOS provider-specific buffer.
11029	0x00002B15	WSA_QOS_ESDMODEOBJ	An invalid shape discard mode object was found in the QOS provider-specific buffer.
11030	0x00002B16	WSA_QOS_ESHAPERATEOBJ	An invalid shaping rate object was found in the QOS provider-specific buffer.
11031	0x00002B17	WSA_QOS_RESERVED_PETYPE	A reserved policy element was found in the QOS provider-specific buffer.

Error			Description
decimal	Hexadecimal	Name	
12000	0x00002EE0	ERROR_SXS_SECTION_NOT_FOUND	The requested section was not present in the activation context.
12001	0x00002EE1	ERROR_SXS_CANT_GEN_ACTCTX	This application has failed to start because the application configuration is incorrect. Reinstalling the application may fix this problem.
12002	0x00002EE2	ERROR_SXS_INVALID_ACTCTXDATA_FORMAT	The application binding data format is invalid.
12003	0x00002EE3	ERROR_SXS_ASSEMBLY_NOT_FOUND	The referenced assembly is not installed on your system.
12004	0x00002EE4	ERROR_SXS_MANIFEST_FORMAT_ERROR	The manifest file does not begin with the required tag and format information.
12005	0x00002EE5	ERROR_SXS_MANIFEST_PARSE_ERROR	The manifest file contains one or more syntax errors.
12006	0x00002EE6	ERROR_SXS_ACTIVATION_CONTEXT_DISABLED	The application attempted to activate a disabled activation context.
12007	0x00002EE7	ERROR_SXS_KEY_NOT_FOUND	The requested lookup key was not found in any active activation context.
12008	0x00002EE8	ERROR_SXS_VERSION_CONFLICT	A component version required by the application conflicts with another component version already active.
12009	0x00002EE9	ERROR_SXS_WRONG_SECTION_TYPE	The type requested activation context section does not match the query API used.
12010	0x00002EEA	ERROR_SXS_THREAD_QUERIES_DISABLED	Lack of system resources has required isolated activation to be disabled for the current thread of execution.
12011	0x00002EEB	ERROR_SXS_PROCESS_DEFAULT_ALREADY_SET	An attempt to set the process default activation context failed because the process default activation context was already set.
12012	0x00002EEC	ERROR_SXS_UNKNOWN_ENCODING_GROUP	The encoding group identifier specified is not recognized.
12013	0x00002EED	ERROR_SXS_UNKNOWN_ENCODING	The encoding requested is not recognized.
12014	0x00002EEE	ERROR_SXS_INVALID_XML_NAMESPACE_URI	The manifest contains a reference to an invalid URI.
12015	0x00002EEF	ERROR_SXS_ROOT_MANIFEST_DEPENDENCY_NOT_INSTALLED	The application manifest contains a reference to a dependent assembly which is not installed.
12016	0x00002EF0	ERROR_SXS_LEAF_MANIFEST_DEPENDENCY_NOT_INSTALLED	The manifest for an assembly used by the application has a reference to a dependent assembly which is not installed.
12017	0x00002EF1	ERROR_SXS_INVALID_ASSEMBLY_IDENTITY_ATTRIBUTE	The manifest contains an attribute for the assembly identity which is not valid.
12018	0x00002EF2	ERROR_SXS_MANIFEST_MISSING_REQUIRED_DEFAULT_NAMESPACE	The manifest is missing the required default namespace specification on the assembly element.
12019	0x00002EF3	ERROR_SXS_MANIFEST_INVALID_REQUIRED_DEFAULT_NAMESPACE	The manifest has a default namespace specified on the assembly element but its value is not "urn:schemas-microsoft-com:asm.v1".
12020	0x00002EF4	ERROR_SXS_PRIVATE_MANIFEST_CROSS_PATH_WITH_REPARSE_POINT	The private manifest probe has crossed the reparse-point-associated path.
12021	0x00002EF5	ERROR_SXS_DUPLICATE_DLL_NAME	Two or more components referenced directly or indirectly by the application manifest have files by the same name.
12022	0x00002EF6	ERROR_SXS_DUPLICATE_WINDOWCLASS_NAME	Two or more components referenced directly or indirectly by the application manifest have window classes with the same name.
12023	0x00002EF7	ERROR_SXS_DUPLICATE_CLSID	Two or more components referenced directly or indirectly by the application manifest have the same COM server CLSIDs.
12024	0x00002EF8	ERROR_SXS_DUPLICATE_IID	Two or more components referenced directly or indirectly by the application manifest have proxies for the same COM interface IIDs.
12025	0x00002EF9	ERROR_SXS_DUPLICATE_TLBID	Two or more components referenced directly or indirectly by the application manifest have the same COM type library TLBIDs.
12026	0x00002EFA	ERROR_SXS_DUPLICATE_PROGID	Two or more components referenced directly or indirectly by the application manifest have the same COM ProgIDs.

Error			Description
decimal	Hexadecimal	Name	
12027	0x00002EFB	ERROR_SXS_DUPLICATE_ASSEMBLY_NAME	Two or more components referenced directly or indirectly by the application manifest are different versions of the same component which is not permitted.
12028	0x00002EFC	ERROR_SXS_FILE_HASH_MISMATCH	A component's file does not match the verification information present in the component manifest.
12029	0x00002EFD	ERROR_SXS_POLICY_PARSE_ERROR	The policy manifest contains one or more syntax errors.
12030	0x00002EFE	ERROR_SXS_XML_E_MISSINGQUOTE	Manifest Parse Error : A string literal was expected, but no opening quote character was found.
12031	0x00002EFF	ERROR_SXS_XML_E_COMMENTSYNTAX	Manifest Parse Error : Incorrect syntax was used in a comment.
12032	0x00002F00	ERROR_SXS_XML_E_BADSTARTNAMECHAR	Manifest Parse Error : A name was started with an invalid character.
12033	0x00002F01	ERROR_SXS_XML_E_BADNAMECHAR	Manifest Parse Error : A name contained an invalid character.
12034	0x00002F02	ERROR_SXS_XML_E_BADCHARINSTRING	Manifest Parse Error : A string literal contained an invalid character.
12035	0x00002F03	ERROR_SXS_XML_E_XMLDECLSYNTAX	Manifest Parse Error : Invalid syntax for an XML declaration.
12036	0x00002F04	ERROR_SXS_XML_E_BADCHARDATA	Manifest Parse Error : An invalid character was found in text content.
12037	0x00002F05	ERROR_SXS_XML_E_MISSINGWHITESPACE	Manifest Parse Error : Required white space was missing.
12038	0x00002F06	ERROR_SXS_XML_E_EXPECTINGTAGEND	Manifest Parse Error : The character '>' was expected.
12039	0x00002F07	ERROR_SXS_XML_E_MISSINGSEMICOLON	Manifest Parse Error : A semi colon character was expected.
12040	0x00002F08	ERROR_SXS_XML_E_UNBALANCEDPAREN	Manifest Parse Error : Unbalanced parentheses.
12041	0x00002F09	ERROR_SXS_XML_E_INTERNALERROR	Manifest Parse Error : Internal error.
12042	0x00002F0A	ERROR_SXS_XML_E_UNEXPECTED_WHITESPACE	Manifest Parse Error : White space is not allowed at this location.
12043	0x00002F0B	ERROR_SXS_XML_E_INCOMPLETE_ENCODING	Manifest Parse Error : End of file reached in invalid state for current encoding.
12044	0x00002F0C	ERROR_SXS_XML_E_MISSING_PAREN	Manifest Parse Error : Missing parenthesis.
12045	0x00002F0D	ERROR_SXS_XML_E_EXPECTINGCLOSEQUOTE	Manifest Parse Error : A single or double closing quote character (\' or \") is missing.
12046	0x00002F0E	ERROR_SXS_XML_E_MULTIPLE_COLONS	Manifest Parse Error : Multiple colons are not allowed in a name.
12047	0x00002F0F	ERROR_SXS_XML_E_INVALID_DECIMAL	Manifest Parse Error : Invalid character for decimal digit.
12048	0x00002F10	ERROR_SXS_XML_E_INVALID_HEXIDECIMAL	Manifest Parse Error : Invalid character for hexadecimal digit.
12049	0x00002F11	ERROR_SXS_XML_E_INVALID_UNICODE	Manifest Parse Error : Invalid Unicode character value for this platform.
12050	0x00002F12	ERROR_SXS_XML_E_WHITESPACEORQUESTIONMARK	Manifest Parse Error : Expecting white space or '?'.
12051	0x00002F13	ERROR_SXS_XML_E_UNEXPECTEDENDTAG	Manifest Parse Error : End tag was not expected at this location.
12052	0x00002F14	ERROR_SXS_XML_E_UNCLOSEDTAG	Manifest Parse Error : The following tags were not closed: %1.
12053	0x00002F15	ERROR_SXS_XML_E_DUPLICATEATTRIBUTE	Manifest Parse Error : Duplicate attribute.
12054	0x00002F16	ERROR_SXS_XML_E_MULTIPLEROOTS	Manifest Parse Error : Only one top level element is allowed in an XML document.
12055	0x00002F17	ERROR_SXS_XML_E_INVALIDATROOTLEVEL	Manifest Parse Error : Invalid at the top level of the document.
12056	0x00002F18	ERROR_SXS_XML_E_BADXMLDECL	Manifest Parse Error : Invalid XML declaration.
12057	0x00002F19	ERROR_SXS_XML_E_MISSINGROOT	Manifest Parse Error : XML document must have a top level element.
12058	0x00002F1A	ERROR_SXS_XML_E_UNEXPECTEDEOF	Manifest Parse Error : Unexpected end of file.
12059	0x00002F1B	ERROR_SXS_XML_E_BADPEREFINSUBSET	Manifest Parse Error : Parameter entities cannot be used inside markup declarations in an internal subset.
12060	0x00002F1C	ERROR_SXS_XML_E_UNCLOSEDSTARTTAG	Manifest Parse Error : Element was not closed.

Error			Description
decimal	Hexadecimal	Name	
12061	0x00002F1D	ERROR_SXS_XML_E_UNCLOSEDENDTAG	Manifest Parse Error : End element was missing the character '>'.
12062	0x00002F1E	ERROR_SXS_XML_E_UNCLOSEDSTRING	Manifest Parse Error : A string literal was not closed.
12063	0x00002F1F	ERROR_SXS_XML_E_UNCLOSEDCOMMENT	Manifest Parse Error : A comment was not closed.
12064	0x00002F20	ERROR_SXS_XML_E_UNCLOSEDDECL	Manifest Parse Error : A declaration was not closed.
12065	0x00002F21	ERROR_SXS_XML_E_UNCLOSEDCDATA	Manifest Parse Error : A CDATA section was not closed.
12066	0x00002F22	ERROR_SXS_XML_E_RESERVEDNAMESPACE	Manifest Parse Error : The namespace prefix is not allowed to start with the reserved string "xml".
12067	0x00002F23	ERROR_SXS_XML_E_INVALIDENCODING	Manifest Parse Error : System does not support the specified encoding.
12068	0x00002F24	ERROR_SXS_XML_E_INVALIDSWITCH	Manifest Parse Error : Switch from current encoding to specified encoding not supported.
12069	0x00002F25	ERROR_SXS_XML_E_BADXMLCASE	Manifest Parse Error : The name 'xml' is reserved and must be lower case.
12070	0x00002F26	ERROR_SXS_XML_E_INVALID_STANDALONE	Manifest Parse Error : The standalone attribute must have the value 'yes' or 'no'.
12071	0x00002F27	ERROR_SXS_XML_E_UNEXPECTED_STANDALONE	Manifest Parse Error : The standalone attribute cannot be used in external entities.
12072	0x00002F28	ERROR_SXS_XML_E_INVALID_VERSION	Manifest Parse Error : Invalid version number.
12073	0x00002F29	ERROR_SXS_XML_E_MISSINGEQUALS	Manifest Parse Error : Missing equals sign between attribute and attribute value.
13000	0x000032C8	ERROR_IPSEC_QM_POLICY_EXISTS	The specified quick mode policy already exists.
13001	0x000032C9	ERROR_IPSEC_QM_POLICY_NOT_FOUND	The specified quick mode policy was not found.
13002	0x000032CA	ERROR_IPSEC_QM_POLICY_IN_USE	The specified quick mode policy is being used.
13003	0x000032CB	ERROR_IPSEC_MM_POLICY_EXISTS	The specified main mode policy already exists.
13004	0x000032CC	ERROR_IPSEC_MM_POLICY_NOT_FOUND	The specified main mode policy was not found.
13005	0x000032CD	ERROR_IPSEC_MM_POLICY_IN_USE	The specified main mode policy is being used.
13006	0x000032CE	ERROR_IPSEC_MM_FILTER_EXISTS	The specified main mode filter already exists.
13007	0x000032CF	ERROR_IPSEC_MM_FILTER_NOT_FOUND	The specified main mode filter was not found.
13008	0x000032D0	ERROR_IPSEC_TRANSPORT_FILTER_EXISTS	The specified transport mode filter already exists.
13009	0x000032D1	ERROR_IPSEC_TRANSPORT_FILTER_NOT_FOUND	The specified transport mode filter does not exist.
13010	0x000032D2	ERROR_IPSEC_MM_AUTH_EXISTS	The specified main mode authentication list exists.
13011	0x000032D3	ERROR_IPSEC_MM_AUTH_NOT_FOUND	The specified main mode authentication list was not found.
13012	0x000032D4	ERROR_IPSEC_MM_AUTH_IN_USE	The specified quick mode policy is being used.
13013	0x000032D5	ERROR_IPSEC_DEFAULT_MM_POLICY_NOT_FOUND	The specified main mode policy was not found.
13014	0x000032D6	ERROR_IPSEC_DEFAULT_MM_AUTH_NOT_FOUND	The specified quick mode policy was not found.
13015	0x000032D7	ERROR_IPSEC_DEFAULT_QM_POLICY_NOT_FOUND	The manifest file contains one or more syntax errors.
13016	0x000032D8	ERROR_IPSEC_TUNNEL_FILTER_EXISTS	The application attempted to activate a disabled activation context.
13017	0x000032D9	ERROR_IPSEC_TUNNEL_FILTER_NOT_FOUND	The requested lookup key was not found in any active activation context.
13018	0x000032DA	ERROR_IPSEC_MM_FILTER_PENDING_DELETION	The Main Mode filter is pending deletion.
13019	0x000032DB	ERROR_IPSEC_TRANSPORT_FILTER_PENDING_DELETION	The transport filter is pending deletion.
13020	0x000032DC	ERROR_IPSEC_TUNNEL_FILTER_PENDING_DELETION	The tunnel filter is pending deletion.
13021	0x000032DD	ERROR_IPSEC_MM_POLICY_PENDING_DELETION	The Main Mode policy is pending deletion.
13022	0x000032DE	ERROR_IPSEC_MM_AUTH_PENDING_DELETION	The Main Mode authentication bundle is pending deletion.
13023	0x000032DF	ERROR_IPSEC_QM_POLICY_PENDING_DELETION	The Quick Mode policy is pending deletion.

Error			Description
decimal	Hexadecimal	Name	
13801	0x000035E9	ERROR_IPSEC_IKE_AUTH_FAIL	IKE authentication credentials are unacceptable.
13802	0x000035EA	ERROR_IPSEC_IKE_ATTRIB_FAIL	IKE security attributes are unacceptable.
13803	0x000035EB	ERROR_IPSEC_IKE_NEGOTIATION_PENDING	IKE Negotiation in progress.
13804	0x000035EC	ERROR_IPSEC_IKE_GENERAL_PROCESSING_ERROR	General processing error.
13805	0x000035ED	ERROR_IPSEC_IKE_TIMED_OUT	Negotiation timed out.
13806	0x000035EE	ERROR_IPSEC_IKE_NO_CERT	IKE failed to find valid machine certificate.
13807	0x000035EF	ERROR_IPSEC_IKE_SA_DELETED	IKE SA deleted by peer before establishment completed.
13808	0x000035F0	ERROR_IPSEC_IKE_SA_REAPED	IKE SA deleted before establishment completed.
13809	0x000035F1	ERROR_IPSEC_IKE_MM_ACQUIRE_DROP	Negotiation request sat in Queue too long.
13810	0x000035F2	ERROR_IPSEC_IKE_QM_ACQUIRE_DROP	Negotiation request sat in Queue too long.
13811	0x000035F3	ERROR_IPSEC_IKE_QUEUE_DROP_MM	Negotiation request sat in Queue too long.
13812	0x000035F4	ERROR_IPSEC_IKE_QUEUE_DROP_NO_MM	Negotiation request sat in Queue too long.
13813	0x000035F5	ERROR_IPSEC_IKE_DROP_NO_RESPONSE	No response from peer.
13814	0x000035F6	ERROR_IPSEC_IKE_MM_DELAY_DROP	Negotiation took too long.
13815	0x000035F7	ERROR_IPSEC_IKE_QM_DELAY_DROP	Negotiation took too long.
13816	0x000035F8	ERROR_IPSEC_IKE_ERROR	Unknown error occurred.
13817	0x000035F9	ERROR_IPSEC_IKE_CRL_FAILED	Certificate Revocation Check failed.
13818	0x000035FA	ERROR_IPSEC_IKE_INVALID_KEY_USAGE	Invalid certificate key usage.
13819	0x000035FB	ERROR_IPSEC_IKE_INVALID_CERT_TYPE	Invalid certificate type.
13820	0x000035FC	ERROR_IPSEC_IKE_NO_PRIVATE_KEY	No private key associated with machine certificate.
13822	0x000035FE	ERROR_IPSEC_IKE_DH_FAIL	Failure in Diffie-Helman computation.
13824	0x00003600	ERROR_IPSEC_IKE_INVALID_HEADER	Invalid header.
13825	0x00003601	ERROR_IPSEC_IKE_NO_POLICY	No policy configured.
13826	0x00003602	ERROR_IPSEC_IKE_INVALID_SIGNATURE	Failed to verify signature.
13827	0x00003603	ERROR_IPSEC_IKE_KERBEROS_ERROR	Failed to authenticate using Kerberos.
13828	0x00003604	ERROR_IPSEC_IKE_NO_PUBLIC_KEY	Peer's certificate did not have a public key.
13829	0x00003605	ERROR_IPSEC_IKE_PROCESS_ERR	Error processing error payload.
13830	0x00003606	ERROR_IPSEC_IKE_PROCESS_ERR_SA	Error processing SA payload.
13831	0x00003607	ERROR_IPSEC_IKE_PROCESS_ERR_PROP	Error processing Proposal payload.
13832	0x00003608	ERROR_IPSEC_IKE_PROCESS_ERR_TRANS	Error processing Transform payload.
13833	0x00003609	ERROR_IPSEC_IKE_PROCESS_ERR_KE	Error processing KE payload.
13834	0x0000360A	ERROR_IPSEC_IKE_PROCESS_ERR_ID	Error processing ID payload.
13835	0x0000360B	ERROR_IPSEC_IKE_PROCESS_ERR_CERT	Error processing Cert payload.
13836	0x0000360C	ERROR_IPSEC_IKE_PROCESS_ERR_CERT_REQ	Error processing Certificate Request payload.
13837	0x0000360D	ERROR_IPSEC_IKE_PROCESS_ERR_HASH	Error processing Hash payload.
13838	0x0000360E	ERROR_IPSEC_IKE_PROCESS_ERR_SIG	Error processing Signature payload.
13839	0x0000360F	ERROR_IPSEC_IKE_PROCESS_ERR_NONCE	Error processing Nonce payload.
13840	0x00003610	ERROR_IPSEC_IKE_PROCESS_ERR_NOTIFY	Error processing Notify payload.
13841	0x00003611	ERROR_IPSEC_IKE_PROCESS_ERR_DELETE	Error processing Delete Payload.
13842	0x00003612	ERROR_IPSEC_IKE_PROCESS_ERR_VENDOR	Error processing VendorId payload.
13843	0x00003613	ERROR_IPSEC_IKE_INVALID_PAYLOAD	Invalid payload received.
13844	0x00003614	ERROR_IPSEC_IKE_LOAD_SOFT_SA	Soft SA loaded.
13845	0x00003615	ERROR_IPSEC_IKE_SOFT_SA_TORN_DOWN	Soft SA torn down.
13846	0x00003616	ERROR_IPSEC_IKE_INVALID_COOKIE	Invalid cookie received..
13847	0x00003617	ERROR_IPSEC_IKE_NO_PEER_CERT	Peer failed to send valid machine certificate.
13848	0x00003618	ERROR_IPSEC_IKE_PEER_CRL_FAILED	Certification Revocation check of peer's certificate failed.
13849	0x00003619	ERROR_IPSEC_IKE_POLICY_CHANGE	New policy invalidated SAs formed with old policy.
13850	0x0000361A	ERROR_IPSEC_IKE_NO_MM_POLICY	There is no available Main Mode IKE policy.
13851	0x0000361B	ERROR_IPSEC_IKE_NOTCBPRIV	Failed to enabled TCB privilege.
13852	0x0000361C	ERROR_IPSEC_IKE_SECLOADFAIL	Failed to load SECURITY.DLL.
13853	0x0000361D	ERROR_IPSEC_IKE_FAILSSPINIT	Failed to obtain security function table dispatch address from SSPI.
13854	0x0000361E	ERROR_IPSEC_IKE_FAILQUERYSSP	Failed to query Kerberos package to obtain max token size.

Error			Description
decimal	Hexadecimal	Name	
13855	0x0000361F	ERROR_IPSEC_IKE_SRVACQFAIL	Failed to obtain Kerberos server credentials for ISAKMP/ERROR_IPSEC_IKE service. Kerberos authentication will not function. The most likely reason for this is lack of domain membership. This is normal if your computer is a member of a workgroup.
13856	0x00003620	ERROR_IPSEC_IKE_SRVQUERYCRED	Failed to determine SSPI principal name for ISAKMP/ERROR_IPSEC_IKE service (QueryCredentialsAttributes).
13857	0x00003621	ERROR_IPSEC_IKE_GETSPIFAIL	Failed to obtain new SPI for the inbound SA from Ipsec driver. The most common cause for this is that the driver does not have the correct filter. Check your policy to verify the filters.
13858	0x00003622	ERROR_IPSEC_IKE_INVALID_FILTER	Given filter is invalid.
13859	0x00003623	ERROR_IPSEC_IKE_OUT_OF_MEMORY	Memory allocation failed.
13860	0x00003624	ERROR_IPSEC_IKE_ADD_UPDATE_KEY_FAILED	Failed to add Security Association to IPSec Driver. The most common cause for this is if the IKE negotiation took too long to complete. If the problem persists, reduce the load on the faulting machine.
13861	0x00003625	ERROR_IPSEC_IKE_INVALID_POLICY	Invalid policy.
13862	0x00003626	ERROR_IPSEC_IKE_UNKNOWN_DOI	Invalid DOI.
13863	0x00003627	ERROR_IPSEC_IKE_INVALID_SITUATION	Invalid situation.
13864	0x00003628	ERROR_IPSEC_IKE_DH_FAILURE	Diffie-Hellman failure.
13865	0x00003629	ERROR_IPSEC_IKE_INVALID_GROUP	Invalid Diffie-Hellman group.
13866	0x0000362A	ERROR_IPSEC_IKE_ENCRYPT	Error encrypting payload.
13867	0x0000362B	ERROR_IPSEC_IKE_DECRYPT	Error decrypting payload.
13868	0x0000362C	ERROR_IPSEC_IKE_POLICY_MATCH	Policy match error.
13869	0x0000362D	ERROR_IPSEC_IKE_UNSUPPORTED_ID	Unsupported ID.
13870	0x0000362E	ERROR_IPSEC_IKE_INVALID_HASH	Hash verification failed.
13871	0x0000362F	ERROR_IPSEC_IKE_INVALID_HASH_ALG	Invalid hash algorithm.
13872	0x00003630	ERROR_IPSEC_IKE_INVALID_HASH_SIZE	Invalid hash size.
13873	0x00003631	ERROR_IPSEC_IKE_INVALID_ENCRYPT_ALG	Invalid encryption algorithm.
13874	0x00003632	ERROR_IPSEC_IKE_INVALID_AUTH_ALG	Invalid authentication algorithm.
13875	0x00003633	ERROR_IPSEC_IKE_INVALID_SIG	Invalid certificate signature.
13876	0x00003634	ERROR_IPSEC_IKE_LOAD_FAILED	Load failed.
13877	0x00003635	ERROR_IPSEC_IKE_RPC_DELETE	Deleted via RPC call.
13878	0x00003636	ERROR_IPSEC_IKE_BENIGN_REINIT	Temporary state created to perform reinitialization. This is not a real failure.
13879	0x00003637	ERROR_IPSEC_IKE_INVALID_RESPONDER_LIFETIME_NOTIFY	The lifetime value received in the Responder Lifetime Notify is below the Windows 2000 configured minimum value. Please fix the policy on the peer machine.
13880	0x00003638	ERROR_IPSEC_IKE_QM_LIMIT_REAP	SA reaped because QM limit was reached.
13881	0x00003639	ERROR_IPSEC_IKE_INVALID_CERT_KEYLEN	Key length in certificate is too small for configured security requirements.
13882	0x0000363A	ERROR_IPSEC_IKE_MM_LIMIT	Max number of established MM SAs to peer exceeded.
13883	0x0000363B	ERROR_IPSEC_IKE_NEGOTIATION_DISABLED	IKE received a policy that disables negotiation.
13884	0x0000363C	ERROR_IPSEC_IKE_QM_LIMIT	Reached maximum quick mode limit for the main mode. New main mode will be started.

7.4.2 Troubleshooting/diagnostics

General checks:

- Check the hardware and software requirements described in this documentation (TwinCAT version, CE image version, etc.);
- Check the installation instructions (e.g. installation of the CAB files on a CE system);
- If connection problems occur, the PING command can be used to determine whether the communication partner can be reached via the network connection. If this is not the case, check the network configuration and the firewall settings [▶ 868];

- Sniffer tools such as Wireshark® enable logging of the entire network communication. The log can then be analyzed by Beckhoff support staff;
- Check the configuration of the data points (type, address of the information object, etc.);
- Check whether the other communication partner outputs an error code;
- Activate the debug output [▶ 866] when establishing and terminating the connection and/or the ASDU data. Open the TwinCAT System Manager and activate the LogView window. Check the debug output.
- Check the use of the FB_SocketCloseAll function block and the LISTEN_MODE_CLOSEALL parameter if you communicate via several TCP/IP connections (server/clients) in a runtime system. If you have several connections, only activate one instance of the FB_SocketCloseAll function block once in the initialization step at program start. The LISTEN_MODE_CLOSEALL parameter may no longer be used in this case;

High-level interface:

- Especially at the central station:
  - Check whether the FB_IEC870_5_104Master [▶ 492] function block outputs an error code/error source [▶ 805];
  - Compare/check the compatibility list of the central station [▶ 67] with the compatibility list of the substation;
  - Check whether the network address and port number that you transfer to the FB_IEC870_5_104Master function block are correct;
  - Check the protocol parameters [▶ 651] set on the FB_IEC870_5_104Master function block (iK, iW, t0, t1, t2, t3, APDULength, etc.). Compare the protocol parameters with the parameters in the substation;
  - Check the system parameters [▶ 588] set on the FB_IEC870_5_104Master function block (ASDU address, length of the ASDU address, length of the information object address, length of the cause of transfer COT, etc.). Compare the system parameters with the parameters in the substation;
  - Check the parameters set on the FB_IEC870_5_104Master function block for cyclic data acquisition [▶ 593] (initialization sequence, cyclic general interrogation, cyclic counter interrogation, cyclic test commands, etc.);
- Especially at the substation:
  - Check whether the FB_IEC870_5_104Slave [▶ 495] function block outputs an error code/error source [▶ 805];
  - Check the compatibility list of the substation [▶ 85] with the compatibility list of the central station;
  - Check whether the input parameters that you transfer to the F_CreateServerHnd function are correct (network address, port number, etc.);
  - Check the protocol parameters [▶ 651] set on the FB_IEC870_5_104Slave function block (iK, iW, t0, t1, t2, t3, APDULength, etc.). Compare the protocol parameters with the parameters in the control station;
  - Check the system parameters [▶ 588] set on the FB_IEC870_5_104Slave function block (ASDU address, length of the ASDU address, length of the information object address, length of the cause of transfer COT, etc.). Compare the system parameters with the parameters in the control station;

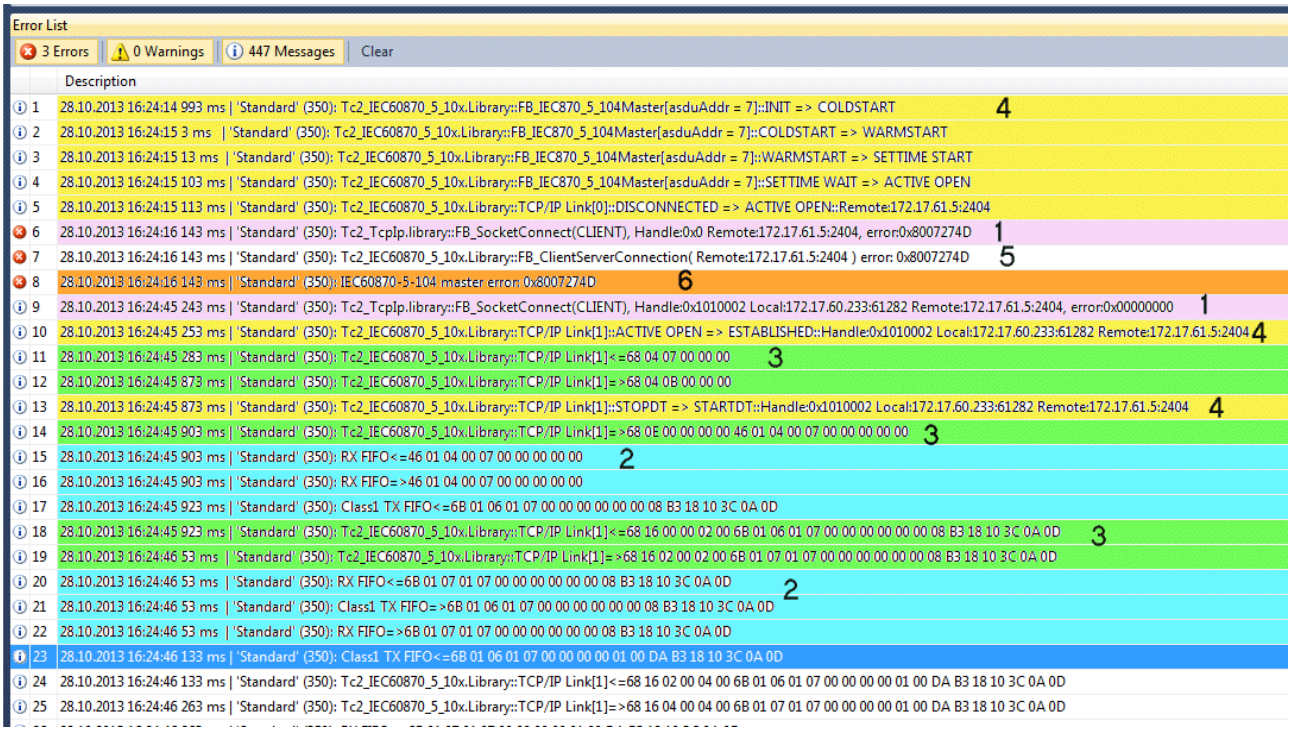
Low-level interface:

- Check whether the FB_IEC870_5_104TProtocol [▶ 514] function block outputs an error code [▶ 805];
- Compare/check the compatibility list of the central station with the compatibility list of the substation;
- Check whether the network address and port number that you transfer to the F_CreateServerHnd function or to the FB_ClientServerConnection or FB_ServerClientConnection function block are correct;
- Check the protocol parameters [▶ 651] set on the FB_IEC870_5_104TProtocol function block (iK, iW, t0, t1, t2, t3, APDULength, etc.). Compare the protocol parameters with the parameters of the communication partner;

- Check the address lengths configured on the TX/RX data buffer (instance of ST_IEC870_5_101TBuffer [▶ 510]): length of the ASDU address, length of the information object address, length of the cause of transfer, max. ASDU length. Compare the address lengths with the configuration of the communication partner;

7.4.3 Debug messages in TwinCAT->Error List

Debug messages that are written in TwinCAT XAE->Error List facilitate troubleshooting in the system. To view the activated debug messages, start TwinCAT XAE and select in the menu: View->Other Windows->Error List. A debugging output is shown below. The different types of messages were marked with numbers.



The messages are activated in different ways (by different parameters) depending on whether you are using the "low-level" or "high-level" interface. Below is a list of the different parameter names.

1. Debug messages logged when establishing or terminating the TCP/IP connection:

Name	ON	OFF	Substation		Central station	
			Low-Level	High-Level	Low-Level	High-Level
F_CreateServerHnd	CONNECT_MODE_ENABLEDBG	0	Yes	Yes	-	-
FB_ClientServerConnection	CONNECT_MODE_ENABLEDBG	0	-	-	Yes	-
FB_IEC870_5_104Master [▶ 492].sysPara.dbgMode	IEC870_DEBUGMODE_LINKERROR	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

2. Hexadecimal output of ASDUs (without link layer control header). 32 ASDU data bytes per row are output as hexadecimal numbers. Longer ASDUs are spread across several rows:

Name	ON	OFF	Substation		Central station	
			Low-Level	High-Level	Low-Level	High-Level
ST_IEC870_5_101TBuffer [▶ 583].eDbg	eIEC870_FIFO_DBG_ALL	eIEC870_FIFO_DBG_OFF	Yes	-	Yes	-
FB_IEC870_5_104Slave [▶ 495].sysPara.dbgMode	IEC870_DEBUGMODE_ASDU	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_104Master [▶ 492].sysPara.dbgMode	IEC870_DEBUGMODE_ASDU	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

3. Hexadecimal output of the APDUs (TCP/IP telegrams). 32 APDU data bytes per row are output as hexadecimal numbers. Longer APDUs are spread across several rows:

Name	ON	OFF	Substation		Central station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_104TProtocol [▶ 514].bOutDbg	TRUE	FALSE	Yes	-	Yes	-
FB_IEC870_5_104Slave [▶ 495].sysPara.dbgMode	IEC870_DEBUGMODE_LINKLAYER	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_104Master [▶ 492].sysPara.dbgMode	IEC870_DEBUGMODE_LINKLAYER	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes-

4. Station status messages:

Name	ON	OFF	Substation		Central station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_104Slave [▶ 495].sysPara.dbgMode	IEC870_DEBUGMODE_DEVSTATE	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_104Master [▶ 492].sysPara.dbgMode	IEC870_DEBUGMODE_DEVSTATE	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

5: Output of link layer errors as messages:

Name	ON	OFF	Substation		Central station	
			Low-Level	High-Level	Low-Level	High-Level
FB_IEC870_5_104Slave [▶ 495].sysPara.dbgMode	IEC870_DEBUGMODE_LINKERROR	IEC870_DEBUGMODE_DISABLED	-	Yes	-	-
FB_IEC870_5_104Master [▶ 492].sysPara.dbgMode	IEC870_DEBUGMODE_LINKERROR	IEC870_DEBUGMODE_DISABLED	-	-	-	Yes

6. Messages logged by the PLC application (sample project) using the ADSLOGSTR, ... functions.

Further diagnostic tools:

- TwinCAT ADS Monitor;
- Network monitor;
- Wireshark®;
- Ethereal;
- Diverse protocol test suite products;

7.4.4 Firewall settings

IEC 60870-5-104 uses the TCP/IP protocol for communication. It is therefore essential to make sure that the corresponding TCP port will be activated during the using of a firewall. The table below lists standard ports, that have to be considered by using a firewall.

D escription	Type	Protocol	Port
IEC 60870-5-104 telecontrol protocol	Protocol	TCP	2404

The configuration of the Windows Firewall is done via the corresponding dialog in the control panel. For further information please see the Windows resp. Firewall documentation.

7.5 Glossary

Term	Description
substation, slave, server, controlled station	Synonyms for a subordinate station (which is monitored)
central station, master, client, controlling station	Synonyms for a higher-level station (control station, monitors others stations)
control direction	Data transfer direction from the central station to the substation
monitoring direction	Data transfer direction from the substation to the central station
application objects	IEC information objects in the TwinCAT PLC application (single points, double points, measured values, short floating point values etc.)
APDU	Application protocol data unit
APCI	Application protocol control information
ASDU	Application service data unit
IOA, information object address	Address of the single point, double point, measured value, short floating point value etc.
primary station	Primary station sends commands/requests to the secondary station and controls the data transmission of the secondary station.
secondary station	Secondary station anwers to the requests of the primary station.
combined station	Combined stations are able to act as primary and secondary station (balanced mode).
balanced mode	In balanced mode both stations act as primary and secondary stations and are able to initialize the data transmission.
unbalanced mode	In unbalanced mode only the primary station is able to initialize the data transmissions. The central station acts as primary station and the substation acts as secondary station.
CAA	Common ASDU address
COT	Cause of transfer

7.6 FAQ – frequently asked questions and answers

In this section frequently asked questions are answered, in order to facilitate your work with the TwinCAT telecontrol.

If you have any further questions, contact please our Support (- 157)

How many data points can be communicated over TCP with the IEC 60870-5-104 protocol? [▶ 869]

? How many data points can be communicated over TCP with the IEC 60870-5-104 protocol?
! This question cannot be answered in general terms. Beckhoff integrates all IEC 60870-5 communication protocols in the PC-based control technology. Therefore, the number of possible data points depends on the system resources and the set cycle time. For the 104 protocol, an additional factor is the quality of the network. In the serial protocols, another important factor is the baud rate.

Trademark statements

Beckhoff®, ATRO® , EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar®, and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

Arm, Arm9 and Cortex are trademarks or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere.

Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

Wireshark is a registered trademark of Sysdig, Inc.

More Information:
www.beckhoff.com/tf6500

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