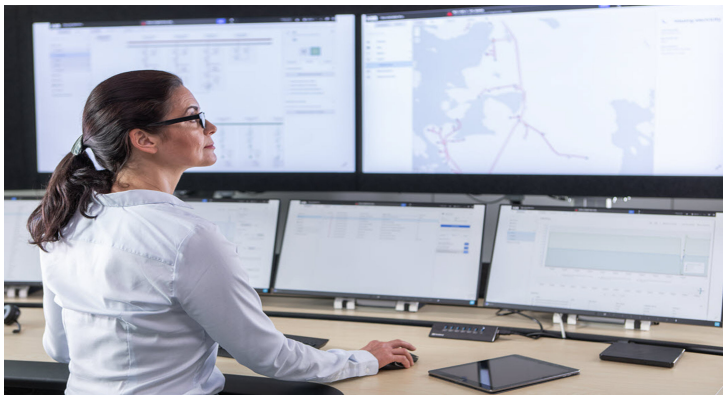
GRID AUTOMATION PRODUCTS

MicroSCADA X SYS600 10.2

IEC 60870-5-101 Slave Protocol



Document ID: 1MRK 511 490-UEN
Issued: March 2021
Revision: A
Product version: 10.2

Table of contents

Section 1 Copyrights

The information in this document is subject to change without notice and should not be construed as a commitment by Hitachi Power Grids. Hitachi Power Grids assumes no responsibility for any errors that may appear in this document.

In no event shall Hitachi Power Grids be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall Hitachi Power Grids be liable for incidental or consequential damages arising from the use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from Hitachi Power Grids, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Trademarks

ABB is a registered trademark of ABB Asea Brown Boveri Ltd. Manufactured by/for a Hitachi Power Grids company. All other brand or product names mentioned in this document may be trademarks or registered trademarks of their respective holders.

Guarantee

Please inquire about the terms of guarantee from your nearest Hitachi Power Grids representative.

Third Party Copyright Notices

List of Third Party Copyright notices are documented in "3rd party licenses.txt" and other locations mentioned in the file in SYS600 and DMS600 installation packages.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (https://www.openssl.org/). This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Section 2 Introduction

2.1 This manual

This manual provides thorough information on the use of IEC 60870-5-101 Slave Protocol and information related to it. It describes how to configure the base system and the communication system to establish communication to IEC 60870-5-101 master.

In addition to this configuration, the base system needs to be configured for other communication tasks, for example, process communication. For information on this subject, see other manuals, for example, SYS600 Application Objects and SYS600 System Objects.

2.1.1 IEC 60870-5-101 Slave Protocol

The IEC 60870-5-101 Slave protocol is mainly used for upper level communication between SYS600 and a Substation Control System (SCS) as illustrated by Figure 1.

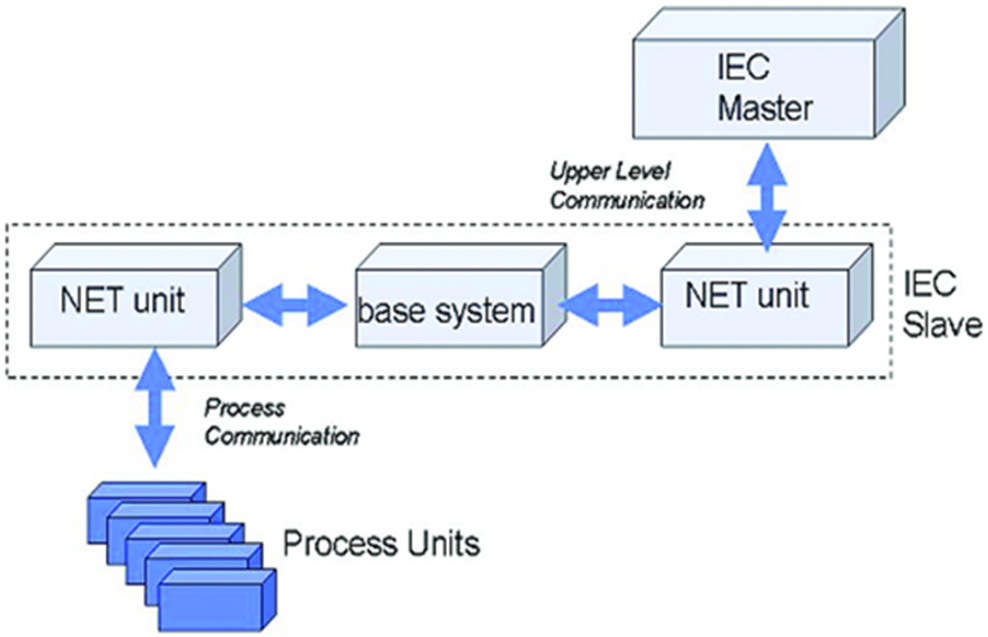


Figure 1: The IEC master sees the NET unit and the process behind it as a slave

The data from the process activates a certain event channel and a command procedure in the base system. This command procedure forwards the information to the NET unit and the IEC master.

2.2 Use of symbols

This publication includes warning, caution and information symbols where appropriate to point out safety-related or other important information. It also includes tips to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

Warning icon indicates the presence of a hazard which could result in personal injury.

Caution icon indicates important information or a warning related to the concept discussed in the text. It might indicate the presence of a hazard, which could result in corruption of software or damage to equipment/property.

Information icon alerts the reader to relevant factors and conditions.

Tip icon indicates advice on, for example, how to design a project or how to use a certain function.

Although warning hazards are related to personal injury, and caution hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all warnings and caution notices.

2.3 Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a dialog, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for file names.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the CTRL key. Although the Enter and Shift keys are not labeled, they are written in capital letters, for example, press ENTER.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the space bar, comma key and so on.
- Press CTRL+C indicates that the user must hold down the CTRL key while pressing the C key (in this case, to copy a selected object).
- Press ALT E C indicates that the user presses and releases each key in sequence (in this case, to copy a selected object).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
    - The following convention is used for menu operations: **Menu Name/Menu Item/ Cascaded Menu Item**. For example: select **File/Open/New Project**.
    - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.
- System prompts/messages and user responses/input are shown in the Courier font. For example, if the user enters a value that is out of range, the following message is displayed: `Entered value is not valid`.
  The user may be told to enter the string MIF349 in a field. The string is shown as follows in the procedure: `MIF349`
- Variables are shown using lowercase letters: sequence name

2.4 Related documents

The following SYS600 manuals should be available for reference during the use of this manual:

Name of the manual	Document ID
SYS600 10.2 Communication Gateway, COM500*i*	1MRK 511 468-UEN
SYS600 10.2 System Configuration	1MRK 511 481-UEN
SYS600 10.2 System Objects	1MRK 511 482-UEN
SYS600 10.2 Application Objects	1MRK 511 467-UEN

2.4.1 Other referenced manuals

The IEC 60870-5-101 protocol is based on the following documents by the IEC Technical Committee 57:

IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Data Link Transmission Services
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Information Elements
IEC 60870-5-5	Basic Application Functions

2.5 Document revisions

Revision	Version number	Date	History
A	10.2	31.03.2021	New document for SYS600 10.2

Section 3 Safety information

This section has information on the prevention of hazards and taking backups from the system.

3.1 Backup copies

3.1.1 Taking backup copies

We recommend taking backup copies before making any changes, especially ones that might have side effects. Software and data need to be copied to another place.

Backup copying makes it easier to restore the application software in case of disk crash or other severe failure where stored data is lost. It is therefore recommended that backup copies are taken regularly.

There should be at least two system backup copies and two application copies. A new backup is copied over the oldest backup. This way the latest version is always available, even if the backup procedure fails.

Detailed information on how to take backup copies should be delivered to the customer with the application.

3.1.2 System backup

Usually a system back up is taken after the application is made. It should be taken again when changes are made to the SYS600 system. This is required when the driver configuration or the network setup is changed.

3.1.3 Application backup

An application backup is also taken at the same time with the system backup, after the application is made. It should be taken again when changes are made to the application, for example, if pictures or databases are edited or new pictures are added.

3.2 Fatal errors

A fatal error is an error that causes a breakdown or a locked situation in the SYS600 program execution.

3.2.1 Handling

In case of a fatal error:

1. Write down the possible SYS600 error messages.
2. Shut down the SYS600 main program. If this cannot be done in the SYS600 Control Panel, try to end the task in Windows Task Manager.

Files may be damaged if the base system computers are shut down by switching the power off.

3. The data kept in the main memory at the moment of a fatal error is placed in the drwtsn32.log file with Windows 2003 Server, Windows XP and earlier. By default, it is placed under %SYSTEMDRIVE%\Documents And Settings\All Users\Application Data \Microsoft\Dr Watson. Log and dump file paths can be checked with the drwtsn32 application. (Start -> run -> drwtsn32.exe). Analyze and copy the data in these files. Starting with Windows Server 2008 and Windows 7 the crash handling has changed. The location of the dump files can be read from the registry under the key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting \LocalDumps. The DumpFolder value tells the location of the dump files. Collect the data from this location.
4. Restart the system.

Report the program break-down together with the possible SYS600 error messages and the information from the drwtsn32.log file to the SYS600 supplier.

3.2.2 Status codes

Error messages in SCIL are called status codes. A list of status codes and short explanations for them can be found in SYS600 Status Codes.

Section 4 Instructions

4.1 Communication

In SYS600 the IEC 60870-5-101 Slave protocol is implemented only in the PC-NET software. PC-NET unit communicates over an INTEGRATED link and via the serial or LAN ports of the base system computer.

Setting the attributes of SYS600 system objects can modify the communication parameters.

The base system sees each IEC device as a station (STA object) that has been created to a line of a NET unit. Each IEC station works as a protocol converter that converts data between the internal protocol of SYS600 and the IEC 60870-5-101 protocol.

The protocol can be used with virtual serial ports by using a special setting of the line attributes. See NET line attribute CM for more information.

4.2 Installation

The SYS600 installation is required.

4.3 Configuration

Configuration can be made either by using the System Configuration Tool or by using SCIL statements. For more information on the System Configuration Tool, see SYS600 System Configuration manual, chapter "PC-NET start-up with System Configuration Tool" and "System Configuration Tool". The usage of the System Configuration Tool is recommended, but if there is a need to create the communication configuration using SCIL, it is instructed in the following chapters. In this case, the configuration can be divided into two parts:

The configuration can be divided into two parts:

- Base system configuration
- Communication system configuration

The attribute descriptions presented in chapter 'Communication system configuration' are the same for configurations created with System Configuration Tool or with SCIL.

4.3.1 Base system configuration

It is assumed here that the base system configuration for objects other than the communication has been made according to the instructions in the System Configuration manual.

The extra steps needed to configure the communication are:

1. Define a node number for a PC_NET instance.
2. Reserve a link number for a PC_NET instance. Creating the link as instructed in step 6 starts the PC_NET instance.
3. Create the base system STA object for each remote IED (master function) or for each NCC connection (slave function).

- IEC 60870-5-101 Slave protocol uses the station type IEC (STY type 29)

The STA objects are created to SYS_BASCON.COM using the template or with a separate creation sequence. If the template is not used, the sequence should contain the line:

```
#create STA'Sta_Nb':B = %Sta
```

where 'Sta_Nb' is the number of the station object in the base system. %Sta is a list object which should contain at least the following settings: TT = "EXTERNAL", ST = station type, ND = node number defined in step 1 and TN = translated object number (usually the same as 'Sta_Nb'. See SYS600 System Objects manual for more information on the base system object attributes for STA object).

4. Edit the PC_NET.CF1 according to the description in chapter "Start-up definition file PC_NET.CF1" in the SYS600 System Configuration manual
5. Create a command procedure which creates the lines and stations to the NET object (= pc_net instance) using the S-attributes.
   See Section 4.3.2 for more information on the attribute setting. A sample creation script is presented at the end of this manual.
6. Create a command procedure which creates the link of type 'INTEGRATED' to the base system. This procedure should contain the line:

   ```
   #set LIN'i_Integrated_Link_Number':BLT = "INTEGRATED"
   ```

   where 'i_Integrated_Link_Number' is the number of the link reserved in step 2. The PC_NET executable is defined with the SC attribute of the link and it must set before setting of the LT attribute.

The testing of the communication system can be done as follows:

1. Execute the procedure created in step 6. This starts the PC_NET instance and enable the setting of the S-attributes.
2. Execute the procedure created in step 5. If the lines and stations are set to IU = 1 (that is, they are in use) and the configuration is correct and complete in both ends, the communication starts.

For automatic start-up of the communication, the created command procedures must be attached to the APL_INIT_1:C procedure.

4.3.2 Communication system configuration

Each NET instance contains a set of system objects which specify the existence and the usage of the communication lines and the station objects connected to those lines. These objects can be created, modified, and deleted by SCIL, and setting the attributes defines the functionality of these objects.

Access to the attributes can be one of the following:

- **Read-only**: The attribute can only be read. There are still a few exceptions in which the values can be reset.
- **Write-only**: The attribute can only be written (set).
- **Read, conditional write**: The attribute can be both read and written, but the object must be set out of use (IU = 0) before writing.
- **No limitations**: The attribute can be both read and written without limitations.

The configuration of the communication system in SYS600 can be divided into two layers: line layer and station layer. Both of these layers have a specific functionality and a set of attributes of their own.

The purpose of the communication system configuration is to:

- Create all the system objects needed to establish communication between the master and the slave. Related attributes for creation are PO (Line) and DV (Station).
- Adjust the values of the system object attributes to match the physical communication channel and the properties of the remote partner/partners. The menu selection 'Configuration->Preview->PC_NET' in the System Configuration Tool may provide an example of the SCIL based configuration script of any setup.

4.3.2.1 Setting the attribute values

All the line and station attributes have sensible default values, but the value of each attribute must be checked against the requirements of the actual communication system.

The attribute values depend on:

- The physical communication media (for example leased telephone line, radio link, power line carrier), which affects the attributes of the line, such as the baud rate and parity.
- The network topology used (point-to-point, multi-drop), which affects the link type.
- The size (number of stations) of the system, which affects the timeout parameters; the slower the media and larger the system, the longer timeouts are needed.
- The remote system(s), which affects both the line and station attributes, and the message types used.

4.3.2.2 Network topologies

The implementation of the IEC 60870-5-101 Slave protocol in SYS600 supports direct and serial bus topologies. The direct topology (point-to-point) can be a direct physical cable from point-to-point, a two-node radio, or modem network. The serial bus topology (multi-drop) is commonly made up of many modems with their outputs/inputs tied together or connected using a star-coupler.

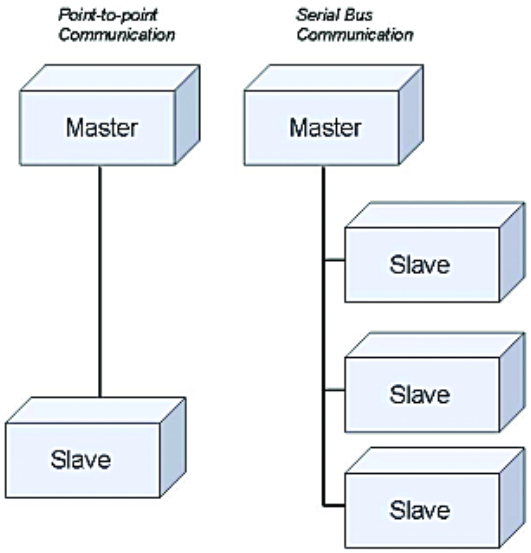


Figure 2: Network topologies

Redundant line functionality is described in Section 5.4.5

When the serial bus (multi-drop) network topology is used, only the unbalanced mode of the IEC 60870-5-101 protocol can be used.

When making the IEC connection, an agreement about the used communication parameters should be made with the supplier or owner of the master system.

4.3.2.3 IEC 60870-5-101 line layer

The line process of a NET unit performs the functions of the line layer. The purpose of the line layer is to send and receive messages to/from external devices using the IEC 60870-5-101 protocol. The line layer provides frame synchronization and link control.

According to the IEC 60870 standards, the line layer performs the following functions:

- Provides access to the transmission medium.
- Serializes and deserializes frames.
- Adds and removes frame delimiters, if this is not performed by data circuit terminating equipment.
- Detects frame synchronization errors.
- Detects frame size errors.
- Monitors signal distortion, if this is not performed by data circuit terminating equipment.
- Recognizes frames addressed to a designated station.
- Prevents the station transmitting without pause.
- Protects messages against loss and errors within predetermined limits.
- Reports on persistent transmission errors.
- Reports on the status of link configuration.
- Supports initiation and maintenance functions.

Line layer attributes
The following attributes can be used for configuring IEC 60870-5-101 master lines in SYS600.

IU	In Use
Indicates whether the line is in use (value 1) or not in use (value 0).	
Data type:	Integer
Value:	0 or 1
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	No limitations

PO	Protocol
The data transfer protocol used on the line. The line is defined to the NET by setting this attribute. By setting the attribute to 0 the line definition including all the line attributes are deleted.	
Data type:	Integer
Value:	0...45 Value with IEC 60870-5-101 Slave protocol: 29 (unbalanced mode) or 30 (balanced mode)
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

SD **System Device Name**

Associates the NET line numbers of PC-NET with the device names of the physical channels of serial ports. By default, the line number 1 is connected to the COM1, the line 2 to the COM2 and so on. By using the SD attribute, it is possible to override these default values. This may be necessary if the COM ports are used as NET lines or if, for example, a RocketPort card is used.

Data type:	Text
Value:	See above
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

PS **Buffer Pool Size**

Specifies the number of message buffers reserved for the line. Fixed buffer poll sizes are used in versions 9.3 FP1 and newer and this attribute is retained because of the backward compatibility. Setting the value for PS is not possible anymore. See the attributes PS, NB and PB from the System Objects manual for more information.

Data type:	Integer
Value:	1...250
Index range:	1...12 (NET line numbering)
Access:	Read (conditional write accepted but has no effect)

BR **Baud Rate**

Transmission rate used on the line.

Data type:	Integer
Value:	1...19200 (384 = 38400 bauds, 576 = 57600 bauds)
Unit:	Bits / s
Index range:	1...12 (NET line numbering)
Default value:	9600
Access:	Read, conditional write

PY **Parity**

Specifies the parity check (if any) used for the characters transferred on the line.

Data type:	Integer
Value:	0 = no parity check
	1 = odd parity
	2 = even parity
Index range:	1...12 (NET line numbering)
Default value:	2
Access:	Read, conditional write

RD **Receiver Data Bit Count**

Specifies the number of data bits in each received character.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Data bits
Index range:	1...12 (NET line numbering)
Default value:	8
Access:	Read, conditional write

SB	Stop Bits

Specifies the number of stop bits attached to each transmitted character.

Data type:	Integer
Value:	1 or 2
Unit:	Stop bits
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

TD	Transmitter Data Bit Count

Specifies the number of data bits in each transmitted character.

Data type:	Integer
Value:	5, 6, 7 or 8
Unit:	Data bits
Index range:	1...12 (NET line numbering)
Default value:	8
Access:	Read, conditional write

PD	Polling Delay

Delay between polling messages. The purpose of this attribute depends on the communication mode.

In the unbalanced mode this attribute is used only to detect if the master is polling the slave. This watchdog is not used when PD is set to 0. If the PD value is larger than 0 and no polls are received between two expirations of the PD timer, the status of the line is reported to be "not OK". In practice, this means that the time from line disconnection to reporting is 2 x PD. The status of the line and the stations are also reported to be "not OK", if the PD value is larger than 0 in the unbalanced mode and the count of the subsequent Request status of link message reaches the value defined with the line attribute EN. This function can be used to reveal a situation in which the communication line operates only in one direction.

In the balanced mode, the link layer of the protocol checks the state of the communication, if the time between two consecutive messages is more than the value of the PD attribute.

Data type:	Integer
Value:	0... 65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	30000 (unbalanced mode) or 5000 (balanced mode)
Access:	Read, conditional write

PL	Polling Limit

This attribute is used for controlling the polling sequence of IEC stations. The purpose of the PL attribute is to limit the number of successive polls of one station (link address). Normally one station is polled until all the data is read. This attribute is used only in the unbalanced mode.

Data type:	Integer
Value:	2...100
Index range:	1...12 (NET line numbering)
Default value:	10
Access:	Read, conditional write

SL	Secondary polling Limit

This attribute is used to control the class 2 polling of IEC stations. The purpose of the SL attribute is to limit the number of successive class 2 polls of one station (link address). If a value of zero is given, the attribute is meaningless and the value of PL applies to class 2 polls. The value of SL is limited to be less or equal to PL. This attribute is used only with the unbalanced mode.

Value:	0 .. 100 (less or equal to PL)
Indexing:	1..12 (NET line numbering)
Access:	Read, conditional write
Default:	1

TW	Transmission Wait Delay

Specifies the transmission delay in milliseconds. The transmission delay is the time that the NET must wait after receiving a CTS (Clear to Send) signal until starting the transmission of a message.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	Read, conditional write

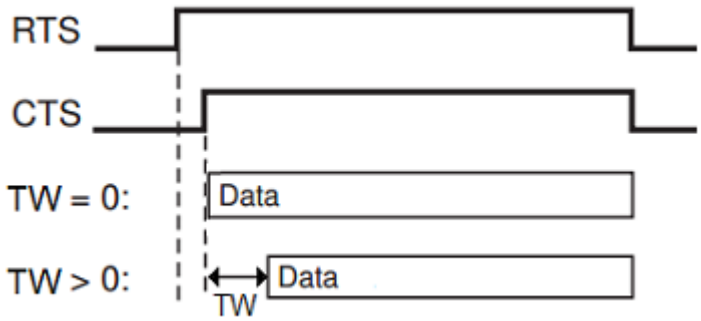


Figure 3: TW attribute

DE	CTS Delay

The maximum waiting time of the rising edge of the CTS signal after the activation of the RTS signal. If CTS is low after the timeout, it causes a CTS error and the transmission does not start. A line disconnection often leads to a CTS error. If value = 0 is given, 500 msec waiting time is applied. If simulated CTS high is used (serial only, line attribute CM, bit 1), there is no rising edge and transmission starts after the waiting time.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	50
Access:	Read, conditional write

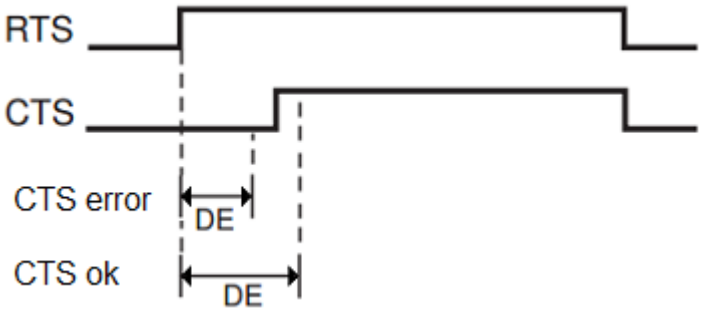


Figure 4: DE attribute

In the Figure 4 the smaller DE value causes a CTS error. A suitable DE value varies with each system.

HT	Header Timeout

Specifies the maximum waiting time in milliseconds within which the first byte of a line layer response from the IEC slave should have been received after the transmission of a message. If no response has been received within this time, new attempts are performed the number of times specified by the Enquiry limit. If a response is still not obtained, the station is suspended. The attribute is meaningful only in balanced mode.

Data type:	Integer
Value:	0...65535
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	2000
Access:	Read, conditional write

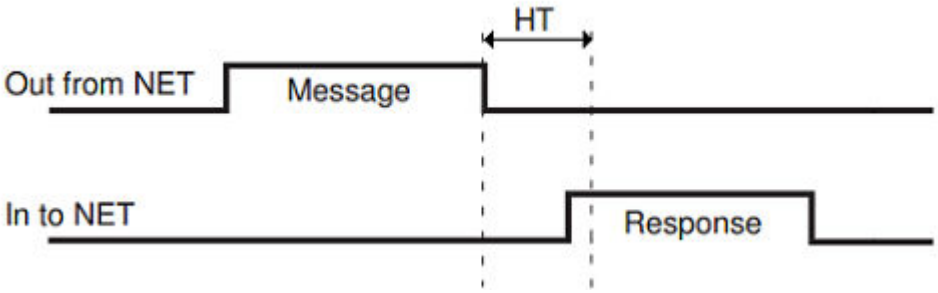


Figure 5: HT attribute

TI	Response Timeout

The time in seconds that the IEC link waits for the end of the line layer response.

Data type:	Integer
Value:	0...255
Unit:	Seconds
Index range:	1...12 (NET line numbering)
Default value:	2
Access:	No limitations

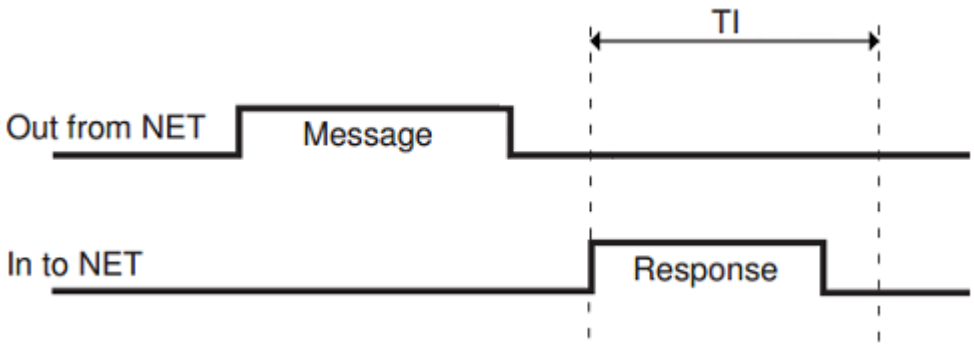


Figure 6: TI attribute

A slow communication speed and a long response might cause error with too small TI attribute value.

RY	RTS Keepup Delay

This attribute defines how long time the RTS-pin of the RS232-port is kept in the signal state after the serial driver completes the write operation. The write operation here means a transmission of any message. See also line attribute CM (Com Port Mode), bit 3.

Data type:	Integer
Value:	0...20
Unit:	Bytes (absolute time depends on baud rate)
Index range:	1...12 (NET line numbering)
Default value:	1
Access:	Read, conditional write

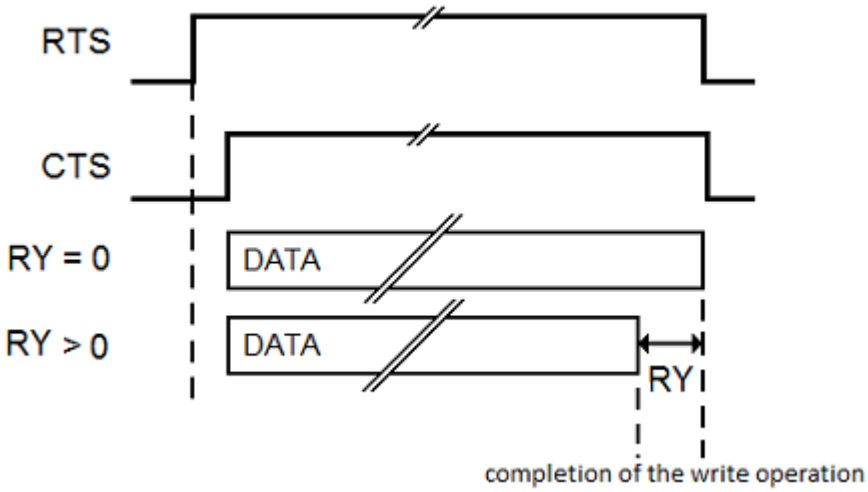


Figure 7: RY attribute

RK	RTS Keep Up Padding Characters

The number of padding characters (null characters) inserted to the end of a telegram to delay the passivation of the RTS (Request To Send) signal.

Data type:	Integer
Value:	0...255

Table continues on next page

RK	RTS Keep Up Padding Characters
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	Read, conditional write

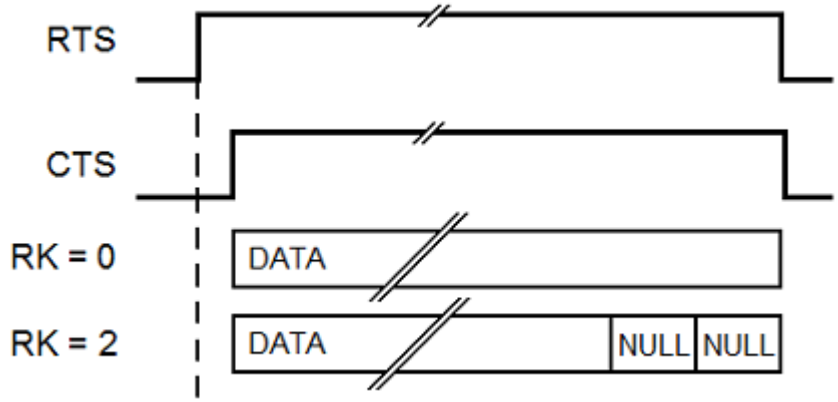


Figure 8: RK attribute

Instead of using RK attribute, the RTS Keepup Delay (RY) attribute can be used.

RI	Receive Interrupt Enable Delay

Defines the delay in milliseconds after which the receiver of a NET line is enabled after a message has been issued. If the given value is too big, the first characters of the response are not necessarily received. Attribute is meaningful only in balanced mode if line attribute LK is not 12 or 13.

Data type:	Integer
Value:	0...255 0 = receiver is always enabled 1...9 = receiver enabled right after transmission 10... = receiver enabled as stated by the value
Unit:	Milliseconds
Index range:	1...12 (NET line numbering)
Default value:	0 (balanced mode) or 5 (unbalanced mode)
Access:	No limitations Read, conditional write

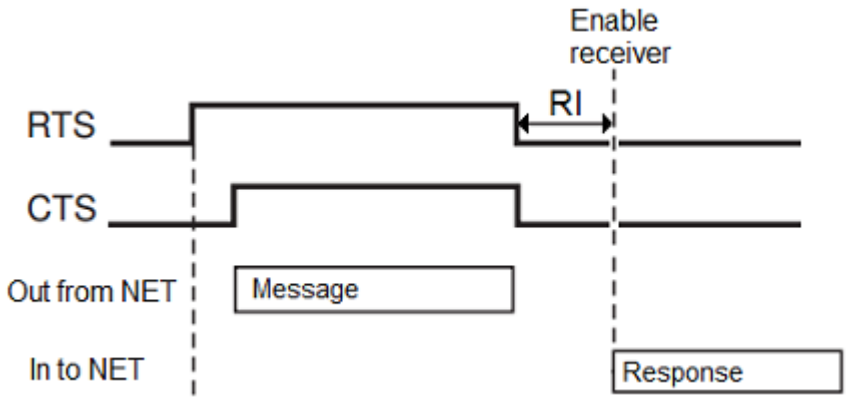


Figure 9: RI attribute

EN	Enquiry Limit

Specifies the maximum number of times that a message is retransmitted after a timeout. In the unbalanced mode, this attribute specifies the maximum amount of subsequent Request status of link messages before the line is reported to be "not OK". This function is active only if the value of the line attribute PD is bigger than 0, also see the corresponding attribute description.

Data type:	Integer
Value:	1...255
Index range:	1...12 (NET line numbering)
Default value:	3
Access:	Read, conditional write

SG	Modem Signal

An attribute for direct supervision and control of the state of the modem signal. This attribute applies to all protocols. It is used for diagnostics and testing.
If the incoming signal DCD or CTS is wanted to have a simulated high value all the time, value = 1 can be written to these signals. This feature may be necessary for easier cabling or with virtual serial ports. If value = 0 is written to these signals, the actual state of signal is used. The default mode of operation is the actual state. See also attribute CM Com Port Mode.

Data type:	Integer
Value:	0 = Passive signal 1 = active signal
Incoming:	DCD and CTS signals
Outgoing:	DTR signal
Index range:	100 * line no + signal no. Signal no. 5 = CTS, 8 = DCD, 20 = DTR
Access:	Read-only, write possible to signals 5 = CTS and 8 = DCD

Examples:

```
#SET NET1:SSG208 = 1 ; line 2 of NET1 should behave as DCD is 'high'
all the time
#SET NET1:SSG205 = 1 ; line 2 of NET1 should behave as CTS is 'high'
all the time
#SET NET1:SSG208 = 0 ; line 2 of NET1 should use the actual state of
the DCD
#SET NET1:SSG205 = 0 ; line 2 of NET1 should use the actual state of
the CTS
```

CM	COM Port Mode

This attribute consists of a set of flags which control the behavior and functionality of the serial port of the line. Each flag is one bit of this attribute.

Data type:	Integer
Value:	0 ... 15 (see below)
Default value	0
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write
Bit 0:	UART error handling When this bit is 0, the UART errors are read before the bytes are read from the serial port. This is the default mode. When the bit is 1, the UART errors are read as a separate operation after the bytes are read from the serial port. This mode is similar to PC_NETs older than 9.2SP2 and it does not detect all errors detected by the serial port hardware. If the line has a lot of disturbances, this mode may result in better performance than the default mode.

Table continues on next page

CM	COM Port Mode
Bit 1:	Simulated high of the CTS signal When this bit is 0, the actual state of the CTS signal is used in the protocol. This is the default mode. When this bit is 1, the CTS signal is simulated to be in the high state all the time and the line behaves according to that. This setting may be necessary with a virtual serial port or for easier cabling. See line attribute DE how transmission starts when CTS is constantly 'high'.
Bit 2:	Simulated high of the DCD signal When this bit is 0, the actual state of the DCD signal is used in the protocol. This is the default mode. When this bit is 1, the DCD signal is simulated to be in the high state all the time and the line behaves according to that. This setting may be necessary with a virtual serial port or for easier cabling.
Bit 3:	Calculated RTS Keep up time When this bit is 0, the keep up time of the RTS signal is not calculated using the length of the message but it is assumed that the driver of the serial port blocks the execution of the sending process until the message is actually sent. This setting should be used if the setting 'Wait on physical transmission before completing write' or similar is selected in the driver for the port in question. The tuning of the RTS keep up time should be done with line attribute RY. This is the default setting. When this bit is 1, the keep up time of the RTS signal is calculated using the length of the sent message and the baud rate of the port. The RTS keep up time defined with the line attribute RY is added to the calculated time. This setting is not needed if the setting 'Wait on physical transmission before completing write' or similar is selected in the driver for the port in question. This setting is the most common with the virtual serial ports, too. If the serial driver does not provide setting 'Wait on physical transmission before completing write' or similar and RTS signal is actively used by the modem hardware, it is worth to test both alternatives. For accurate analysis using protocol analyzer function, see also the description of the bit 2 of the line attribute AU Analyzer Usage.

Having a simulated value in CTS or DCD may have an effect on how a RS-232 line disconnection is detected and reported to the MicroSCADA application.

MI	Message Identification
Object address of system messages.	
Data type:	Integer
Value:	1...32760
Index range:	1...12 (NET line numbering)
Default value:	6000 + (100 * NET number) + line number
Access:	Read, conditional write

MS	Message Application
The number of the application that is the receiver of the system messages generated by the line.	
Data type:	Integer
Value:	1...250
Default value:	1
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

LK **Link Type**

The type of data link connection used on the line. This attribute controls the behavior of the RTS-control line. The states of the CTS and DCD signals of the serial port can have simulated values. The usage of this feature may be necessary, if the line uses a virtual serial port or the hardware connected to the serial port requires a special cable. See the description of the line attribute CM for more information.

Data type:	Integer
Value:	4: Radio link 12: Ready to Send (RTS) signal always set, balanced mode 13: RTS/CTS controlling, balanced mode
Index range:	1...12 (NET line numbering)
Default value:	4 (unbalanced mode), 12 (balanced mode)
Access:	Read, conditional write

With values 0..10, the behavior is similar to 4: Radio Link, in which RTS/CTS controlling is used and the messages are received in unbalanced fashion. This value should be used in the unbalanced mode. In the balanced mode, the value must be 12 or 13.

CB **Carrier Blocking**

This attribute determines whether the incoming Carrier Detect (DCD) signal of the serial port must be set in order for the IEC line to receive messages. If Carrier Detect is expected (CB=1), the incoming bytes of the messages are discarded while the DCD signal is not set. Historically, the Carrier Detect was used with modems (for example, V.23) in order to indicate the direction of the data flow and filter out corrupted characters when the data flow direction changes. A hardware solution is to connect the DCD pin to the DTR pin in the RS-232 cable. In this case the messages are always received when the line is in use (IU=1). In versions 9.3FP2 and newer, the line attribute CM, bit 2 can be used to achieve the same effect. The usage of the line attribute CM is recommended, since it is supported by all serial protocols.

Data type:	Integer
Value:	0 = Carrier blocking not used, messages are received regardless of the DCD state 1 = Carrier blocking used, DCD must be signaled in order to receive messages
Default value:	1
Index range:	1...12 (NET line numbering)
Access:	Read, conditional write

DC **Diagnostic Counters**

The line protocols gather statistical information about the events on the lines by incrementing a number of diagnostic counters. All the major events and error situations of the communication have their own counters. When accessing diagnostic counters, the attribute is indexed according to the formula:
100 * (line number) + (diagnostic counter number)
The IEC 60870-5-101 Slave protocol supports the following counters:

1. Transmitted telegrams
2. Failed transmissions
4. Transmitted commands
5. Transmitted replies
11. Received messages
12. Parity errors
13. Overrun errors
14. Check sum errors
15. Framing errors
16. Buffer overflow errors

Data type:	Integer

Table continues on next page

DC	Diagnostic Counters
Value:	0...30000
Index range:	See above
Access:	Read-only, the values can be reset

OM	Operating Mode

This attribute consists of a set of flags which control the behavior and functionality of the IEC line. Each flag is one bit of this attribute. The bits are the following:

Data type:	Integer
Value:	0..65535
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	Read, conditional write (No limitations in balanced mode)
Bit 0:	Balanced mode handshake When this bit is 0, the sending of the handshaking messages (request, status of link, reset of remote link) are **NOT** restarted when a Request status of link message is received from the remote end. When the bit is 1, the sending of the handshaking messages is restarted when a Request status of link message is received. Note that if SYS600 is used in both ends, only one of them should have this bit set. This bit is meaningful only in the balanced modes.
Bit 1:	Redundant line behavior in balanced mode When the bit is 0, the redundant line behaves as described in the document Norwegian User Conventions for IEC60870-5-101. When the bit is 1, the redundant line behaves as described in the document Norwegian User Conventions for IEC 60870-5-101, except an ACK (FUNC=0) response is given instead of a NACK (Func=1) response when the master issues a test function for link request. In balanced mode, an automatic line switch is made, when the master issues a command to the backup line. Bit 1 is meaningful in balanced mode only.
Bit 5:	Link initialization in both directions required When this bit is 0, the data messages are accepted before the link has been initialized completely in both directions. This is the default mode. When this bit is 1, the data messages are not accepted until the communication link has been initialized in both primary and secondary directions. In this mode, in case of a received 'request status of link' message or a primary link reset due to timeouts, it is assumed that the secondary link is not initialized anymore. Bit 5 is meaningful in balanced mode only.

UI	UAL event Identification

The UI attribute is used to define the name for the line object, and it is used to identify the source of the UAL (user activity logging) events. This string is added to the identification information of all user activity events from this line object. A unique value within the node is preferred. The node level module name (also in attribute UI) is added to form the full identification information for the user activity event. If a line identifier is not needed, an empty string should be assigned to this attribute.

Data type:	String
Value:	String containing a line level identifier with maximum length of 16 characters
Indexing:	1..12 (if not used, node attribute will be referred)
Default value:	”.LINEx”, where x = line number
Access:	Read, write

4.3.2.4 IEC 60870-5-101 station object

The main purpose of the station layer is the protocol conversion between the IEC 60870-5-101 and the internal protocol of SYS600. The station objects also take care of the application level communication with the master.

The STA objects created in a NET unit perform the functions of the station object. Several STA objects of the type IEC devices are allowed on the same line. It is also possible that multiple stations share the same remote IP address.

The STA objects created in a NET unit perform the functions of the station object. Some attributes are used for the station configuration and others are used for device communication. The configuration attributes are presented in this chapter and the communication attributes are presented in the next one.

Station objects can be configured to use secure authentication using the attributes described in chapter 'Authentication attributes'. IEC 60870-5-101 Secure Authentication is based on IEC technical specifications 62351-5 and 60870-5-7. Users and their roles and keys are created to slave device on-line with IEC 60870-5-101 using symmetric or asymmetric methods.

The databases for user sets and necessary keys are created using separate tools (see SYS600 System Configuration manual, chapter 'Secure authentication using IEC/TS 62351-5' for more information). This database is called "key storage" in the descriptions of the authentication attributes. The key storage which is used by the PC-NET instance is defined with NET Node attribute KS (see System Objects manual for a detailed description). Key storage file is always encrypted.

Station attributes

The following attributes can be used for configuring the IEC 60870-5-101 Slave stations in SYS600.

IU	In Use
Indicates whether the station is in use (value 1) or not in use (value 0).	
Data type:	Integer
Value:	0 or 1
Default value:	0
Access:	No limitations

LI	Line Number
The number of the NET line the station is connected to.This attribute is also used for setting the number of the back-up line, if redundant IEC lines are used. Note that indexes 1 and 2, that is, the main and back-up line numbers, are switched when a line switch operation is executed.	
Data type:	Integer
Value:	1...12 (NET line numbering)
Indexing:	None if redundant lines are used Index 1 is for the number of the main line Index 2 is for the number of the back-up line
Access:	Read, conditional write

PA	Polling Address
The link address of the IEC 60870-5-101 station.	
Data type:	Integer
Value:	0...254, when PL attribute = 1 0…65535, when PL attribute = 2
Default value:	1
Access:	Read, conditional write

Address 255 is reserved for broadcast messages (PL=1).
Address 65535 is reserved for broadcast messages (PL=2).

Note that if the configuration contains multiple stations connected to the same line (LRU configuration), they all share a common polling address. If a different link address is required, an individual line and COM-port must be used for each station and the combination of these lines must be solved with hardware.

SA	Station Address

The station address of the IEC 60870-5-101 station, the common address of ASDU in an IEC message.

Data type:	Integer
Value:	0...255 , when SL attribute = 1 0...65535, when SL attribute = 2
Default value:	1
Access:	Read, conditional write

DR	Direction

States if the IEC Slave station acts as the station A (primary station) or as the station B (secondary station).

Data type:	Integer
Value:	0 or 1
Default value:	0 (secondary station)
Access:	Read, conditional write

PL	Polling Address Length

The length of the link address in octets.

Data type:	Integer
Value:	1 or 2
Default value:	1
Access:	Read, conditional write (line IU must also be 0 when writing)

SL	Station Address Length

The length of the station address (common address of ASDU) in octets.

Data type:	Integer
Value:	1 or 2
Default value:	1
Access:	No limitations

IL	Information Address Length

The length of the information object address in octets.

Data type:	Integer
Value:	1...3
Default value:	2
Access:	Read, conditional write

CL	Length of Cause of Transmission Information

The length of the cause of transmission field in an IEC 60870-5-101 message in octets. If the originator addresses are used, the value of this attribute should be set to 2.

Data type:	Integer
Value:	1 or 2

Table continues on next page

CL	Length of Cause of Transmission Information
Default value:	1
Access:	No limitations

AL	Allocation

Allocates the station to an application. When the AL attribute has the value 1, the station is reserved by the application specified by the AS attribute. All spontaneous messages from the station are sent to this application.

Data type:	Integer
Value:	0 or 1
Access:	No limitations

AS	Allocating Application

Specifies the allocating application of the station (see the AL attribute). The allocating application gets all spontaneous process data from the station. This application is also the only one that is allowed to set the device communication attributes.

Data type:	Integer
Value:	0...250, 0 = no application
Access:	Read-only, conditional write

When the AL attribute is set to 0, AS also gets the value 0.

MI	Message Identification

Object address of system messages.

Data type:	Integer
Value:	1...32760
Default value:	29000 + station number
Access:	No limitations

MS	Message Application

The number of the application, that is the receiver of the system messages generated by the line.

Data type:	Integer
Value:	1...250
Default value:	1
Access:	No limitations

SE	System Messages Enabled

Specifies whether the system messages generated by the NET and related to the station are sent to applications (value 1) or not (value 0). By using this attribute, it is possible to disable the system messages related to the station.

Data type:	Integer
Value:	0 or 1
Default value:	1
Access:	No limitations

CA	Command Address

The object address of the bit stream process object in the SYS600 process database, where unidentified messages are sent.

Data type: Integer

Value: 1...65535

Default value: 32000

Access: No limitations

The unit number (UN attribute) of the bit stream process object must be the same as the STA object number.

CD	Command Delay

The Command Delay attribute specifies the maximum delay for timestamped commands. If the timestamp of the incoming command message indicates that the transmission delay has been bigger than the value defined by this attribute, the command is not accepted. The attribute defines a time window in which the timestamped command is accepted.

The IEC Station attribute CC controls how the "summer time" bit of the incoming command is handled. See the corresponding description for more information.

The value of the NET node attribute TZ (Time zone) is used to compensate the timestamps of different time zones.

The timestamped control commands are normally used only in IEC60870-5-104, and not in IEC60870-5-101.

Value: 0...65535

Unit: Milliseconds

Default: 1000

Access: Read/Write

Example:

If STA1:SCD is 2000, only the commands with timestamps +-2000 ms around the current time of the slave computer is accepted.

ML	Maximum Message Length

The maximum amount of octets containing information objects. The link header including link address and The Application Protocol Control Information (APCI) fields and Data Unit Identifier fields are excluded from this value. This attribute is automatically set to a smaller value, if the total message length would exceed 253 bytes with the given value.

Data type: Integer

Value: 20...255

Default value: 253

Access:

The length of the incoming messages can always be at their maximum.

SR	Single Char Response

Enables or disables single char responses. If single char responses are enabled, the station object replies with a 0xE5 character as a link layer acknowledgment. If single char responses are enabled in the unbalanced mode, 0xE5 character is also replied instead of a "requested data not available" message to a poll.

Data type: Integer

Value: 0 or 1

Default value: 0 (single char responses disabled)

Access: Read, conditional write

AQ	ASDU Queue length

The length of the ASDU Queues which are used to buffer the data messages. Using this attribute it is possible to limit the amount of the buffered data if required by the master. The default value is the maximum length used in versions 9.4FP2 and older. If the AQ is increased to its maximum 3000, one queue may store up to 102000 time-stamped binary or double binary events (M_SP_TA_1/M_DP_TA_1).

Data type:	Integer
Value:	20..3000
Indexing:	No indexing
Default value:	200
Access:	Read/Write

Example 1:

```
#set STA1:SAQ1=50    ; max 50 messages is buffered per queue (1 or 2)
```

EA	Event buffer overflow Address

Defines the information object address of a single indication which is used to indicate the event buffer overflow situation.
With index 2 it possible to define a percentage of the buffer space. If a value different from 100 is defined, a single indication indicating the event buffer overflow is not resent until the event buffer space has gone below the defined level of the total buffering space at least once. See the example below. The total buffering space can be limited or increased with attribute AQ ASDU Queue length.

See the descriptions of:

- Bit 6 of the RM attribute
- PRI parameter of the EV attribute

Value:	Integer
	Index 1 or no index: 0..16777215
	Index 2: 0..100
Indexing:	1..2 or no index
	Index 1 or no index : Information object address
	Index 2 : Percentage of buffer space
Default value:	Index 1 or no index:32001, Index 2 = 100 (Percentage limit not used)
Access:	Read/Write

Example 1:

```
#SET STA1:SEA1=65535 ; Event buffer overflow event is sent to address
65535
#SET STA1:SEA2=80
```

When the event buffer overflow event has been sent, the same event is not sent again until the total amount of unsent ASDUs in class1 and class2 queues has gone under or equal to 80% of the queue size defined with attribute AQ ASDU Queue size.

XT	eXecute Timeout

The maximum time an execute command is waited for after a select command. The value is meaningful only if the bit 4 of the RM attribute is not set (select-execute checking for control commands) or bit 11 of the RM attribute is set (select-execute checking for setpoints). See the RM attribute description for further information.

Data type:	Integer
Value:	0... 65

Table continues on next page

XT	eXecute Timeout
Unit:	Seconds
Default:	30
Access:	No limitations

PC	Process Data Confirmation

Controls how the confirmation message is sent to the master station. The manual confirmation (value 0) means that the user has to confirm the incoming message from SCIL by setting the CF attribute. The automatic confirmation (value 1) means that the confirmation is done automatically by NET once the base system accepts the data message sent by NET. System and Application commands are not confirmed automatically if the attribute is set to the automatic test mode.

Data type:	Integer
Value:	0, 1
Default value:	0
Access:	Read, conditional write

TC	Time Synchronization

Determines the behavior of the slave device when it receives a time synchronization message as follows:

Data type:	Integer
Value:	0...3 Value 0 = The synchronization message is handled and the clock of the base system is set to the received time. The synchronization message is not sent to the process database. 1 = The clock of the base system is set to the received time and the synchronization message is also sent to the process database (to a bit stream process object with address as defined by the CA attribute). 2 = The clock of the base system is not set, but the synchronization message is sent to the process database. 3 = Synchronization messages are not handled at all.
Default value:	1
Access:	No limitations

IV	Invalid Time

The "invalid time" information in timestamped messages follows the value of this attribute. When a time synchronization is received from the IEC60870-5-101/IEC60870-5-104 line, this attribute is set to value 0 and timestamped messages are transmitted with a valid time. If the application writes value 1 to this attribute, timestamped messages are transmitted with an invalid time. If value 0 is written, messages are transmitted with a valid time. This attribute is useful, for example, if the connection to GPS making the synchronization is lost. If the IV attribute is not written from application, the "invalid time" bit behaves as described in the RM attribute bit 1.

Data type:	Integer
Value:	0 or 1
Indexing:	No
Access:	No limitations

RM	Running Mode

Consists of a set of flags that control the behavior and functionality of the IEC Slave station. Each flag is one bit of this attribute. The bits are as follows:

Bit 0:	The hour transmission method of the events from the slave station. When this bit is 0, the slave device sends a spontaneous clock synchronization message (ASDU 103) to the master whenever the hour changes. When this bit is 1, the synchronization message is not sent and the master can add its own time to events.
Bit 1:	Time synchronization method. When this bit is 0, the slave station waits for a synchronization command from the master station and marks the time stamps as invalid until the synchronization command is received. When this bit is 1, the synchronization message is not expected and the time stamps are not marked as invalid.
Bit 2:	Confirmation queue (significant only in the unbalanced mode). When this bit is 0, all the confirmation messages are put to the class 1 queue. When this bit is 1, confirmation messages are put to the class 2 queue.
Bit 3:	Handling of unrecognized commands. When this bit is 0, unrecognized command messages are ignored. When this bit is 1, unrecognized command messages sent by the master are forwarded to a bit stream process object with an address as defined by the CA attribute.
Bit 4:	Select-execute checking enable/disable. When this bit is 0, the validity of each select and execute command for "single command" or "double command" types is checked. A negative confirmation is automatically returned if: • the execute command is received without a preceding select command • the object address, ASDU type or value is not equal to the preceding select command • another select command is received (except for matching deactivation which is accepted) The length of the select-execute timeout is defined with the XT attribute. As mentioned, this check applies to single command and double command ASDUs only. Up to 10 select commands may be pending at the same time. When the bit is 1, the commands are not checked and the process objects are always updated. The value of XT is meaningless in this case for these ASDU types. See also RM bit 11.
Bit 5:	SQ=1 packing enable/disable. When this bit is 1, the data from ASDUs 1,3,9,11,13 is packed in SQ=1 style. The usage of this packing style may speed up the communication, if the addresses entered in ComTool contain blocks. In these blocks the addresses are incremented sequentially. This means that addresses like baseaddr, baseaddr+1 or baseaddr+2 are used. The SQ=1 packing is not supported for time tagged data. When this bit is 0, SQ=1 packing is not used and the packing is made in standard SQ=0 packing style.
Bit 6:	If this bit is 1 and error 13856 ICCC_ASDU_QUEUE_FULL is returned during writing to the EV attribute, the included timestamp is stored. A single indication message using this timestamp is transmitted automatically as the first event message when communication to the master proceeds. The information object address of the event is defined with the EA station attribute. Using the attribute EA, index 2, it is possible to define a percentage level which limits the resending of this event overflow indication. If this bit is 0, no action is taken when an error is returned.
Bit 7:	When this bit 0, the parameter ASDU types 110-112 (P_ME_NA_1, P_ME_NB_1 and P_ME_NC_1) received from the master update a bitstream process object defined with the application layer attribute CA. When this bit is 1, the incoming parameter ASDU types 110-112 are handled similarly to set-point command ASDUs 48-50 or ASDUs 61-63, that is, an analog input process object is updated.
Bit 8:	Exceptional command checking. If this bit is 1 and an exceptional command is received, an immediate response with cause of transmission 44..47 (unknown type identification, unknown cause of transmission, unknown common address of ASDU or unknown information object address) is sent back to the master. If this bit is 0, the cause of transmission of the response is defined in the MicroSCADA application.

Table continues on next page

Bit 9:	Clock message before each indication event. When this bit is 1, each time-tagged single and double indication event (M_SP_TA_1/M_DP_TA_1) is preceded with a spontaneous clock synchronization message (ASDU 103) containing a full timestamp. Bit 0 of this RM bit must not be set in order to use this feature. When bit 9 is 0, the transmission of the spontaneous clock synchronization message (ASDU 103) follows only the setting of bit 0. This is the default functionality.
Bit 10:	Delete oldest events in overflow. If this bit is 1 and event queue overflow occurs, a ASDU message containing the oldest events is deleted and the new event is placed to the same queue instead of returning the error 13856 = ICCC_ASDU_QUEUE_FULL. When bit 10 bit is 0, the oldest events are retained in the queue and error 13856 is returned when new events are written. This is the default functionality.
Bit 11:	Select-execute checking disable/enable for setpoints. When this bit is 1, the validity of each select and execute command for setpoint commands, that is, ASDU types 48-50 and ASDU types 61-63, are checked. A negative confirmation is automatically returned if: • the execute command is received without a preceding select command • the object address, ASDU type or value is not equal to the preceding select command • another select command is received (except for matching deactivation which is accepted) The length of the select-execute timeout is defined with the XT attribute. Up to 10 select commands may be pending at the same time. When the bit is 0, the commands are not checked and the process objects are always updated. This is default functionality for setpoints. The value of XT is meaningless in this case for these ASDU types. See also RM bit 4.
Data type:	Integer
Value:	0...65535, see above
Default value:	0
Access:	Read, conditional write

DC Diagnostic Counters

The values of the diagnostic counters which the NET unit keeps for the station. The counters have the following meaning:

1. Suspension information (0 = OK, 1 = suspended)
2. Suspension counter
3. Transmitted data messages
4. Transmitted command messages
5. Transmitted confirmation messages
6. Received data messages
7. Received command messages
8. Received confirmation messages
9. Received unknown messages
10. Received too long messages
15. Application response timeouts

Data type:	Integer
Value:	1...65535
Index range:	1...15
Access:	No limitations

OS	Object Status

The current status of the IEC station object. When value 1 is written to this attribute, the station object retransmits its current status code to the system message process object.

Data type:	Integer
Value:	when Read, 0 = OK_STATUS or non-zero value = communication is not normal at the moment
Access:	No limitations (write is possible only with value 1)

QI	Queue Information

Information from the class queues. Returns the amount of unsent ASDUs with matching values for COT (Cause Of Transmission) or TYPE ID (Type Identification).

When accessed, the formula for indices is:

100*OPERATION+FIRST..100*OPERATION+LAST

The different values of OPERATION:

OPERATION =	1 : class 1 ASDUs in which COT is in range FIRST..LAST 2 : class 2 ASDUs in which COT is in range FIRST..LAST 3 : class 1 + class 2 ASDUs in which COT is in range FIRST..LAST 4 : class 1 ASDUs in which TYPE_ID is in range FIRST..LAST 5 : class 2 ASDUs in which TYPE_ID is in range FIRST..LAST 6 : class 1 + class 2 ASDUs in which TYPE_ID is in range FIRST..LAST
Data type:	Integer
Value:	0...300000
Index:	1...6 (see above)

The last index is optional (see above).

Access:	Read-Only

Examples:

```
STA1:SQI103 ;Returns the amount of unsent ASDUs from class 1 with COT=3
;(Spontaneous)
STA1:SQI(320..336) ;Returns the amount of unsent ASDUs with COT=20..36
;(Interrogated/Group interrogated)
STA1:SQI513 ;Returns the amount of unsent ASDUs from class 2 with TYPE
;ID=13 (M_ME_NC_1)
STA1:SQI(699..701) ;Returns the amount of unsent ASDUs from with TYPE
;ID=99..101 (Interrogation commands)
```

ST	SYS Waiting Time

The maximum time that the slave station waits for a reply from the base system.

Data type:	Integer
Value:	0...60000
Unit:	Milliseconds
Default value:	5000
Access:	No limitations

MT	Maximum Delayed Response Time

The maximum time to delay response for writing data to the SD and EV attributes, if the number of items in the queue is greater than the value of the RW attribute.

Data type:	Integer
Value:	0...600 0 = Delayed response mechanism not used

Table continues on next page

MT	Maximum Delayed Response Time
Unit:	Seconds
Default value:	0
Access:	No limitations

A high value of this attribute may cause queuing of command procedure executions in the base system during communication disturbance. It is recommended to use value 0. If more than one NCC is configured to COM500i, value 0 must be used.

CC	Command Control

This attribute defines how the incoming timestamped control commands and time synchronization commands are handled.

See NET node attribute TZ (Time Zone) and IEC station object attribute CD (Command delay) for more information.

The term "summer time" is used in IEC standards as is a synonym for "daylight saving time".
The timestamped control commands are normally used only in IEC60870-5-104 and not in IEC60870-5-101.
Values of the "summer time" bit are:

0 = standard time
1= summer time/daylight saving time

Data type:	Integer
Default:	0
Value:	0... 1

Value 0 = The "summer time" bit of the timestamp in the incoming command is ignored. The timestamp of the incoming command is corrected with the value of the NET node attribute TZ (Time zone). In order to accept the incoming command, the difference between this timestamp and the arrival time (from local clock) of the message must not be bigger than the time window defined with the station attribute CD.

Value 1 = The "summer time" bit of the timestamp in the incoming command is expected to contain a correct value. It is taken into account in the clock setting or in the time comparison with the arrival time. The value of the TZ is also taken into account, which means that if the timestamp of the incoming command is in UTC time, the TZ attribute of the NET node value should contain the amount of minutes between the local time zone in winter time and the UTC time.

If the value of the "summer time" bit is different from the "daylight saving time" information of the local clock, the one hour difference is compensated before setting the clock and the comparison to the arrival time. In order to accept the control command, the difference must not be bigger than the time window defined with the station attribute CD.

SU	Summer Time

States whether summer time is used or not. With this attribute, the user can change the SU flag in the IEC time tag. The SU attribute can be used to tell a master system that the time tagged event uses summer time. If the master system does not use summer time, it is then able to change the time to its own time.

Data type:	Integer
Value:	0 or 1
Default value:	0 (summertime not used)
Access:	No limitations

RW **Reply Window Size**

Defines how many data items (binary values, analog values) can be written from the base system to NET without a reply or request from the master. If the slave station cannot send data items spontaneously to the master, it stores data into local buffers and creates a local reply to the base system and the execution of the SCIL program can continue. The slave station stores items until the number of items in local buffers is equal to RW. After that the slave station delays the replies to the base system until the number of items drops below RW again (data sent to master and reply received). This delay is configurable and it is defined by the MT attribute.

Data type:	Integer
Value:	0…10000
Default value:	10
Access:	No limitations

SF **Status Filtering**

This attribute specifies a filter time for returned error statuses of attribute writings/readings. If a non-zero value is given to the SF-attribute, the station object checks whether the same error status is to be returned within the time specified with SF, it is replaced with OK_STATUS for the SCIL application. This behaviour limits the error code amount in notification window and in the sys_error.log file. The status is not modified to OK_STATUS in following conditions:

1. SF is 0

2. The status is different from the previous status

3. The time defined with SF has been changed

If the previous returned status is modified to OK_STATUS and the status to be returned changes back to OK_STATUS, status
13889 = ICCC_DATA_WRITE_STATUS_CHANGES_TO_OK is returned.
This is needed to see the timestamp of the actual moment of the status returning back to OK_STATUS. If the feature provided with the SF attribute is used, The SCIL application should handle the mentioned status as OK_STATUS.
If the value of SF is 0, filtering is not used and the functionality is the same as in version 9.4 or older. The value of SF can be set on-line without taking the station object out of use and it is effective immediately. If index 2 of the SF attribute is read, it returns the amount of filtered error statuses from previous returned error. The feature can be used to verify the actual amount of data updates when error filtering is used (SF>0).

Data Type:	Integer
Value:	0..180
Unit	Seconds
Default:	0 (= filtering not used)
Access:	No limitations (index 2 is read-only)

Example:

#SET STA1:SSF=10

An error status from Station object 1 is printed out with 10 second interval. This configuration may be used with a station object connected to a IEC 60870-5-101 Slave line with the COM500*i* application. When the NCC connection is broken and the event queues are full, without this setting each new event or analog update would generate a printout which fills the notification window and the sys_error.log. With the example setting above, the amount of the printouts is much less and the off-line analysis of the problem situation is easier.

UA	UAL Event Used

Attribute UA defines whether the UAL (User activity logging) events are generated by the station object. Generation of the UAL events are recommended if the secure authentication is used, see attribute ZA. With ‘Standard logging’, all user status changes are logged. With ‘Special logging’, also all successful authentications, key change negotiations, authorization failures and error situations are logged. The event codes with 'Special logging' are supported by SYS600 but not necessarily, for example, with SDM600 product. In case the secure authentication is used and more detailed information from the system behaviour is needed, the usage of value 3 = 'Special logging' is recommended.

Data type:	Integer
Value:	0 = Disabled 1 = Standard logging 2 = Extended logging 3 = Special logging
Indexing:	No indexing
Default value:	1 (Enabled)
Access:	No limitations

UI	UAL Event Identification

The UI attribute is used to define the name for the station object, and it is used to identify the source of the UAL (user activity logging) events. This string is added to the identification information of all UAL events from this station object. A unique value within the node is preferred. The node level module name (also in attribute UI) is added to form the full identification information for the user activity event. The default value of the string is based on the translated station number which is equal to B-attribute TN (Translated Object Number) in the base system. If a station identifier is not needed, an empty string should be assigned to this attribute.

Data type:	String
Value:	String containing a station level identifier with maximum length of 16 characters
Indexing:	No indexing
Default value:	”.STA(TN=x)”, where x = translated station number
Access:	No limitations

4.3.2.5 Redundant line attributes

SYS600 provides support for redundant IEC 60870-5-101 Slave lines. This means that one IEC slave station can have two lines. One of them is the main line, initially meant to be the primary communication channel. The other line is the back-up line, meant to serve as a secondary communication channel in case the main line fails. If a communication disturbance is detected, the communication can be switched from the main line to the back-up line and vice versa without losing any messages to be sent to the master. A line switch is initiated by the master.

The main line and the back-up line can have different communication parameters, for example, baud rate. In addition to this, both of the lines can have dial-up configured.

When a redundant IEC 60870-5-101 Slave connection is configured, the following steps should be taken:

1. Define the main line.
2. Define the back-up line.
3. Define the station.

The following line attribute is used with redundant IEC 60780-5-101 lines:

RU	Redundant Line Station

This attribute defines the number of the STA object connected to redundant IEC lines. This attribute should be set both for the main and back-up lines. The information provided by this attribute is needed when a line switch operation is executed. Value 0 indicates that redundant lines are not used.

Data type:	Integer
Value:	0...255
Index range:	1...12 (NET line numbering)
Default value:	0
Access:	Read, conditional write

The following station attribute is used with redundant IEC 60780-5-101 lines:

LI	Line Number

The number of the back-up line is set to index 2 of the LI attribute. Refer to the LI attribute presented earlier in the Section 4.3.2.3.1.

Example:

Example:

```
#SET STA1:SLI(2) = 5
```

When using redundant IEC 60870-5-101 lines, the IEC master must provide the corresponding functionality. The implementation of the line switch mechanism is described in Section 5.4.5.

The backup line must be defined with the LI attribute right after the station has been created.

There can be only one IEC station for a pair of redundant lines.

When using redundant IEC 60870-5-101 lines, the ASDU number 128 is used for communication line activation and the ASDU M_SR_NA_1 Parameter Byte String cannot be used.

4.3.2.6 Authentication attributes

This chapter describes only the attribute interface related to secure authentication. See SYS600 System Configuration Manual, chapter 'Secure authentication using IEC/TS 62351-5' for detailed steps for configuring the system for secure authentication according to IEC/TS 62351-5.

ZA	Authentication Used

The ZA attribute defines whether secure authentication is used by the station object or not.

Value:	0 = Disabled 1 = Enabled, pre-shared update keys 2 = Enabled, update key negotiation
Indexing:	No indexing
Default value:	0 (Disabled)
Access:	Read/write

ZG **Aggressive Mode**

The ZG attribute defines whether the aggressive mode of authentication is used by the station object. The aggressive mode uses less bandwidth and using it is recommended. The value of this attribute is meaningless if authentication is disabled (see attribute ZA). Modifying this attribute is possible only if it is enabled in the key storage using the setting 'Allow external modification of security attributes' (see attribute DZ, index 255).

Value:	0 = Aggressive mode disabled 1 = Aggressive mode enabled
Indexing:	No indexing
Default value:	1 (Enabled)
Access:	Read, write possible if the modification is enabled in the key storage

ZT **Key Storage Id**

The ZT attribute is used to define the keys and user set of the STA object in the key storage. In case there is a need to change the station number and/or its translated number TN and the corresponding user set and key configuration is already configured, keeping the original value in the ZT attribute associates the existing user set with the new station object. The ZT value must be unique within the STA objects accessing the same key storage. Error is returned when setting to a reserved value is attempted. Value = 0 means that the station object is not attached to any user set and the enabling of the authentication using attribute ZA is not possible.

Value:	Identification of the STA object in the key storage, range 0..65535
Default value:	Same as the TN of the STA object
Indexing:	No Indexing
Access:	Read/write

ZU **Default User**

The ZU attribute is used to define the user in case the user is not or cannot be received from the MicroSCADA application. This attribute is meaningless in STA objects connected to slave lines.

Value:	When read, the number of the active user is returned When written, number or the name of the wanted active user
Default value:	1 (Default user)
Indexing:	No indexing
Access:	Read/write

ZR **Authenticated Users**

The ZR attribute is used to read the user numbers and names defined in the key storage for the station. See also the ZU attribute for defining the default user. This attribute is indexed using the user number and it is not supported by the System Configuration tool.

Value:	String containing a user name with the maximum length of 32 characters.
Indexing:	1..65535 (User number) If index = 0 is given when read, a vector of configured user numbers for the STA object is returned. If only one user is configured, the returned value is a scalar of type integer. If ZA is 0 or no users is configured, no object is returned.
Access:	Read

ZI **Association Id**

The ZI attribute is used to define the association identification value of the user. This attribute is used to fully identify the user. The attribute is indexed using the user number and is not supported by the System Configuration tool.

Value:	The association id of the user defined by the index
Default value:	0
Indexing:	1..65535 (User number)

ZV	Authentication Vector

The ZV attribute defines the constants used by the specific user of the station object. Modifying these constants is possible but it may require some testing to be usable with the remote IED and its configuration. The given index defines the user, value 0 has a special meaning. The values of the vector cannot be modified one-by-one but when written, only a vector containing all values is accepted.
Fourth parameter of ZV0 is a bitmask which defines compatibility flags due to different interpretations of IEC62351-5 standard. Bit 0 affects MAC calculation of critical messages and bit 1 affects the handling of CSQ (Challenge sequence number) field. When bits 0 and 1 are set, the functionality is similar to DNP3 SAv5. In case the remote system or its software is North American originated, setting of bits 0 and 1 may be necessary.

The session key length indicated with ZV(1) automatically adapts to the key length defined in master when session keys are negotiated. Length range is 16..32 bytes.
If the interval defined with ZV(8) has expired twice without session key renegotiation between the expirations, the session keys for the user are invalidated and the corresponding UAL event is reported. After this, no critical operations for the user are authenticated in either directions and session keys must be renegotiated. See also attribute ZP, index 1.
ZV(6) is not used with IEC60870-5-101/104 slave protocols but corresponding threshold values is used instead. See attributes DZ and ZH, index 11.
If session key negotiation fails or session key status is requested repeatedly up to limit defined with ZV(11), a special warning is logged to the UAL event log. Value range 0..255, default 5.
The attribute is indexed using the user number and is not supported by the System Configuration tool. Index 0 contains special compatibility flags applied for all users.

Value:

For index 0, vector of max 4 integers
ZV(1) User name null termination
(1 = User name transmitted without null termination (default))
(2 = User 'Common' with null termination, others without)
(3 = User names transmitted with null termination)
ZV(2) Authority certification key length with SHA-1 Update Key Change Method
(16= First 16 bytes used in MAC calculation)
(32 = All 32 bytes used in MAC calculation (default))
ZV(3) Not used at the moment
ZV(4) Compatibility bitmask
(Bit 0 not set = European interpretation of IEC62351-5 MAC calculation (default))
(Bit 0 set = North American interpretation of IEC62351-5 MAC calculation, same as in DNP3)
(Bit 1 not set = European interpretation of IEC62351-5 CSQ handling (default))
(Bit 1 set = North American interpretation of IEC62351-5 CSQ handling, same as in DNP3)
For indices > 0, vector of 11 integers
ZV(1) Session key length
ZV(2) Key wrap algorithm of the session keys
(2=AES-256)
ZV(3) Challenge data length critical request
ZV(4) HMAC algorithm of the critical requests
(2 = SHA-256)
ZV(5) HMAC length
(8 = SHA-256 serial)
(16 = SHA-256 network)
ZV(6) Not used, see DZ(11) and ZH(11)
ZV(7) Session key change count
ZV(8) Key change interval in seconds
ZV(9) Challenge data length session key status
ZV(10) Challenge data length update key reply
ZV(11) Maximum Session Key Status Count (meaningful in slave only)

Table continues on next page

ZV	Authentication Vector
Default values:	Index 0: ZV(1) : 1 ZV(2) : 32 ZV(3) : 0 ZV(4) : 0 Indices >0: ZV(1) : 16 ZV(2) : 2 (AES-256) ZV(3) : 8 ZV(4) : 2 (SHA-256) ZV(5) : 8 (SHA-256 Serial) ZV(6) : 2 ZV(7) : 1000 ZV(8) : 900 ZV(9) : 8 ZV(10) : 32 ZV(11) : 5
Indexing	1..65535 (User number), 0 (compatibility flags)
Access:	Read-only

Example 1:

```
@USERS = STA1:SZR
#LOOP_WITH I=1..LENGTH(%USERS)
     ; Change each user to use SHA-256 and AES-256 in TCP mode, session
key change interval 120 secs
    @USER=%USERS(%I)

    #SET STA1:SZV'USER'=(16,2,8,2,16,2,1000,120,8,32,5)

#LOOP_END
```

ZS	Authentication Status
The ZS attribute returns the current values of the authentication parameters and states of the specific user. This attribute is implemented only for debugging purposes and modifying these values directly is not possible. The attribute is indexed using the user number and is not supported by the System Configuration tool.	
Value:	Vector of 8 integers ZS(1) State of the authentication ZS(2) Not used at the moment ZS(3) Not used at the moment ZS(4) Challenge sequence number CSQ ZS(5) Key change sequence number KSQ ZS(6) Key status KST ZS(7) Last error in ZS(8) Last error out Values of Key Status in ZS(6) are : 1 = OK 2 = Not initialized 3 = Communication failure 4 = Authentication failure
Default value:	ZS(1) : 0 ZS(2) : 0 ZS(3) : 0 ZS(4) : 0 ZS(5) : 0 ZS(6) : 2 ZS(7) : 0 ZS(8) : 0

Table continues on next page

ZS	Authentication Status
Values of ZS(1)	0 : Initial State 10 : Wait for Key Status 11 : Wait for Key Change Confirmation 12 : Security Idle 13 : Wait for Reply 14 : Wait for User Change Response 15 : Wait for Update Key Reply 16 : Wait for Update Key Confirm
Indexing:	1..65535 (User number)
Access:	Read-only

ZO	User Roles

The ZO attribute is used to return user numbers and their roles for the authentication. The user set for the station is created using the authority tool and stored to the key storage file.

Value:	String containing a user role with maximum length of 32 characters. Standard roles are: 0 = 'Viewer' (has 'Monitor Data' permission) 1 = 'Operator' (has 'Monitor Data' and 'Operate Controls' permissions) 2 = 'Engineer' (has 'Monitor Data', 'Change Config', 'File access' and 'Local login' permissions) 3 = 'Installer' (has 'Monitor Data', 'Change Config', 'Change code', 'File Access' and 'Local login' permissions) 4 = 'SecAdm' (has 'Change Security Config', 'Change code' and 'Local login' permissions) 5 = 'SecAud' (has 'Monitor Data' and 'Local login' permissions) 6 = 'RBacMnt' (has 'Monitor Data', 'Change Config' and 'Change Security Config' (roles only) permissions See attribute CR for more information. Permissions applicable to IEC 60870-5-101 slave implementation in SYS600 are 'Monitor Data', 'Operate Controls' and 'Change Config'.
Default value:	Empty string
Indexing:	1..65535 (User number)
Access:	Read-only

ZN	Outstation Name

The ZN attribute is used to return the name of the outstation. This attribute is preconfigured using the Authority Tool and must match the value configured to master/slave.

Value:	String containing an outstation name with maximum length of 128 characters
Default value:	Empty string
Indexing:	No indexing
Access:	Read-only

ZD **Authentication Diagnostics**

The ZD attribute provides user specific counters for diagnostic purposes. These counters can be used to verify the activity of each user and to give information in problem situations. The same counters for all users can be read from the attribute DZ.

Value:	Vector of 17 integers (0..65535) ZD(1) Unexpected Messages ZD(2) Authorization Failures ZD(3) Authentication Failures ZD(4) Reply Timeouts ZD(5) Rekeys Due to Authentication Failure ZD(6) Total Messages Sent ZD(7) Total Messages Received ZD(8) Critical Messages Sent ZD(9) Critical Messages Received ZD(10) Discarded Messages ZD(11) Error Messages Sent ZD(12) Error Messages Received ZD(13) Successful Authentications ZD(14) Session Key Changes ZD(15) Failed Session Key Changes ZD(16) Update Key Changes ZD(17) Failed Update Key Changes
Default values:	ZD(1) ..ZD(17) : 0
Indexing:	1..65535 (User number)
Access:	Read/write (write only value 0)

ZP **Authentication Parameters**

The ZP attribute defines authentication parameters which has an effect on all users configured for the station object. The given index specifies the parameter.
ZP(1) defines the function 'Expected Session Key change count' defined in 62351-5 standard. If amount of 2*ZP(1) critical or non-critical ASDUs is transmitted without session key negotiation messages from master, session keys of all users are invalidated and no further critical messages are accepted before new session keys are negotiated. The default value for ZP(1) is 1000. In case the ASDU transmission rate is high, bigger value is worth to be considered but the value should still be in line with the master configuration. If ZP(1)value is 0, function is not used, that is, the session keys are not invalidated because of the message count. Master devices usually have corresponding configuration parameter to trigger the session key negotiation after certain amount of received ASDUs. If this parameter is bigger than 2*ZP(1) in slave, it is possible that the keys are invalidated unexpectedly and next critical request such as a control command is rejected. Slave device does not explicitly report when it has invalidated the session keys.
The attribute is not supported by the System Configuration tool. No other indices but 1 is used the moment.

Value:	ZP(1) Expected session key change count
Default values:	ZP(1) : 1000 (range 0..2147483647, value=0 means disabled)
Indexing	1..1
Access:	Read/Write

DZ	Diagnostics of Authentication

The DZ attribute provides diagnostic counters of authentication related functions. These counters can be used to verify the activity of communication concerning data which requires authentication. The same counters can be read as user specific from the attribute ZD.
Index 255 has a special meaning. If DZ(255) returns 1, the editing of the attributes CR and ZG is allowed by a special setting in the key storage.
Special sets of the same counter are provided using indices 101..117 and 201..217. Indices 101..117 return the same diagnostic counters calculated from the beginning of the PC_NET start-up, without resetting. Indices 201..217 return the next (unreset) counter value which will trigger the special threshold action defined in IEC TS 62351-5. Special actions are caused only by counters 2 (Authentication failures), 4 (Reply Timeouts), 5 (Rekeys due to to Authetication Failure) and 11 (Error Messages Sent). See example below for special actions.

Value:	Vector of 17 integers (0..2147483647) DZ(1) Unexpected Messages DZ(2) Authorization Failures DZ(3) Authentication Failures DZ(4) Reply Timeouts DZ(5) Rekeys Due to Authentication Failure DZ(6) Total Messages Sent DZ(7) Total Messages Received DZ(8) Critical Messages Sent DZ(9) Critical Messages Received DZ(10) Discarded Messages DZ(11) Error Messages Sent DZ(12) Error Messages Received DZ(13) Successful Authentications DZ(14) Session Key Changes DZ(15) Failed Session Key Changes DZ(16) Update Key Changes DZ(17) Failed Update Key Changes
Default values:	DZ(1) ..DZ(17) : 0 DZ(255) : 0 (attribute editing not allowed)
Indexing :	255 : editing of the attributes CR and ZG is allowed by a special setting in the key storage 1..17 : Diagnostic counters mentioned above, is reset 101..117 : Diagnostic counters mentioned above, is unreset 201..217 : Diagnostic counters mentioned above, next threshold value for special action
Access:	Read/write (write only value 0 for indices 1..17)

Example (all special actions):

If DZ(102), authentication failures, is updated to be bigger than DZ(202), action = session keys are invalidated to status AUTH_FAIL + counter 'rekeys due to authentication failures' is incremented.

If DZ(105), rekeys due to authentication failures, is updated to be bigger than DZ(205), action = TCP Connection (IEC60870-5-104 only) is closed.

If DZ(104), reply timeouts, is updated to be bigger than DZ(204), action = session keys are invalidated to status COMM_FAIL.

If authentication failure occurs and DZ(111), error messages sent, is bigger than DZ(211), action = error reply is not sent anymore.

When next threshold value is reseted, for example, because of action or successful session key negotiation, threshold value defined with attribute ZH is added to the current unreset value of the diagnostic counter in question.

ZH **Security statistics threshold**

The ZH attribute defines the threshold values for diagnostic counters of authentication related functions. Some indices are used also as threshold value for special actions, see description and example of attribute DZ. This means no other values but ZH(2), ZH(4), ZH(5) and ZH(11) are meaningful.
At the moment, transmission time-tagged security statistics ASDU S_IT_TC_1 <41> is not supported but corresponding statistic values can be read as diagnostic counters to SCIL application from attribute ZD (user specific) and DZ (not user specific).

Value: Vector of 17 integers (0..65535)
ZH(1) Unexpected Messages
ZH(2) Authorization Failures
ZH(3) Authentication Failures
ZH(4) Reply Timeouts
ZH(5) Rekeys Due to Authentication Failure
ZH(6) Total Messages Sent
ZH(7) Total Messages Received
ZH(8) Critical Messages Sent
ZH(9) Critical Messages Received
ZH(10) Discarded Messages
ZH(11) Error Messages Sent
ZH(12) Error Messages Received
ZH(13) Successful Authentications
ZH(14) Session Key Changes
ZH(15) Failed Session Key Changes
ZH(16) Update Key Changes
ZH(17) Failed Update Key Changes

Default values: ZH(1) : 3
ZH(2) : 5
ZH(3) : 5
ZH(4) : 3
ZH(5) : 3
ZH(6) : 100
ZH(7) : 100
ZH(8) : 100
ZH(9) : 100
ZH(10) : 10
ZH(11) : 10
ZH(12) : 10
ZH(13) : 100
ZH(14) : 10
ZH(15) : 5
ZH(16) : 1
ZH(17) : 1

Indexing : 1..17 : Threshold value for the diagnostic counters mentioned above

Access: Read/write

CR	Critical Requests

The CR attribute defines which requests are critical. The modification of this vector is possible but not recommended.
The default values follow the IEC TS 60870-5-7. Modifying this attribute is possible only it is enabled in the key storage using the setting 'Allow external modification of security attributes' (see attribute DZ, index 255). In IEC 60870-5-101, control commands are sent using ASDUs 45..51 (without time tag) . The required permissions for a user are 'Operate Controls', see attribute ZO for more information. Index of the CR attribute defines the ASDU number in question.
This attribute is not supported by the System Configuration Tool.

Value:	Vector of 255 integers with values 0 = not critical 1 = critical 128 = not applicable
Default values:	Required permissions are given after the function name. If not mentioned, 'Monitor data' permissions for the function are adequate. Required permissions are given after the function name. If not mentioned, 'Monitor data' permissions for the function are adequate. CR(1..44) : 0 (Indication ASDUs) CR(45..51) : 1 (Non-timestamped command ASDUs)('Operate controls') CR(52..57) : 0 (ASDUs not defined) CR(65..69) : 0 (ASDUs not defined) CR(70) : 0 (End of initialization ASDU) CR(71..80) : 0 (ASDUs not defined) CR(81..95) : 128 (Authentication ASDUs) CR(96..99) : 0 (ASDUs not defined) CR(100..102) : 0 (Interrogation and read commands) CR(103) : 1 (Time synchronization command)('Operate controls') CR(104) : 0 (Test command without time tag) CR(105) : 1 (Reset process command)('Operate controls') CR(106) : 0 (Delay acquisition command) CR(108..109) : 0 (ASDUs not defined) CR(110..113) : 1 (Parameter setting and activation) ('Change config') CR(114..119) : 0 (ASDUs not defined) CR(120..126) : 1 (File Transfer) ('File access') CR(127..255) : 0 (ASDUs not defined or private range)
Indexing :	Index 1..255 (defines the ASDU number)
Access:	Read, write if allowed by the key storage config

4.3.2.7 Autodialing attributes

SYS600 provides support for the Autocaller functionality of the IEC 60870-5-101 Slave protocol. Autocaller is a modem with functions for automatic dial-up. The dial-up can be initiated by the IEC master or the IEC slave.

The Autocaller must use the AT (Hayes) command set. Note that when using odd or even parity, the modem must support 11-bit word length. In some cases, this feature must be enabled by using the AT commands. Refer to the documentation of the modem in use for further details.

The following Autocaller attributes are valid for the IEC 60870-5-101 Slave lines:

AC **Autocaller Enabled**

The AC attribute states whether an Autocaller is connected to the line (value 1) or not (value 0).

Data type:	Integer
Value:	0 or 1
Default value:	0
Access:	No limitations

AS **Autocaller State**

This attribute indicates the state of the Autocaller.

Data type:	Integer
Value:	0...4 0 = IDLE, ready to make a call 1 = CONNECTED, transmission is activated 2 = BUSY, Autocaller is dialing 3 = INITIAL, Autocaller is uninitialized 4 = CONFIGURE, the IU attribute of the line is set to 0
Default value:	0
Access:	Read-only

CL **Connection Time Limited**

This attribute determines whether a time limit has been set to the connection (value 1) or not (value 0). The maximum duration of the connection is determined by the CT attribute.

Data type:	Integer
Value:	0 or 1
Default value:	1
Suggested value:	A time limit is necessary on certain radio telephone lines. Limiting the connection time may also be good practice in other cases, if there is a risk that the connection is not otherwise broken.
Access:	No limitations

CT **Connection Time**

The maximum time that a connection is allowed to last (in seconds). This attribute is significant only if time limiting is activated (CL = 1).

Data type:	Integer
Value:	0...600
Unit:	Seconds
Default value:	120
Access:	No limitations

CN **Connection**

The CN attribute is used for dialing devices from the NET and for breaking telephone connections. This attribute has significance only in the unbalanced mode.

A call to a station or workplace is initiated by writing the phone number to the CN attribute. The NET unit then commands the autodialing modem to dial the number. The success of the dialing is reported as a system message. Writing an empty string to CN breaks the connection.

Data type:	Text
Value:	Text string of maximum 25 characters
Default value:	Empty text string

Table continues on next page

CN **Connection**

Access: No limitations

Example:
```
#SET NET1:SCN5 = "123456789S11"
```

CS **Connected Station**

The link address of the station a NET unit is communicating with.

Data type: Integer

Value: 0...65535
0 = Autocaller not defined or no communication

Default value: 0

Access: Read-only

DD **Radio Disconnection Delay**

Delay between the last data transfer and the line disconnection.

Data type: Integer

Value: 0...32767

Unit: Seconds

Default value: 0

Access: No limitations

MC **Modem Command**

Using this attribute, a modem can be controlled directly from SCIL with the AT/Hayes commands. When an AT command is written to the MC attribute, it is transmitted to the modem on the line. The response from the modem is read using the same attribute.

Data type: Text

Value: Text string, an AT/Hayes command

Default value: 0

Access: No limitations

Example:
```
#SET NET1:SMC3 = ("AS0?")'
```

PU **Pulse Dialing**

This attribute determines the dialing principle used.

Data type: Integer

Value: 0 = tone dialing
1 = pulse dialing

Default value: 0

Access: No limitations

RC	Remote Calls Enabled

The RC attribute states whether remote calls are enabled on a line, meaning that the NET unit can be called from the stations connected to the line in question. This attribute applies to lines with autocaller (AC = 1).

Data type:	Integer
Value:	0 = remote calls not enabled 1 = remote calls enabled
Default value:	0
Access:	No limitations

RW	Radio Connection Wait Time

Normally, the DCD (Data Carrier Detect) signal is used to indicate an active connection. However, there are cases where this is not possible, for example, on radiotelephone lines using half-duplex links. The RW attribute defines the waiting time in seconds in such a situation from the finishing of the dialing until the transmission is started.

Data type:	Integer
Value:	0...32767
Unit:	Seconds
Default value:	0
Access:	No limitations

SR	Autocaller AT S Register

The S registers used by the Autocallers follow the AT (Hayes) de facto standard.

All the Autocallers that use the AT command set have a number of S registers. The number of registers used and the meaning of the individual registers varies slightly from one Autocaller model to another. The contents of the S registers are therefore not described in this document. Refer to the modem manuals.

Using the SR attribute, the S register number 2, 6, 7, 8, 9, 10, 11 and 12 are accessed. By using the MC attribute (see above), other S registers can also be accessed. The S registers 11 and 12 cannot be set.

Data type:	Integer
Value:	See the modem manuals
Indexing:	Seconds
Access:	100 * line number + register number

Example:
The S register number 6 of line 2 in NET1 is set = 4:

```
#SET NET1:SSR206 = 4
```

4.3.2.8 File transfer attributes

The IEC file transfer feature transmits all types of files between the relay and the SYS600 computer. Only one transmission per one STA object can be active at any one time. If another file transfer request is made while the first file transfer is in progress, the status code is returned and the second file transfer progress does not start.

FI	File Information

The FI attribute initializes the file transfer system with the required base information.

Value:	Vector of 5 integers 1 = internally used 2 = SEGMENTCOUNT 3 = internally used 4 = QUEUE (unbalanced slave) 5 = PRIORITY (slave) 6 = Section request delay IN (0.1 seconds) 7 = Section filling delay OUT (0.1 seconds)
Indexing:	1..7
Access:	Read/Write
Default value:	1 = 0 2 = 8 3 = 0 4 = 2 5 = 3 (1 is lowest value) 6 = 0 (0 milliseconds) 7 = 10 (1000 milliseconds)

Example:

The following example defines five segments in each file section.

```
#SET STA'sta':SFI(2)=5
```

FD	File Directory

The FD attribute defines to which directory the received files are stored. PC_NET interrupts file receiving if the directory does not exist or if it is write-protected.

Value:	The string contains a valid directory name with the maximum length of 50 characters.
Access:	Read/Write
Default:	C:\TEMP

Example:

The following example defines C:\SC\DATA to the active directory.

```
#SET STA'sta':SFD="C:\SC\DATA"
```

FF	File Transmission Status

The FF attribute indicates the status of file transmission.

Value:	0 = Free to start or the previous transmission is completed 1 = Transmit in progress 2 = Timeout in PC_NET 3 = Not used 4 = Invalid directory or file name 5 = File is not available in the remote end 6 = Service is not available, internal error 7 = Transfer aborted 8 = File reading or writing failed
Access:	Read, Write is allowed when the value is not 6.
Default value:	0

If transmission is in progress when the FF attribute is written, the transmission is aborted, and the file is closed. It does not have any other effects, except the value is set to zero (0).

FT	File Timeout

The FT attribute defines the maximum delay for incoming ack section or ack file request. If the time expires, PC_NET interrupts the file transmission.

Value:	0..255
Unit:	Seconds
Access:	Read/Write
Default value:	30 seconds

FB	File Bytes

The FB attribute returns both counts of received or transmitted bytes from the beginning of the file transfer session (index 1) and the file size (index 2). At the beginning of the file reception, the value of index 1 is automatically set to zero (0). The value of index 2 is updated when the FB attribute is written, and the correct file index is given.

Value:	0..4294967295
Indexing:	When read 1 = Number of bytes (DWORD), out 2 = File size in bytes (DWORD), out 3 = File name in relay (DWORD), out 4 = Number of bytes (DWORD), in 5 = File size in bytes (DWORD), in 6 = File name in relay (DWORD), in
Access:	Read
Default:	0

FN	File Name

The FN attribute collects the information address of a certain file in the file system. When the remote end requests the directory, the information address is reported as a real file, not as a subdirectory (FOR = 0).

Value:	When written IOA NAMEOFFILE FILETYPE STATUSOFFILE NAMEINFILESYSTEM SIZEINFILESYSTEM DATEINFILESYSTEM MSECSINFILESYSTEM When read NAMEINFILESYSTEM When read
Indexing:	When read, (0..299) Queue out: index = 100+FILENUM Queue in: index = 200+FILENUM

If the FILENUM offset is bigger than the number of files in the list, status code 13887 ICCC_NO_SUCH_FILE is returned.

For more information about status codes, see Section 5.8.

Access:	Read/write

Where when writing

IOA

Information object address.

Type: DWORD

NAMEOFFILE

Defines the name of file field in the file transfer messages.

Type: WORD

FILETYPE

Type: WORD

Values: 1 = Transparent file

Other types are not supported at the moment.

STATUSOFFILE

Type: BYTE

Values: 0 = File waits for transfer

NAMEINFILESYSTEM

String contains a valid file name in the disk with maximum length of 100 characters. If the string is empty, the file is deleted from the list.

SIZEINFILESYSTEM (optional)

The file size.

Type: DWORD

DATEINFILESYSTEM (optional)

Creates file's timestamp (seconds from 1.1.78).

Type: TIME

MSECSINFILESYSTEM (optional)

Creates file's timestamp (milliseconds).

Type: WORD

Example of writing:

```
#SET STA'sta':SFN=(1000, 1,1,0, "error.log", 5000, clock)
```

Example of reading:

The second file is read from the outgoing file list and returns the error.log file.

```
STA'sta':SFN(102)
```

FV	File Values

The FV attribute checks the status of the defined file. During the reading process, the attribute's index defines where the file value is taken, either from the outgoing files list or from the incoming files list.

Value:

Vector
IOA
NAMEOFFILE
FILETYPE
CTRLANDSTATUSOFFILE
SIZEINFILESYSTEM
DATEINFILESYSTEM
MSECSINFILESYSTEM

Indexing:

When read, word (0..65535)
Queue out: index = 100+FILENUM
Queue in: index = 200+FILENUM
If the FILENUM offset is bigger than the number of files in the list, the returned data type is "NONE"

Access:

Read
For more information about status codes, see Section 5.8.
IOA
Information object address
NAMEOFFILE
Defines the name of file field in the file transfer messages.
Type: WORD
If value = 0 is given for both IOA and NAMEOFFILE, a unique value is assigned.
FILETYPE
Type: WORD
Values: 1 = Transparent file
Other types are not supported at the moment.
CTRLANDSTATUSOFFILE
Type: WORD.
The upper byte is a control byte.
Bit 0 : Internally used
Bit 1=0: Not yet transmitted
Bit 1=1: All sections transmitted
Bit 2=0: Transmission not acknowledged by remote
Bit 2=1: Transmission acknowledged by remote
The lower byte is equal to status of file (SOF)
Bit 7=0: File waits for transfer (FA)
Bit 7=1 Transfer of this file is active (FA)
SIZEINFILESYSTEM
Type: DWORD
The file size.
DATEINFILESYSTEM
Type: TIME
Creates file’s timestamp.

Example:

Reading the second file from the outgoing file list.

```
STA'sta':SFV(102)
3 ;Information object address
3 ;Name of file
1 ;Transparent file
1536 ;CTRL=(transmitted, acknowledged), SOF=0
10006 ;Filesize=10006 bytes
847636153 ;Timestamp
```

For more information about status codes, see Section 5.8.

SS	Send Subdirectory
The SS attribute initiates the transmission of the directory listing from the specified address.	
Value:	IOA, (COT) vector
Access:	Write, Where IOA Type: DWORD Is not in use at the moment COT (optional) Type: BYTE Defines the cause of transmission. If the value is not given, value three (3) is used.

Example:

The following example initiates the file directory's transmission by using cause of transmission 3.

```
#SET STA'sta':SSS=(1,3)
```

For more examples on communication system configuration, see Appendix A.

4.4 After configuration

For each input signal received from the process devices the process database should contain a process object whose value changes when process data is received. The change activates an event channel, which in turn starts a command procedure. The command procedure changes a value in the NET unit. From the NET unit, data can be transferred to the IEC master through the communication media.

Besides the configuration of the base system and the communication system the following are also needed:

1. Configure the IEC master.
2. Configure the base system for process communication.
3. Configure the process units.
4. Define the cross-references for signal rerouting, if COM500*i* is used. For more information, see SYS600 COM500*i* User's Guide.
5. Create and define the input and output process objects for the process communication. This is usually done when creating the station picture by using standard functions from an application library.
6. Define event channels for the process objects.
7. Define command procedures for the event channels. If COM500*i* is used, the command procedures that are already in it can be used. Otherwise, the command procedures are programmed. For more information on how to program the command procedures and the values of the attributes, see Section 5.

4.5 How to test the configuration

When the slave and master stations have been physically tested and the configuration has been completed, the connection and configuration can be tested based on the following methods:

- With the IEC 60870-5-101 protocols Request to Send (RTS) is set to the active state before transmission and the transmission starts from the raising edge of the Clear to Send (CTS) signal. If the raising edge of Clear to Send (CTS) does not occur and the CTS signal stays inactive within specified time, a line error is reported. If Clear to Send (CTS) is active all the

time, the transmission starts after DE+TW milliseconds from the raising edge of the RTS signal. DE and TW are line attributes of unit milliseconds.
- In order to receive messages, the Carrier Detect (DCD) signal should be in the active state or the line attribute CB should be set to 0 to ignore the state of the DCD signal.
- The Data Terminal Ready (DTR) signal is kept in signaled state when the line in use. This signal can be used to control the fallback switches in a HSB configuration
- Diagnostic counters. When the communication between the slave and master is running properly and data is moving on the line, the diagnostic counters indicating the number of received and transmitted data messages should be incrementing.
- Object status. The OS attribute of the IEC slave station should be 0.
- By connecting a protocol analyzer supporting the IEC 60870-5-101 standard to the line.

For MicroSCADA version 9.3 and newer, the protocol analyzer included in PC-NET can be used. See the NET line attributes AO and AU in the SYS600 System Objects manual.

One possible way to test the configuration is to use SYS600 as the IEC master/slave. In this case the base system and communication system configuration for the IEC 60870-5-101 Slave line and station(s) have to be made. The IEC master can even be in the same computer. Another benefit of using SYS600 as the test master is that the application pictures can be reused. However, the process database has to be recreated to an IEC.

4.6 Serial cable wiring diagram

When connecting a remote partner to a IEC line using a direct serial cable, the wiring is illustrated in Figure 10. If the used port is a virtual COM port or easier cabling is needed, the line attribute CM provides the possibility to simulate DCD and CTS changes. The line attribute CM is available in SYS600 version 9.3 FP2 and newer.

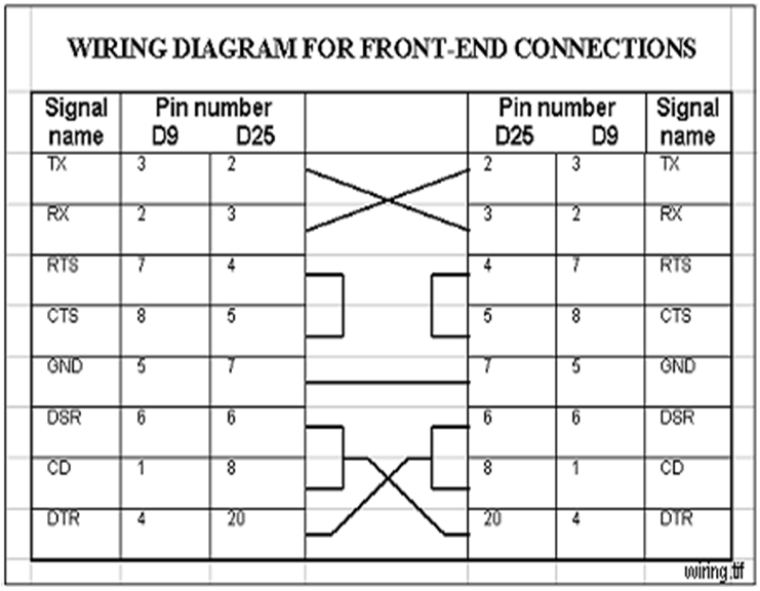


Figure 10: Serial cable wiring diagram

4.7 Communication adjustment guidelines

MicroSCADA has multiple adjustable attributes that affect the communication. The default values might not be suitable for each system and can sometimes cause communication problems. Here are listed the steps of what should be done when the communication is not working correctly.

4.7.1 Basic checks

In case of a communication does not start properly, following checks are worth to do before any further investigation:

- Check that the serial port defined with line attribute SD matches with the computes serial port. Check also that no error is reported to the notify window when the system or PC_NET is started.
- Check that the stations addresses match in master and slave.
- Check that the baud rates and other Data Transmission Attributes (in MicroSCADA System Configuration Tool) match in master and slave.
- Check that the master and slave stations Polling Addresses (PA) match
- Check that the stations Polling Address Length (PL), Length of Cause of Transmission Information (CL), Station Address Length (SL) and Information Address Length (IL) attribute values are correct

4.7.2 Message sending

If the remote end does not receive the messages sent from the line in question, following hints may help:

- Check if CTS errors exist. The DE attribute sets how long time the CTS signal is waited after the RTS signal has been raised. If the rising edge of the CTS does not occur at all (line disconnected) or if the DE attribute value is too small, timeout occurs, and the line enters to a CTS error state. The status point of the line is updated correspondingly
- Check the FIFO buffers of the used serial port. If RTS signal is used to control the data carrier, communication may function better if the FIFO buffers are disabled (or set to 1).
- Check the RTS Keepup Delay (RY) attribute value. If the RTS signal is used to control the data carrier and it is reset too early, the end of the message might be left out when the message is sent. With RY attribute the closing of the RTS signal can be delayed. On the other hand, if the RY delay is too long, the remote device may start transmission when the data carrier is not available. This causes an error.
- Check the COM Port Mode (CM) attribute. With the bit 3 setting the length of the message is used to calculate the RTS keep up time. See attribute documentation for more information.

4.7.3 Message reception

If the remote end send message but those are not received correctly by the line in question, following hints may help:

- Carrier Block (CB) attribute might be blocking the incoming messages. If the CB attribute is 0, carrier blocking is not used and messages are received regardless of the DCD state. If the CB attribute is 1, carrier blocking is used and the DCD must be signaled while characters belonging to the message is received.
- Check the TW attribute value. The TW attribute delays the sending of the message after the RTS and CTS signals are raised. If the beginning of a message is missing when it is received (recorded) in the other end, increasing the TW attribute value might help.
- Check that the Header Timeout (HT) and Response Timeout (TI) attributes lengths are long enough for the communication. The TI time could be too short in cases when the response is long and the baud rate small. This advice applies only in the balanced mode.
- If the messages are sent and received correctly but are not processed, check CRC or other errors from the Diagnostic Counters in System Configuration Tools online-mode.
- The usage of the protocol analyzer and the bit 1 or line attribute AU (Analyzer usage) will help in the tuning of the attributes DE, HT, TI, RI and RY. If bit 2 of the attribute AU is set, internal information related to completion of the write operation to the serial driver is display in the analyzer output.

4.7.4 Data updating

If the remote end sends messages and it seems like those are received correctly by the line in question, but the process objects are not updating, following hints may help:

- Check that the attached process objects addresses are correct.
- Check that the STA object is allocated to the MicroSCADA application in which the process objects are located. Related STA object attributes are AL (Allocation) and AS (Allocating application).
- If the base system attribute TN (Translated Number) or ND (Node number) for the STA object has been modified by the SCIL application, check that those refer to correct STA object in correct PC_NET node.

Section 5 Technical description

5.1 IEC 60870-5-101 Protocol

The IEC Technical Committee 57 (Working Group 03) has developed a protocol standard for telecontrol, teleprotection and associated telecommunications for electric power systems. The result of this work is IEC 60870-5. The first five documents listed in Section 2.4 specify the base of IEC 60870-5.

The IEC Technical Committee 57 has also generated a companion standard IEC 60870-5-101 for telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes. This standard utilizes the series of documents for IEC 60870-5.

IEC is designed according to the Enhanced Protocol Architecture (EPA) and it specifies the following Open Systems Interconnection (OSI) layers:

- Physical layer
- Data link layer
- Application layer

The physical layer can be any bit-serial physical layer, for example RS-232 C, RS-485, a fiber transceiver In SYS600, the communication takes place using the serial port(s) of the base system computer. The interface used is RS-232 C.

A lot of the functionality of the protocol is implemented in SCIL by using the application and system objects as presented in this document. The reason for this is flexibility and versatility. Although different IEC masters are implemented according to the same standard, functionality and requirements still vary from one system and application to another. When some of the central parts of message handling are implemented in SCIL, changes can easily be made, even online.

5.2 Level of implementation

In IEC 60870-5-101 the application level messages are called Application Service Data Units (ASDUs). Each ASDU consists of one or several information objects that contain the actual user data. SYS600 supports the ASDUs presented in Section 5.9 Private ASDUs, that is, the ones not included in the IEC 60870-5-101 companion standard, are indicated with an asterisk (*). The time stamped command ASDUs 58..64 used in IEC60870-5-104 protocol are mentioned in the following chapters but they are not a part of the IEC60870-5-101 companion standard. These ASDUs are supported but their usage is not recommended in IEC60870-5-101.

For more information, see Section 5.9.

5.3 Supported process object types

Since in the IEC 60870-5-101 Slave protocol input data is sent to the master by using SCIL statements, there is no strict relation between the SYS600 process object types and IEC ASDUs, but the relation shown in Table 1 can be made.

Table 1: Possible relations between the SYS600 process object types and IEC 60870-5-101 Slave ASDUs

Type id	Description	Process object type
1, 2, 30	Single point information	Binary input
3, 4, 31	Double point information	Double binary input
5, 6, 32	Step position information	Digital input, analog input
7, 8, 33	Bit string of 32 bit	Bit stream
9…14, 34, 35, 36	Measured value	Analog input
15, 16, 37	Integrated totals	Pulse counter
45, 58	Single command	Binary input
46, 59	Double command	Double binary input
47, 60	Regulating step commands	Double binary input
48…50, 61...63	Set point command	Analog input
51, 64	Bit string of 32 bit	Analog input

5.4 Communication

This section gives a more detailed description of the implementation of the IEC 60870-5-101 Slave protocol in SYS600, describing also the attributes that can be used for device communication. Examples of how to exchange data between the master and the slave are also given in this section along with information of the IEC 60870-5-101 Slave status codes.

5.4.1 Communication modes

The IEC 60870-5-101 protocol has two modes or link layer transmission procedures: unbalanced mode and balanced mode. In the unbalanced mode a master station controls the data traffic by polling the outstations sequentially. In this case the master is the primary station that initiates all the message transfer. The outstations are secondary stations (slaves) that can transmit only when they are polled.

In the balanced mode each station, master and slave, can initiate message transfers. The communication mode of a NET line can be selected by using the PO attribute when the line is created, and must be matched with the communication mode of the master station.

The serial bus topology (multi-drop) can be used only in the unbalanced mode.

5.4.2 Protocol converter

Each IEC 60870-5-101 Slave station configured on a line of a NET unit acts as a protocol converter between the IEC 60870-5-101 protocol and a base system. An internal protocol of SYS600 is used in communication between the SYS600 nodes, for example, between a base system and a NET unit.

Unlike in some other slave protocols, a IEC 60870-5-101 Slave station does not have a database in the NET unit, all data is sent directly to the master station. Therefore, no database initialization procedures are needed. If it is necessary, messages are stored temporarily into queues inside the NET unit.

In IEC 60870-5-101, data sent from the slave to the master can be assigned into two classes: class 1 or class 2. The data from the classes is sent to the master either by polling (unbalanced

mode) or spontaneously (balanced mode). In SYS600, the class assignment is done when the IEC message is written to the NET unit by using SCIL statements. In general, class 1 is used for high priority data and class 2 for lower priority data. The class is defined with the index value in the SD and EV attribute writes.

Both class 1 and class 2 queues allocate space for 200 IEC 60870-5-101 messages by default. With default settings, one queue may buffer about 8000 binary or double binary events (ASDU 2 or 4, M_SP_TA_1/M_DP_TA_1).

The buffering space can be limited or increased using station object attribute AQ ASDU Queue Length. With maximum AQ value of 3000, one queue may buffer 102000 binary or double binary events (ASDU 2 or 4, M_SP_TA_1/M_DP_TA_1).

Station object attribute RM, bit 10, can be used to define that the oldest events are deleted from the queue if an overflow situation occurs. When RM bit 10 is set and event cannot be placed to the defined queue anymore, the message from the top of the queue is removed and the new event is placed to the queue according to its priority. The removed message may contain multiple events, depending how it is packed.

5.4.3 Addressing

In IEC 60870-5-101, there are three kinds of addresses:

- **Link address**: The address of the IEC link. This address is defined by the PA (Polling Address) attribute of the IEC station. In most cases it is the same as the station address.
- **Station address**: A common address of an ASDU. There can be several common addresses of an ASDU with the same link address. This address is defined by the SA (Station Address) attribute of the IEC station.
- **Signal address**: An information object address. This address is unique for each signal with the same common address of an ASDU. The Information object address can be given in two ways:
    - As an unstructured address, which is basically just an integer within the range of the information object address.
    - As a structured address which is given byte-wise so that each byte usually represents a level in a hierarchical structure. For example, upper byte = unit number and lower byte = signal address.

SYS600 supports only unstructured addresses. However, this does not prevent communication with the IEC 60870 slaves using structured addresses, since the two types of addresses just demonstrate two different ways of presenting the same address. For example, a two-byte address can be represented as follows:

unstructured = 256*upper byte + lower byte

5.4.4 Data flow

Figure 11 describes the data flow between the process devices and the IEC master. Both directions are described separately since the data is handled in a different way depending on the direction.

See manual COM500*i* User's guide for detailed information about the data flow configuration.

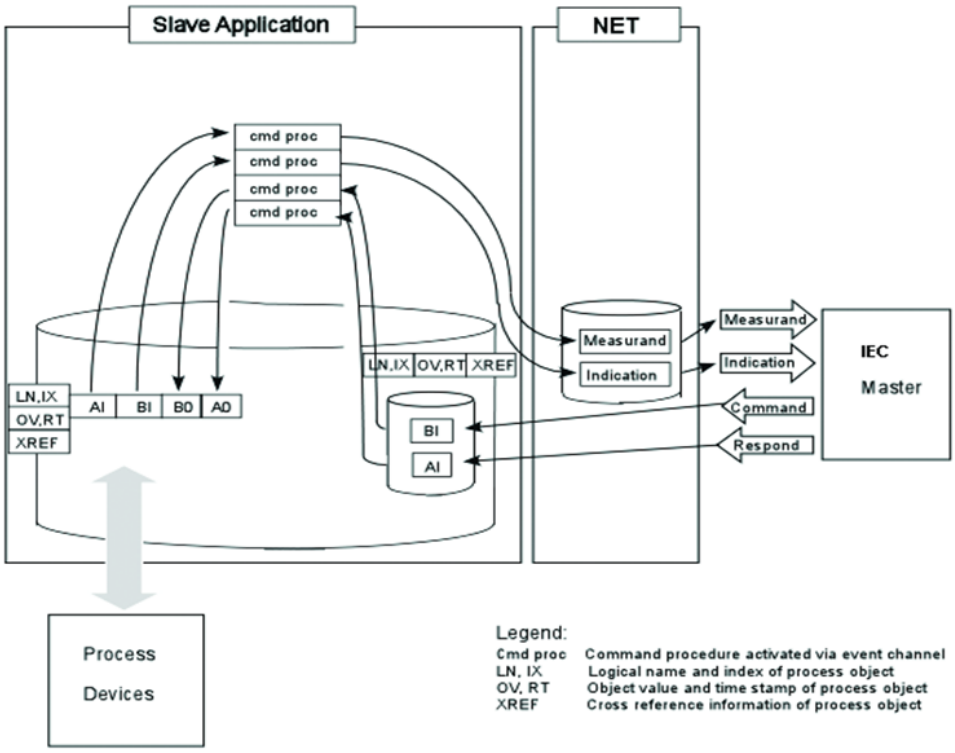


Figure 11: Data flow between the process devices and the IEC master in COM500i

5.4.4.1 Input data

When input data, for example indications and measured values, are sent from the process devices to the IEC master, the following steps are taken:

1. The process devices send data to the SYS600 process database.
2. The updated process object activates an event channel.
3. The event channel executes a command procedure. Some of the attributes of the process object are given as arguments to the command procedure.
4. The command procedure sends the data to the NET database by using the SD and EV attributes based on specific cross-reference information.
5. The NET unit sends data to the IEC master either as static or unsolicited data.

The cross-reference information is the data needed to deliver the signal to the IEC master, for example object address and message type.

The number event channels and command procedures needed to deliver data to the master depends on the application. One solution is to have one command procedure for each process object type. Examples of the command procedures are given later in this document.

In the quiescent operation mode static data is sent to the master as unsolicited messages only at start-up, and when the status of a data point changes. Besides these situations, data must be sent as change events or by means of polling.

5.4.4.2 Output data

When output data, for example object commands and analog setpoints, are sent from the IEC master to the process devices, the following steps are taken:

1. The command is received by the SYS600 process database. There must be a separate input process object for each IEC command address. This part of the process object is created manually when the signal engineering is done.
2. The updated process object activates an event channel.
3. The event channel executes a command procedure. Some of the attributes of the process object are given as arguments to the command procedure.
4. The command procedure sends the process devices by setting the corresponding output process object(s) and, if required, sends a confirmation to the IEC master via the NET unit.

Cross-reference data can also be used with commands. It can contain, for example, information about the logical names and indices of the output process objects. Examples of the command procedures are given later in this document.

5.4.4.3 Other messages

In addition to the input and output data described above, there are also some other messages transferred between the master and the slave. These messages contain, for example, synchronization commands and interrogation commands. Some of these commands are handled automatically by the NET unit while others require SCIL programming. The ones that require SCIL programming are described in detail later in this document.

System and application commands are sent to the process database as follows:

- System commands (for example test command) are sent to a bit stream process object.
- Application commands (for example general interrogation) are sent to an analog input (AI) process object.

Using these process objects the messages can be interpreted and processed by using SCIL. Unrecognized commands from the IEC master can be received by the bit stream process object. This feature can be configured using the RM attribute of the IEC slave station.

5.4.5 Redundancy

SYS600 supports redundant lines in both the balanced and unbalanced mode. The use of this feature requires that the corresponding functionality is also implemented in the IEC master, as stated in the document "Norwegian User Conventions for IEC 60870-5-101".

The master controls the mechanism that activates a line switch. The main line communication works as without redundancy. However, the master only sends a cyclical "request status of link" message on the back-up line, to which the slave responds with a "status of link" message. No other messages are allowed. When the master wants to switch lines, it sends a "reset of link" message to the redundant line, which the slave acknowledges. After this, the former main line acts as the former back-up line and vice versa. In the balanced mode, the links are initialized in both directions in the main line and also in the backup line. The main line operates normally and a cyclical "test function of link" message is transmitted on the backup line. The line switch occurs only when the master issues a command to the backup line. Bit 1 of the line attribute OM provides an optional functionality in the behavior of the backup line.

5.4.6 Device communication attributes

TD	Transparent Data
The TD attribute is used for sending transparent data (for example SPA messages) to the IEC master as a response to the Read command (ASDU 102).	
Data type:	Vector Value Vector (TYPE, ADDR, COT, TDT)
Value range:	0...255, for other parameters but ADDR

Table continues on next page

TD	Transparent Data
Index Range:	1...2, the number of the class the message is assigned to.
Access:	Write-only

Description of the vector parameters:

TYPE:	Type identification of the ASDU, integer. The type identifications shown in Table 2 are allowed when transparent data is sent to the IEC master/slave by using the TD attribute.

Table 2: The type identifications allowed when using the TD attribute

Type id	ASDU	Description
130	M_SB_NA_1	101 Encapsulated SPA bus reply message

ADDR:	Information object address, integer
Value range:	0...255, when IL attribute = 1 0...65535, when IL attribute = 2 0...16777215, when IL attribute = 3
COT:	Cause of transmission of the message, integer. Valid value: 7 = activation confirmation
TDT:	Transparent data (for example SPA message) as a text string

For more detailed information, see the examples and the interoperability list later in this document.

SD	Spontaneous Data

Enables SYS600 to send spontaneous or cyclic data to the IEC master. This data is sent without time stamp. By reading the SD attribute the user can enquire the number of data messages (ASDUs) in the queues not yet sent to the IEC master. When reading, the station attribute QI is recommended since it provides more specific information about the contents of the queues during runtime.

Data type:	Vector or integer
Value:	When writing: Vector (TYPE, ADDR, VAL, COT, QU, PRI, OW) When reading: Integer
Index range:	1...2, 101...102 When writing: The number of the class the message is assigned to. If value 101 is given, the item is added to class 1 queue but is not sent until an index value 1 or 2 is written. If value 102 is given, the item is added to class 2 queue but is not sent until an index value 1 or 2 is written. When reading: the index is meaningless, the returned value is the number of ASDUs not yet sent to the IEC master. If station attribute MT is bigger than 0, instead of reading of the SD-attribute, it is preferable to use the attribute QI. Indexes 101 and 102 are only useful in the balanced mode.
Access:	No limitations

Description of the vector parameters:

TYPE:	Type identification of the ASDU as a text string (for example "M_IT_NA_1")

The type identifications presented in Table 3 are allowed when data is sent to the IEC master by using the SD attribute.

Table 3: The type identifications allowed when using the SD attribute

Type id	ASDU	Description
1	M_SP_NA_1	Single-point information without time tag
3	M_DP_NA_1	Double-point information without time tag
5	M_ST_NA_1	Step position information
7	M_BO_NA_1	Bit string of 32 bit
9	M_ME_NA_1	Measured value, normalized value
11	M_ME_NB_1	Measured value, scaled value
13	M_ME_NC_1	Measured value, short floating point number
15	M_IT_NA_1	Integrated totals
70	M_EI_NA_1	End of initialization
110	P_ME_NA_1	Parameter of measured values, normalized value
111	P_ME_NB_1	Parameter of measured values, scaled value
112	P_ME_NC_1	Parameter of measured values, short floating point number

ADDR: Information object address

Value range: 0...255,
when IL attribute = 1
0...65535,
when IL attribute = 2
0...16777215,
when IL attribute = 3

VAL: The value of the object. The value range depends on the data object type. For more information about the SCIL programs, refer to Section 5.5.2.1.

COT: Cause of transmission of the ASDU. This parameter describes the reason why a message is sent. If the originator address is used, it can be set by adding 256*originator address to the COT value. The causes of transmission as shown in Table 4 are valid when using the SD attribute.

Table 4: The causes of transmission valid for the SD attribute

COT	Description
1	Periodic, cyclic
2	Background scan
3	Spontaneous
4	Initialized
5	Request or requested
11	Return information caused by a remote command
12	Return information caused by a local command

QU: Integer 1...255, bit mask of the qualifier descriptor bits of the ASDU. Qualifier descriptor bits indicate the current state of the data object. The bits in the qualifier byte are different for different ASDUs (see the examples later in this document). The relation between the qualifier descriptor bits and process object attributes is presented in Table 5.

Table 5: The relation between the qualifier descriptor bits and process object attributes

Bit	Name	Description	Attr.
IV	Invalid	A value is valid if it was correctly acquired. After the acquisition function recognizes abnormal conditions of the information source (missing or non-operating updating devices), the value is then marked invalid. The value of the information object is not defined under this condition. The mark Invalid is used to indicate to the destination that the value may be incorrect and cannot be used.	OS ==1
NT	Not topical	A value is topical if the most recent update was successful. It is not topical if it was not successfully updated during a specified time interval or it is unavailable.	OS ==2
SB	Substituted	The value of the information object is provided by input of an operator (dispatcher) or by an automatic source.	SB
BL	Blocked	The value of the information object is blocked for transmission; the value remains in the state that was acquired before it was blocked. Blocking and deblocking may be initiated, for example, by a local lock or a local automatic cause.	BL
CA	Counter adjusted	Counter was/was not adjusted since the last reading.	-
OV	Overflow	The value of the information object is beyond a predefined range of value (mainly applicable to analogue values).	OR
CY	Carry	Counter overflow occurred/did not occur in the corresponding integration period.	OF

PRI: Priority of the information object. This parameter defines how the information object sent using the SD attribute is placed in the class 1 and class 2 queues. This parameter can be used for setting the priority of data types or individual objects.

Value: Integer 0 … 3:
0 = The information object is always placed at the end of the queue.
1 or 2 = The information object is placed in the queue according to the priority value. Objects with higher priority pass the lower priority objects. If the queue contains more than one object with the same priority, the new object is placed behind the older one.
3 = The information object is always placed in the beginning of the queue.

OW: This parameter defines whether or not a new information object overwrites an older information object in the queue.

Value: Integer 0 or 1
0 = No overwriting
1 = A new information object overwrites an older object in the queue

EV	Event Data with Time Stamp

By writing data to the EV attribute the user can send event data, that is, time-tagged messages from SYS600 to the master. By reading the EV attribute the user can enquire the number of information objects (events) in the queue not yet sent to the IEC master. The maximum value of the EV attribute is limited by the RW attribute of the IEC slave station.

Data type:	Vector or integer
Value:	When writing: Vector (TYPE, ADDR, VAL, COT, QU, PRI, RT, RM, [CTRL]) When reading, Integer 0...65535
Index range:	1...2, 101...102 When writing: the number of the class the message is assigned to. If value 101 is given, the item is added to class 1 queue but is not sent until an index value 1 or 2 is written. If value 102 is given, the item is added to class 2 queue but is not sent until an index value 1 or 2 is written. When reading: the index is meaningless, returned value is the number of data items not yet sent to the IEC master. If station attribute MT is bigger than 0, instead of reading of the EV-attribute, it is preferable to use the attribute QI. Indexes 101 and 102 are useful only in the balanced mode.
Access:	No limitations

Description of the vector parameters:

TYPE: Type identification of the ASDU, either as a text string (for example "M_IT_NA_1") or as an integer

The type identifications presented in Table 6 are allowed when data is sent to the IEC master by using the EV attribute.

Table 6: The type identification allowed when using the EV attribute

Type id	ASDU	Description
2	M_SP_TA_1	Single-point information with time tag
4	M_DP_TA_1	Double-point information with time tag
6	M_ST_TA_1	Step position information with time tag
8	M_BO_TA_1	Bit string of 32 bit
10	M_ME_TA_1	Measured value, normalized value with time tag
12	M_ME_TB_1	Measured value, scaled value with time tag
14	M_ME_TC_1	Measured value, short floating point number with time tag
16	M_IT_TA_1	Integrated totals with time tag
30	M_SP_TB_1	Single-point information with time tag CP56Time2a
31	M_DP_TB_1	Double-point information with time tag CP56Time2a
32	M_ST_TB_1	Step position information with time tag CP56Time2a
33	M_BO_TB_1	Bitstring of 32 bit with time tag CP56Time2a
34	M_ME_TD_1	Measured value, normalized value with time tag CP56Time2a
36	M_ME_TF_1	Measured value, short floating point number with time tag CP56Time2a
37	M_IT_TB_1	Integrated totals with time tag CP56Time2a

ADDR:	Information object address
Value range:	0...255, when IL attribute = 1 0...65535, when IL attribute = 2 0...16777215, when IL attribute = 3 (default)
VAL:	The value of the object. The value range depends on the data object type. See the examples later in this document.
COT:	Cause of transmission of the ASDU. See the description of the SD attribute above.
QU:	Bit mask of the qualifier descriptor bits of the ASDU. See the description of the SD attribute above.
Value:	Integer 1...255
PRI:	Priority of the information object. See the description of the SD attribute above.
Value:	Integer 0...3
	If value = 255 is given, the event data is not recorded. However, the timestamp given with the RT and RM attributes is recorded as the timestamp of the event buffer overflow situation. A single indication message using this timestamp is transmitted automatically as the first event message when the communicating to the master proceeds. The information object address of the event is defined with the EA station attribute. If a percentage value defined with the attribute EA, index 2 is defined to be less than 100, the event is sent only if the current state of the queues is below or equal to the defined level.
RT:	Registration time. The registration time included in the time stamp of the unsolicited message.
RM:	Registration milliseconds. The millisecond part of the time stamp of the unsolicited message.
Value:	Integer 0...999
CTRL:	This is an optional field, which can be given to define the contents of the information object more specifically. The value is a bit pattern. Bits 1 and 2 control the "summer time" bit of the timestamp of the outgoing event messages. The term "summer time" is used in IEC standards as a synonym for "daylight saving time". Bits 2 and 3 define if the timestamp of the event is in UTC time or in local time. Values of the "summer time" bit are: 0 = standard time 1= summer time/daylight saving time. This bit exists only in ASDU types with the extension "with time tag CP56Time2a". In monitor direction, the corresponding ASDU numbers are 30..40. These ASDU numbers are usually used only with IEC60870-5-104 slave protocol and not with IEC60870-5-101 slave. Bit 0 = 0: The time invalid bit of the timestamp of this information object is not explicitly set, but it follows the value of the station attribute IV. Bit 0 = 1: The time invalid bit of the timestamp of this information object is set to state 1 = invalid. Bit 1 = 0: The summer time bit of the timestamp of this information object is not explicitly set, but it follows the value of the station attribute SU. Bit 1 = 1: The summer time bit of the timestamp of this information object is set to state 1 = summertime. Bit 2 = 0: No conversion from UTC (Coordinated Universal Time) to local time is requested. The given timestamp is sent as such and the state summer time bit follows the rules defined in the description of the bit 1 above. Bit 2 = 1: The timestamp is given as UTC and the conversion to the local time is requested. The timestamp of the event message is converted to the local time of the computer and the "summer time" bit of the timestamp is set according to the localized event timestamp. The setting of bit 2 overrides the summer time bit setting rules defined in the description of the bit 1 above. This bit should not be used together with bit 3 of this parameter. Bit 3 = 0: No conversion from local time to UTC (Coordinated Universal Time) is requested. The given timestamp is sent as such and the state summer time bit follows the rules defined in the description of the bit 1 above. Bit 3 = 1: The timestamp is given as local time and the conversion to the UTC time is requested. The timestamp of the event message is converted to the UTC time and the "summer time" bit of the timestamp is set to 0 (standard time). Setting of bit 3 overrides the summer time bit setting rule defined in the description of the bit 1 above. This bit should not be used together with bit 2 of this parameter.

CF	Command Confirmation

Manual confirmation of the received messages. In IEC 60870-5-101 commands received by the slave station are confirmed by using specific confirmation messages. They are basically original command messages with a different Cause Of Transmission (COT). Confirmation can be either positive or negative. The number of confirmations and the COT used depends on the type of the command. See the examples later in this document.
In SYS600 the CF attribute is used to confirm received messages manually. When the user writes only the COT value to the CF attribute, NET sends the confirmation message with written COT to the master. In this case the confirmation message is formed from the latest received command.
The user may also define which command is confirmed. NET automatically stores the incoming commands to an internal table (70 latest commands are stored). In case ADDR (or ADDR and TYPE) is given in CF-writing, the confirmed command is searched (latest first) from the table. If no command with matching address (or address and ASDU-type) is found, error ICCC_NO_ACTIVE_COMMAND is returned and the confirmation message is not sent. If a matching command is found, the given COT is set and the command is sent as a confirmation.

Data type: Vector
Value Vector (COT, [ADDR, [TYPE, [PRIO]]])

Access: Write-only

Description of the vector parameters:

COT: Cause of transmission of the confirmation message, integer. This parameter describes the reason why a message is sent. By adding 64 to the COT value, the confirmation is set negative, that is, the command is not accepted or failed. If the originator address is used, it can be set by adding 256*originator address to the COT value. The causes of transmission shown in Table 7 are valid when using the CF attribute.

Table 7: The causes of transmission valid for the CF attribute

COT	Description
7	Activation confirmation
9	Deactivation confirmation
10	Activation termination

ADDR: Information object address of the confirmed command, integer. For data commands, this is the address of the process object receiving the command. For system commands the address must be resolved from the command message and for application commands the address is 0.

Value range: 0...255, when IL attribute = 1
0...65535, when IL attribute = 2
0...16777215, when IL attribute = 3 (default)

TYPE: Type identification of the ASDU, integer. The CF attribute can also be used for sending transparent data as an IEC 60870-5-104 confirmation message. See the ASDUs and transport mechanisms later in this document.

PRIO: Priority of the confirmation message 0..4 (or 100..104 or 200..204 when the queue needs to be defined)

The priority of the confirmation message cannot be specified, if the transparent data is sent described below. For other ASDU types, the defining of the priority is possible and the used values are the same as in the SD and EV attributes. A bigger value means higher priority.

The messages are sorted in the transmission queue so that the higher priority messages are sent first. The priority is optional and if not specified, fixed values which depend on a given COT value are used. If the value of the PRIO is 100..104, the confirmation messages are always placed to the class 1 queue. Correspondingly, if the value of the PRIO is 200..204, the confirmation messages are always placed to the class 2 queue. When the queue is explicitly defined using PRIO values >= 100, the actual priority in the queue is PRIO modulo 100. With PRIO values 0..4 or when the PRIO is not defined, the queue selection is based on the value of the RM attribute, bit 2.

The type identifications presented in Table 8 are allowed commands ASDUs confirmed by the CF attribute.

Table 8: The type identifications allowed when using the CF attribute

Type id	ASDU	Description
45	C_SC_NA_1	Single command
46	C_DC_NA_1	Double command
47	C_RC_NA_1	Regulating step command
48	C_SE_NA_1	Set point command, normalized value
49	C_SE_NB_1	Set point command, scaled value
50	C_SE_NC_1	Set point command, short floating point number
51	C_BO_NA_1	Bit string of 32 bit
58	C_SC_TA_1	Single command with time tag CP56Time2a
59	C_DC_TA_1	Double command with time tag CP56Time2a
60	C_RC_TA_1	Regulating step command with time tag CP56Time2a
61	C_SE_TA_1	Set point command, normalized value with CP56Time2a
62	C_SE_TB_1	Set point command, scaled value withCP56Time2a
63	C_SE_TE_1	Set point command, short floating point number with CP56Time2a
64	C_BO_TA_1	Bit string of 32 bit with CP56Time2a
100	C_IC_NA_1	Interrogation command
101	C_CI_NA_1	Counter interrogation command
102	C_RD_NA_1	Read command
103	C_CS_NA_1	Clock synchronization command
104	C_TS_NA_1	Test command
105	C_RP_NA_1	Reset process command
107	C_TS_TA_1	Test command with CP56Time2a
110	P_ME_NA_1	Parameter of measured values, normalized value
111	P_ME_NB_1	Parameter of measured values, scaled value
112	P_ME_NC_1	Parameter of measured values, short floating point number

When sending transparent data, the syntax of the CF attribute is as follows:

vector (COT, [ADDR, [TYPE], [TDT]])

Description of the vector parameters:

COT: Cause of transmission of the confirmation message, integer. Valid value: 7 = activation confirmation.

ADDR: Information object address of the confirmed command, see above.

TYPE: Type identification of the ASDU, integer. The type identifications presented in Table 9 are allowed when transparent data is sent to the IEC master by using the CF attribute.

Table 9: The type identifications allowed when using the CF attribute

Type id	ASDU	Description
131	C_SR_NA_1	Parameter, byte string
133	C_SB_NA_1	101 Encapsulated SPA bus message

TDT: Transparent data (SPA message) as a text string. For more detailed information, see the examples and the interoperability list later in this document.

RS	Reset Event Queues

RS is used for clearing the content of event queues in the slave station. If an optional COT (Cause of Transmission) value is given, only the ASDUs with a matching COT are cleared from the specified queue. If the COT is not specified, the whole queue/s is/are cleared.

Data type: Integer or Integer's vector

Value: (VAL, [COT])
VAL= 1..3
1: reset the class 1 queue
2: reset the class 2 queue
3: reset both queues
COT= Integer 0..255 (optional)

If specified, only the ASDUs with the Cause of Transmission field matching with COT are cleared.

Access: Write-only

Example:

```
#SET STA1:SRS=(3,20) ; Clears the ASDUs with COT=20 (Interrogated) from
queues
; 1 and 2
```

5.5 Command procedures

5.5.1 Command procedures in COM500*i*

Signals are sent from the process units to an NCC and commands from an NCC to the process unit. COM500*i* reroutes the signals using command procedures and cross-references, see Figure 12. The command procedures, which are used for signal rerouting, are described here.

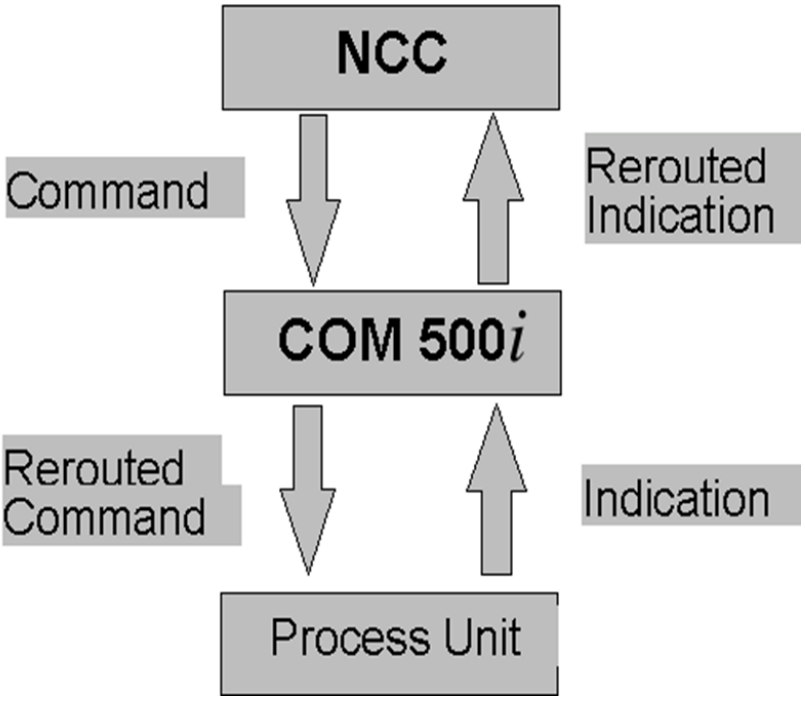


Figure 12: COM500i reroutes the signals

If the IEC 60870-5-101 Slave protocol is used with COM500*i*, the command procedures available for COM500*i* are used. The cross-reference information is entered in the Signal Cross-reference tool. For more information, see SYS600 COM500*i* User's Guide. Table 10 shows the used COM500*i* command procedures and event channels.

Table 10: Used event channels and command procedures

Process Object Type	Event Channel	Command Procedure
Analog Input	COM_USAI	COM_USAI
Single Indication	COM_USDI	COM_USDI
Double Indication	COM_USDB	COM_USDB
Digital Input	COM_USDI	COM_USDI
Pulse Counter	COM_USPC	COM_USPC
Binary Outputs	COM_DSBO	COM_DSBO
Analog Outputs	COM_DSAO	COM_DSAO
Digital Outputs	COM_DSDO	COM_DSDO

5.5.2 Command procedures in SYS600

5.5.2.1 Command procedures for process data

The command procedures presented in this chapter cannot cover all combinations and requirements, but they provide examples of the implementation.

The connection between the SYS600 process objects and messages to and from the IEC master is made by using cross-reference data. The cross-reference data is written to the FX (Free teXt) attribute of the process objects by using the Process Object Definition Tool.

The cross-references for data transfer from SYS600 to the IEC master are kept in the FX attribute, which is a string of max 30 characters (10 before revision 8.4.2). The general syntax for a cross-reference is:

```
<STA_NUMBER><TYPE><CLASS><ADDRESS><PRI><OW>
```

STA_NUMBER: Logical STA number (range 1 … 65535). The value is stored as 5 ASCII digits with leading spaces added if necessary. FX string positions 1...5 are used for this value. Actually the maximum value of the station number depends on the value of the SL attribute of the IEC station, but this example is according to the maximum range (SL=2).

TYPE: Type identification of the ASDU used when the value of the process object is sent to the IEC master. The value is stored (in the text format) as 9 ASCII digits with leading spaces added if necessary. Type identification uses the FX string positions 6...14.

CLASS: The class (1 or 2) the signal is assigned to. The value is stored as one ASCII digit. Static variation uses the FX string position 15.

ADDRESS: Information object address (range 1...16777215). The value is stored as 8 ASCII digits with leading spaces added if necessary. Address uses the FX string positions 16...23. Actually the maximum value of the information object address depends on the value of the IL attribute of the IEC station, but this example is according to the maximum range (IL=3).

PRI: The priority of the information object in the NET queues. The value is stored as one ASCII digit. Priority parameter uses the FX string position 24.

OW: The parameter that determines whether the information object overwrites an older one with the same address in the NET queues. Stored as one ASCII digit. The overwrite parameter uses the FX string position 25.

The used STA object attribute depends on whether the information object is sent with or without a time tag, that is, as static data or event. The qualifier byte is derived from the attributes of the process object.

An example of the cross-reference of a double indication signal assigned to class 1 and sent with time tag, overwrite and high priority.

FX					1	M	_	D	P	_	T	A	_	1	1			5	6	2	0	3	3	1					
CHAR	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0

Another example of the cross-reference of a scaled analog input signal assigned to class 2 and sent without time tag, no overwrite and low priority.

FX					1	M	_	M	E	_	N	B	_	1	2					7	7	1	9	0	0				
CHAR	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0

Analog inputs

In the IEC 60870-5-101 protocol analog values can be sent in three different ways:

- As 32-bit floating point values.
- As normalized values, that is, as 16-bit fixed point values with range –1...1-(2/32768).
- As scaled values, that is, as 16-bit integers with range –32768...32767.

When analog values are sent to NET by using the SD and EV attributes, both the normalized and scaled values are handled as signed 16-bit integers. This means that the maximum value of the analog signal is written as value 32767 and the minimum value is written as value –32768. The interpretation of the value is done by the IEC master based on the type identification of the ASDU.

If an analog signal is sent as a normalized or scaled value, it must be limited or scaled to the value range –32768...32767. This can happen, for example, by using a specific scale object and the SCIL SCALE function.

All the analog values are connected to the same event channel/command procedure combination. The command procedure is activated each time the process object is updated. It reads the updated value, and if necessary, limits or scales it and sends it to NET by using the cross-reference data. An example of the command procedure is listed below. A linear scale object AI_TO_S16 is used for the scaling of the analog values with the value range as presented in Table 11.

Table 11: The value range for analog values that are to be sent to NET

	Process	SYS600 Database
Low	-32768	-32767
High	32768	32767

This object does not scale the value but limits it to the correct range. If scaling is also wanted, it can be done by changing the "process" side value range.

```
#ERROR CONTINUE
;read cross-reference data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))
;scale the value if scaled or normalised value, and set the overfow bit
```

```
#CASE SUBSTR(%TYPE,7,1)
#WHEN "A","B" #BLOCK
#IF %AI < -32768 OR %AI > 32767 #THEN #BLOCK
@VALUE = ROUND(SCALE(%AI,"AI_TO_S16","INPUT"))
@OR = 1
#BLOCK_END
#ELSE #BLOCK
@VALUE = ROUND(%AI)
@OR = 0
#BLOCK_END
#BLOCK_END
#OTHERWISE #BLOCK
@VALUE = %AI
@OR = 0
#BLOCK_END
#CASE_END
;calculate the qualifier byte
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@QUALIFIER = %OR+16*%BL+32*%SB+64*%NT+128*%IV
;send data to NET with spontaneous COT
#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag
#SET
STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%VALUE,3,%QUALIFIER,%PRI,%RT,%RM)
#ELSE - ;send without time tag
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%VALUE,3,%QUALIFIER,%PRI,%OW)
```

Single indications

All the binary inputs are connected to an event channel/command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data. An example of the command procedure is listed below:

```
#ERROR CONTINUE

;read cross-reference data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;calculate the qualifier byte
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@QUAL= %BI+2+16*%BL+32*%SB+64*%NT+128*%IV

;send data to NET with spontaneous COT
#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag
#SET STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%BI,3,%QUAL,%PRI,%RT,%RM)
#ELSE - ;send without time tag
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%BI,3,%QUAL,%PRI,%OW)
```

Double indications

All the double binary inputs are connected to an event channel/command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using the cross-reference data. An example of the command procedure is listed below:

```
#ERROR CONTINUE
```

```
;read cross-reference data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;calculate the qualifier byte
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@DPIS = (0,2,1,3)
@DPI = %DPIS(%DB+1)
@QUAL = BIT(%DPI,0)+2*BIT(%DPI,1)+16*%BL+32*%SB+64*%NT+128*%

;send data to NET with spontaneous COT
#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag
#SET STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%DB,3,%QUAL,%PRI,%RT,%RM)
#ELSE - ;send without time tag
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%DB,3,%QUAL,%PRI,%OW)
```

Digital inputs

In IEC 60870-5-101 there is no direct equivalent of the digital input data type. Digital values can be sent as analog values, or as step position information, such as this information. Step position information is 8-bit data, where the 7 least significant bits are for the value and the most significant bit is set if the equipment is in transient state. A typical example of the use of this ASDU is the position of a tap changer. The command procedure for sending step position information is listed below.

```
#ERROR CONTINUE


;read cross-reference data

@FX = 'LN':PFX'IX'

@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))

@TYPE = SUBSTR(%FX,6,9)

@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))

@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))

@PRI = DEC_SCAN(SUBSTR(%FX,24,1))

@OW = DEC_SCAN(SUBSTR(%FX,25,1))


;calculate the qualifier byte

#IF %OS == 2 #THEN @NT = 1

#ELSE @NT = 0

#IF %OS == 1 OR %OS == 10 #THEN @IV = 1

#ELSE @IV = 0

#IF %OV < -64 OR %OV > 63 #THEN @OR = 1

#ELSE @OR = 0

@QUALIFIER = %OR+16*%BL+32*%SB+64*%NT+128*%IV
```

```
;send data to NET with spontaneous COT

#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag

#SET STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%BI,3,%QUAL,%PRI,%RT,%RM)

#ELSE - ;send without time tag

#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%BI,3,%QUAL,%PRI,%OW)
```

Pulse counters

Pulse counters are called integrated totals in the IEC 60870-5-101 protocol. All the pulse counters are connected to an event channel/command procedure combination that sends the process value to the NET unit. The command procedure is activated each time the process object is updated. It reads the updated value and sends it to NET by using cross-reference data.

The qualifier byte of the IEC 60870-5-101 integrated totals has a five-bit sequence number that is increased every time the information object is successfully sent to the IEC master. In this example the sequence number is stored in the RI attribute of the process object.

```
#ERROR CONTINUE

;read cross-reference data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;calculate the qualifier byte
@SQ = 'LN':PRI'IX' ;sequence number
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@CA = 0
@QUAL= %SQ+32*%OF+64*%CA+128*%IV

;reset status
@S = STATUS

;send data to NET with spontaneous COT
#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag
#SET STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%PC,3,%QUAL,%PRI,%RT,%RM)
#ELSE - ;send without time tag
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%PC,3,%QUAL,%PRI,%OW)

;increase/reset sequence number, if sending successful
#IF STATUS == 0 #THEN #BLOCK
#IF %SQ >= 31 #THEN @SQ = 0
#SET 'LN':PRI'IX' = %SQ+1
#BLOCK_END
```

Bit strings

The IEC 60870-5-101 protocol provides ASDUs 7 and 8 for transferring 32-bit strings. In the SYS600 implementation the bit string must be converted into an integer, when the string is sent to the IEC master by using the SD and EV attributes. This conversion is done by using the BIT_SET function as shown in the example below.

```
;read cross-reference data
@FX = 'LN':PFX'IX'
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
```

```
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;calculate the qualifier byte
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@OR = 0
@QUALIFIER = %OR+16*%BL+32*%SB+64*%NT+128*%IV

;convert the 32 least significant bits of the bit string to an integer
@LIMITED = SUBSTR(%BS,1,32)
@VALUE = 0
#LOOP_WITH I = 0..31
#IF BIT(%LIMITED,%I+1) == 1 #THEN @VALUE = BIT_SET(%VALUE,%I)
#LOOP_END

;send data to NET with spontaneous COT
#IF SUBSTR(%TYPE,6,1) == "T" #THEN - ;send with time tag
#SET STA'STA_NR':SEV(%CLASS)=(%TYPE,%ADDR,%VALUE,3,%QUAL,%PRI,%RT,%RM)
#ELSE - ;send without time tag
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%VALUE,3,%QUAL,%PRI,%OW)
```

5.5.2.2 Command handling in IEC 60870-5-101 protocol

Command confirmation
The IEC 60870-5-101 protocol includes the concept of command confirmations. Basically a confirmation is a message sent by the slave indicating that a command has been received, executed or rejected. Commands are confirmed in two steps as follows:

- A command is **confirmed** when it is received. An activation confirmation can be positive (command accepted) or negative (command rejected).
- A command is **terminated** when its execution is finished. An activation termination can be positive (command successfully completed) or negative (command failed).

The following exceptions apply:

- Select-type data commands, test command (ASDU 104) and reset process command (ASDU 105) are only confirmed, not terminated.
- Clock synchronization commands (ASDU 103) are confirmed automatically by PC-NET.

In the IEC 60870-5-101 Slave protocol commands are confirmed and terminated by using the CF (Command Confirmation) attribute of the IEC slave station.

If bit 8 of the RM-attribute is set, the incoming commands are checked and if the command contains an unsupported cause of transmission, information object address or the ASDU type itself is not supported, a negative confirmation with cause of transmission 44,45 or 47 is sent automatically and the process object in the database is not updated. In addition to this, if bit 3 (Process commands with invalid CAA) of the line attribute OM is set and no STA object with matching Common Address of ASDU is configured, a negative confirmation with cause of transmission 46 is sent. See the descriptions of the station attribute RM and line attribute OM.

The meaning of these cause of transmission values:

44 unknown type identification

45 unknown cause of transmission

46 unknown common address of asdu

47 unknown information objects address

Command transactions
In the SYS600 implementation of the IEC 60870-5-101 slave protocol 60 command transactions can be open at the same time. When a command is confirmed by using the CF attribute, a command with a matching ASDU number and address is searched from the buffer. If a match is found, a confirmation or termination message is sent to the master and if not, status 13860 = ICCC_NO_ACTIVE_COMMAND is returned.

Set the value of the PC attribute of each IEC slave station to 0 when confirming commands from SCIL. Otherwise commands may not be properly confirmed or terminated.

Process object receiving commands
As stated before, commands received from the IEC master are seen as input data by the slave and command messages can be received by IEC type process objects.

These process objects activate an event channel, which executes a command procedure. The following attribute values should be the same for all the process objects receiving a command from the master:

- The value of the AN (Action Enabled) attribute should be 1
- The value of the AA (Action Activation) attribute should be 2 (update)
- The value of the AF (Action at First Update) attribute should be 1

5.5.2.3 Command procedures for data commands

The commands sent from the IEC master to SYS600 are seen as input data and received by IEC type input process objects.

When this kind of input is updated, the value is read by a command procedure and if necessary converted, before it is written to the actual output objects that send the command to the process devices. The command procedure is activated through an event channel which is bound to the input object.

The number of the output process objects and the SCIL statements used for sending the commands to the process devices depend on the protocol of the device. Each controllable SPA object, for example the circuit breaker, has four output process objects, whereas each RTU or IEC object has only one.

It is also possible to utilize the input data to perform arbitrary internal operations in the application program. The procedures that are presented below cover the basic cases when IEC commands and setpoints can be mapped directly to the corresponding SYS600 outputs. If a more complex relationship between input and outputs is desired, it is necessary to build application specific command procedures.

Object commands
Object commands, such as switching device open/close commands or tap changer raise/lower commands, contain the ASDUs shown in Table 12 below. The table also includes the type of process object that can receive these commands.

Table 12: ASDUs of the object commands and process objects that receive these commands

Type id	ASDU	Description	Process Object Type
45	C_SC_NA_1	Single Command	Binary Input
46	C_DC_NA_1	Double Command	Double Binary Indication
47	C_RC_NA_1	Regulating Step Command	Double Binary Indication
Table continues on next page			

Type id	ASDU	Description	Process Object Type
58	C_SC_TA_1	Single command with time tag CP56Time2a	Binary Input
59	C_DC_TA_1	Double command with time tag CP56Time2a	Double Binary Indication
60	C_RC_TA_1	Regulating step command with time tag CP56Time2a	Double Binary Indication

The unit number (UN attribute) of the input process object must be the same as the STA object number of the corresponding IEC slave station, and the address of the process object must equal to the address of the command in the IEC master. Table 13 shows the updated attributes of the input process objects which receive commands.

Table 13: Updated attributes of the input process objects

Attribute	Values	Description
TY	45... 47, 58...60	Type identification of the ASDU
OV	0,1,2	Value of the command 0 = off, 1 = on (single command), 1 = off, 2 = on (double command), 1 = lower, 2 = higher (regulating step command)
RA	0, 1	Execution information of the command: 1 = select, 0 = execute
RB	0... 255	Qualifier of the command
CT	6, 8	Cause of transmission of the command
OG	0... 255	Originator address of the command

In the following examples, the process object receiving the command has the same logical name as the output objects. The station number of the IEC slave station is assumed to be 1. The first example is a single command for a SPA device with output process object indices as follows: open select 11, close select 12, execute 13 and cancel 13.

```
;send activation confirmation if execute command
#IF %RA == 0 #THEN #SET STA1:SCF = (256*%OG+%CT+1,'LN':POA'IX')

;reset status
@S = STATUS

;route the command to the output objects
#IF %CT == 8 #THEN #SET 'LN':POV14 = 1 ; deactivate COT (cancel)
#ELSE_IF %CT == 6 #THEN #BLOCK ; activate COT

;select
#IF %RA == 1 #THEN #BLOCK
#IF %OV == 0 #THEN #SET 'LN':POV11 = 1 ; open select
#ELSE #SET 'LN':POV12 = 1 ; close select
#BLOCK_END

;execute
#ELSE #SET 'LN':POV13 = 1
#BLOCK_END

;send activation confirmation (select) or termination (execute)
#IF STATUS == 0 #THEN @CONF_STATUS = 0 ; positive
#ELSE @CONF_STATUS = 64 ; negative
#IF %RA == 0 #THEN @COT = 10 ;termination
#ELSE @COT = %CT+1 ; confirmation
#SET STA1:SCF = (256*%OG+%CONF_STATUS+%COT,'LN':POA'IX')

The second example is a double command for a RTU device with one output
process object with index 13.
```

```
;send activation confirmation if execute command
#IF %RA == 0 #THEN #SET STA1:SCF = (256*%OG+%CT+1,'LN':POA'IX')

;reset status
@S = STATUS

;route the command to the output objects
#IF %CT == 8 #THEN #SET 'LN':POV14 = 1 ; deactivate COT (cancel)
#ELSE_IF %CT == 6 #THEN #BLOCK ; activate COT

;select
#IF %RA == 1 #THEN #BLOCK
#IF %OV == 0 #THEN #SET 'LN':POV11 = 1 ; open select
#ELSE #SET 'LN':POV12 = 1 ; close select
#BLOCK_END

;execute
#ELSE #SET 'LN':POV13 = 1
#BLOCK_END

;send activation confirmation (select) or termination (execute)
#IF STATUS == 0 #THEN @CONF_STATUS = 0 ; positive
#ELSE @CONF_STATUS = 64 ; negative
#IF %RA == 0 #THEN @COT = 10 ;termination
#ELSE @COT = %CT+1 ; confirmation
#SET STA1:SCF = (256*%OG+%CONF_STATUS+%COT,'LN':POA'IX')
```

Analog setpoints

Analog output block messages from the IEC master can be received by AI (Analog Input) process objects.

The unit number (UN attribute) of the input process object must be the same as the STA object number of the corresponding IEC slave station, and the address of the process object must be equal to the address of the command in the IEC master.

The attributes in the following Table 14 of the input process objects receiving the commands are updated.

Table 14: The updated attributes of the input process objects receiving setpoint commands

Attribute	Values	Description
TY	48...50, 61...63	Type identification of the ASDU
OV	Real	Value, range depends on the ASDU
RA	0, 1	Execution information of the command: 1 = select, 0 = execute
RB	0...255	Control status of the command

If the value of the setpoint command is a scaled or normalized value, its value range is –32768...32767. It may be necessary to scale the value to another range by using a procedure inverse to the one used for sending normalized and scaled analog values to the IEC master. The scale object used in the following example is the same as the one with analog inputs. The process device in this example does not support select-before-execute analog setpoint commands.

```
;send activation confirmation if execute command
#IF %RA == 0 #THEN #SET STA1:SCF = (256*%OG+%CT+1,'LN':POA'IX')

;reset status
@S = STATUS

;accept only activate/execute command
#IF %RA == 1 AND %CT == 6 #THEN #BLOCK
```

```
;scale if scaled/normalizrd value
#CASE %TY
#WHEN 48,49 @VALUE = ROUND(SCALE(%AI,"AI_TO_S16","OUTPUT"))
#OTHERWISE @VALUE = %AI
#CASE_END
#SET 'LN':POV12 = %VALUE
#BLOCK_END

;send activation confirmation (select) or termination (execute)
#IF STATUS == 0 #THEN @CONF_STATUS = 0 ; positive
#ELSE @CONF_STATUS = 64 ; negative
#IF %RA == 0 #THEN @COT = 10 ;termination
#SET STA1:SCF = (256*%OG+%CONF_STATUS+%COT,'LN':POA'IX')
#ELSE @COT = %CT+1 ; confirmation
```

Bit strings

In IEC protocol, it is possible to send 32-bit strings also from the master to the slave. ASDU 51 and 64 are reserved for this purpose. Bit strings can be received by analog input process objects as unsigned 32 bit integers. This analog value can be converted to a bit string by using SCIL statements.

5.5.2.4 Command procedures for application commands

Application commands contain the ASDUs shown in Table 15.

Table 15: The ASDUs that are included in the application commands

Type id	ASDU	Description
100	C_IC_NA_1	Interrogation command
101	C_CI_NA_1	Counter interrogation command
102	C_RD_NA_1	Read command
105	C_RP_NA_1	Reset process command

Application commands are received by an IEC type analog input with a unit number (UN attribute) equal to the STA object number of the IEC slave station. In addition to this, the object address (OA attribute) should be 0. When an application command is received, the attributes of the process object are updated as presented in Table 16.

Table 16: Updated attributes of the process object

Attribute	Values	Description
OV	100... 105	Type identification of the ASDU.
RA	0... 16777215	Information object address of the command. Depends on the value of the IL attribute.
RB	0... 255	Qualifier of the command.
CT	6, 8	Cause of transmission of the command.
OG	0... 255	Originator address of the command. Depends on the value of the CL attribute.

This process object should execute the event channel and command procedures, which recognize the incoming command, execute other command procedures depending on the command and confirm the commands. An example of this kind of command procedure is listed below. The sub-procedures are presented in the following sections.

```
#ERROR CONTINUE
;STA number and originator of the NCC
@STA_NR = 'LN':PUN'IX'
@ORIG = 'LN':POG'IX'
```

```
;send activation confirmation, if not read command
#IF %AI <> 102 #SET STA'STA_NR':SCF = (256*%ORIGINATOR+7,0,%AI)

;command specific part
#CASE %AI

;general interrogation
#WHEN 100 #EXEC GENINT:C (@STA_NUMBER = %STA_NR,@ORIGINATOR = %ORIG)

;counter interrogation
#WHEN 101 #EXEC CNTINT:C (@STA_NUMBER = %STA_NR,@ORIGINATOR = %ORIG)

;read
#WHEN 102 #EXEC RDDATA:C (@STA_NUMBER = %STA_NR,@READ_OBJECT_ADDRESS =
%RA)

;reset process
#WHEN 105 #EXEC RESPRC:C (@STA_NUMBER = %STA_NR,@ORIGINATOR = %ORIG)
#CASE_END
```

General interrogation command

When the IEC slave station receives a general interrogation command from the master, it must send all the input signals except pulse counters to the master without a time tag. The cause of transmission is set to 20. An example of the general interrogation command procedure GENINT is listed below. The procedure is executed by the application command handling procedure of the previous section.

```
#ERROR CONTINUE

;find all process objects with cross-reference data
#LOOP
@XREF_OBJECTS = APPLICATION_OBJECT_LIST(0,"IX","A","F","",-
"IU==1 AND SS>0 AND FX<>""""",("FX","OS","PT","OV","LN",-
"IX","BL","SB","OR"),10000)
#IF LENGTH(XREF_OBJECTS:VLN) > 0 #THEN #BLOCK

;initialize static data objects in NET
#LOOP_WITH SIGNAL = 1.. LENGTH(XREF_OBJECTS:VLN)

;cross-reference data
@FX = XREF_OBJECTS:VFX(%SIGNAL)
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;type id without time tag
#IF SUBSTR(%TYPE,6,1) == "T" #THEN -
@TYPE = SUBSTR(%TYPE,1,5) + "N" + SUBSTR(%TYPE,7,0)

; substitution value and flag for each data type
@OV = XREF_OBJECTS:VOV(%SIGNAL)
@OS = XREF_OBJECTS:VOS(%SIGNAL)
@OR = XREF_OBJECTS:VOR(%SIGNAL)
@BL = XREF_OBJECTS:VBL(%SIGNAL)
@SB = XREF_OBJECTS:VSB(%SIGNAL)
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
#CASE XREF_OBJECTS:VPT(%SIGNAL)
#WHEN 3 #BLOCK ; binary input
@SUBS_VALUE = 1
@QUAL = %OR+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
```

```
#WHEN 9 #BLOCK ;analog input
@SUBS_VALUE = 0
@QUAL= %OV+2+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
#WHEN 12 #BLOCK ;double binary input
@SUBS_VALUE = 1
@DPIS = (0,2,1,3)
@DPI = %DPIS(%OV+1)
@QUAL = BIT(%DPI,0)+2*BIT(%DPI,1)+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
#CASE_END

;if not sampled or error, give substitution value
#IF %OS >= 10 #THEN @VALUE = %SUBS_VALUE
#ELSE @VALUE = XREF_OBJECTS:VOV(%SIGNAL)
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%VALUE,20,%QUAL,%PRI,%OW)
#LOOP_END
#BLOCK_END
#IF NOT XREF_OBJECTS:VMORE #THEN #LOOP_EXIT
#LOOP_END

;send activation termination
#SET STA'STA_NUMBER':SCF = (256*%ORIGINATOR+9,0,100)
```

IEC 60870-5-101 also contains a group interrogation function. When this function is used, the signals to be sent to the IEC master are divided into groups which can be interrogated one at the time. The command procedure listed above can be modified to support group interrogations as follows:

- Interrogation group number (integer 1...16) is added to the cross-reference data stored in the FX attribute.
- The number of the interrogation group is read from the RB attribute of the AI process object that receives the command.
- The query condition of the command procedure is modified to search only the process objects with the specified group number. Note that group 20 stands for the general interrogation and in this case all signals must be sent.
- When executing the group interrogation commands, the cause of transmission of the signals must be set to 20+group number.

Counter interrogation command

The counter interrogation command is like a general interrogation but only for pulse counters. The counter interrogation procedure listed below can also be modified to support the group interrogation as described in the previous section. The only difference is that there are only four possible groups.

```
#ERROR CONTINUE

;find all pulse counter process objects with cross-reference data
#LOOP
@XREF_OBJECTS = APPLICATION_OBJECT_LIST(0,"IX","A","F","",-
"IU==1 AND SS>0 AND PT == 13 AND FX<>""""",("FX","OS","OV","LN",-
"IX","BL","SB","OF","RI"),10000)
#IF LENGTH(XREF_OBJECTS:VLN) > 0 #THEN #BLOCK

;initialize static data objects in NET
#LOOP_WITH SIGNAL = 1.. LENGTH(XREF_OBJECTS:VLN)

;cross-reference data
@FX = XREF_OBJECTS:VFX(%SIGNAL)
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))
```

```
;type id without time tag
#IF SUBSTR(%TYPE,6,1) == "T" #THEN -
@TYPE = SUBSTR(%TYPE,1,5) + "N" + SUBSTR(%TYPE,7,0)
;substitution value and qualifier
@OV = XREF_OBJECTS:VOV(%SIGNAL)
@OS = XREF_OBJECTS:VOS(%SIGNAL)
@OF = XREF_OBJECTS:VOF(%SIGNAL)
@SQ = XREF_OBJECTS:VRI(%SIGNAL)
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
@QUAL= %SQ+32*%OF+128*%IV
#IF %OS >= 10 #THEN @VALUE = 0
#ELSE @VALUE = XREF_OBJECTS:VOV(%SIGNAL)

;reset status
@S = STATUS

;send the value to NET
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%VALUE,37,%QUAL,%PRI,%OW)

;increase/reset sequence number, if sending successful
#IF STATUS == 0 #THEN #BLOCK
#IF %SQ >= 31 #THEN @SQ = 0
#SET 'LN':PRI'IX' = %SQ+1
#BLOCK_END
#LOOP_END
#BLOCK_END
#IF NOT XREF_OBJECTS:VMORE #THEN #LOOP_EXIT
#LOOP_END

;send activation termination
#SET STA'STA_NUMBER':SCF = (256*%ORIGINATOR+9,0,100)
```

Read command

By using the Read command, the user can request the value of an individual signal from the IEC master. The requested signal is sent with the cause of transmission value 5. Note that pulse counters are not included in the Read command.

```
#ERROR CONTINUE

;fx attribute to be searched
@READ_FX = "%%%%%%%%%%%%%%%" + DEC(%READ_OBJECT_ADDRESS,8) + "*"

;find the process objects with the given address
@READ_OBJECT = APPLICATION_OBJECT_LIST(0,"IX","A","F","",-
"(PT==3 OR PT==6 OR PT==9 OR PT==12) AND FX==%READ_FX",-
("FX","OS","PT","OV","LN","IX","BL","SB","OR","OF"),1)
#IF LENGTH(READ_OBJECT:VLN) > 0 #THEN #BLOCK

;cross-reference data
@FX = READ_OBJECT:VFX(1)
@STA_NR = DEC_SCAN(SUBSTR(%FX,1,5))
@TYPE = SUBSTR(%FX,6,9)
@CLASS = DEC_SCAN(SUBSTR(%FX,15,1))
@ADDR = DEC_SCAN(SUBSTR(%FX,16,8))
@PRI = DEC_SCAN(SUBSTR(%FX,24,1))
@OW = DEC_SCAN(SUBSTR(%FX,25,1))

;substitution value and flag for each data type
@OV = READ_OBJECT:VOV(1)
@OS = READ_OBJECT:VOS(1)
@OR = READ_OBJECT:VOR(1)
@BL = READ_OBJECT:VBL(1)
@SB = READ_OBJECT:VSB(1)
@OF = READ_OBJECT:VOF(1)
#IF %OS == 2 #THEN @NT = 1
#ELSE @NT = 0
```

```
#IF %OS == 1 OR %OS == 10 #THEN @IV = 1
#ELSE @IV = 0
#CASE READ_OBJECT:VPT(1)
#WHEN 3 #BLOCK ; binary input
@SUBS_VALUE = 1
@QUAL = %OR+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
#WHEN 6 #BLOCK ; digital input
@SUBS_VALUE = 0
#BLOCK_END
#WHEN 9 #BLOCK ;analog input
@SUBS_VALUE = 0
@QUAL= %OV+2+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
#WHEN 12 #BLOCK ;double binary input
@SUBS_VALUE = 1
@DPIS = (0,2,1,3)
@DPI = %DPIS(%OV+1)
@QUAL = BIT(%DPI,0)+2*BIT(%DPI,1)+16*%BL+32*%SB+64*%NT+128*%IV
#BLOCK_END
#CASE_END

;if not sampled or error, give substitution value
#IF %OS >= 10 #THEN @VALUE = %SUBS_VALUE
#ELSE @VALUE = READ_OBJECT:VOV(1)

;send the value to NET with requested COT
#SET STA'STA_NR':SSD(%CLASS)=(%TYPE,%ADDR,%VALUE,5,%QUAL,%PRI,%OW)
#BLOCK_END
```

Read commands (ASDU 102) are not confirmed by using the CF attribute.

Reset process command
Actions that should be taken with the reset process command may vary from one application to another. This command is problematic since, for example, the base system computer cannot be switched off by using SCIL. Some actions that can be taken with these commands are listed below.

- Restart PC-NET by setting LINn:BLT to "NONE" and after a pause back to "INTEGRATED". After this, the lines and stations configured to this NET unit must be created again.
- Reset the message queues in the NET unit by using the RS attribute of the IEC slave station.
- Send an interrogation command to all the process devices. This updates the process database, and thus the NET database, all the way from the process devices.

Reset process commands (ASDU 105) are confirmed by using the CF attribute.

5.5.2.5 Command procedures for system commands

System commands include the ASDUs presented in Table 17.

Table 17: ASDUs that are included in system commands

Type id	ASDU	Description
103	C_CS_NA_1	Clock synchronization command
104	C_TS_NA_1	Test command (to be used with IEC60870-5-101 only)
107	C_TS_TA_1	Test command with time tag CP56Time2a (to be used with IEC60870-5-104 only)

System commands are received by a bit stream process object with a unit number equal to the STA object number of the IEC slave station, and an address as defined by the CA attribute of the IEC slave station. NET sends the messages to this process object as transparent data and they must be interpreted by using SCIL statements.

Commands not recognized by NET are also sent to this process object, if this feature is enabled by the RM attribute of the IEC slave station.

Since synchronization is handled by the NET and the test command requires only a confirmation, the only action needed is the confirmation of the system commands.

This process object should execute the event channel and command procedures, which recognize the incoming command, interpret the address of the command, and confirm the commands. An example of this kind of command procedure is listed below:

```
;STA number and originator address of the NCC
@STA_NR = 'LN':PUN'IX'
@ORIGINATOR = 'LN':POG'IX'
;unpack the system command
@SYS_CMD = UNPACK_STR(%BS,8)

;type of command
@CMD_TYPE = %SYS_CMD(3)

;calculate the address of the command
@CMD_ADDR = 0
@STA_ADDR_END = 3 + STA'STA_NR':SPL + STA'STA_NR':SCL + STA'STA_NR':SSL
#LOOP_WITH I = 1..(STA'STA_NR':SIL)
@CMD_ADDR = %CMD_ADDR + (256**(%I-1))*(%SYS_CMD(%STA_ADDR_END+%I))
#LOOP_END

;send activation confirmation
#SET STA'STA_NR':SCF = (256*%ORIGINATOR+7,ROUND(%CMD_ADDR),%CMD_TYPE)
```

5.5.2.6 Command procedures for transparent data commands

It is possible to exchange transparent messages between a SYS600 IEC slave and an IEC master. SPA messages are an example of these kinds of messages. Transparent SPA messages are received by a bit stream process object and the corresponding reply is sent as a confirmation by using the CF attribute of the IEC slave station.

The example shown in Figure 13 has two SYS600 base systems, one as the network control system (IEC master) and one as the substation control system (IEC slave). The latter also acts as the master for several SPA devices. In this example, a transparent SPA command is sent from the IEC master to the SPA unit via the SYS600 IEC slave and the answer from the SPA unit is sent back to the IEC master. The following steps are taken according to Figure 13.

Step 1

: The SPA command "RF:" is sent from the IEC master to the IEC slave to address 12345 as an encapsulated SPA message (ASDU 133) by using the TD attribute as in the following:

```
#SET STA1:STD = (133,12345,6,”RF:”)
```

Step 2 and 3:

The message is received by the IEC slave in a bit stream process object with the UN attribute equal to the STA object number of the IEC slave station and the OA attribute value equal to the address of the command, which is 12345 in this case. Attached to this process object is an event channel, which activates a command procedure. The SPA message is parsed by the command procedure as in the following:

```
@SPA_MSG = TYPE_CAST(%BS,"TEXT")
```

The SPA message is sent to the SPA unit and the corresponding answer is read by using the SM attribute of the SPA station as in the following:

```
#SET STA2:SSM = %SPA_MSG
@SPA_ANSW = STA2:SSM
```

Step 4:

The answer is sent back to the IEC master as an activation confirmation of the command, that is, the encapsulated SPA reply message, as in the following:

```
@IEC_STA_NR = 'LN':PUN'IX'
@ORIG = 'LN':POG'IX'
@CMD_ADDR = 'LN':POA'IX'
@CMD_TYPE = 'LN':PTY'IX'
#SET STA'IEC_STA_NR':SCF = -
(256*%ORIG+7,%CMD_ADDR,%CMD_TYPE,%SPA_ASW)
```

The message is received by the IEC master in a bit stream process object with the UN attribute equal to the STA object number of the IEC slave station and the OA attribute equal to the address of the command. In this case, the message can also be interpreted by using the TYPE_CAST function to convert the message into text.

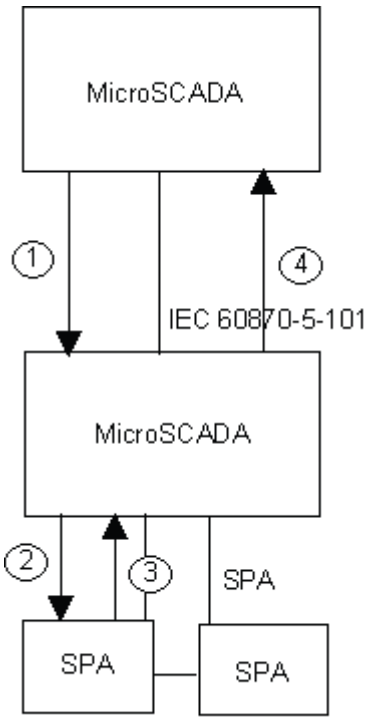


Figure 13: Flow of the transparent SPA messages

By using a mechanism like the one described above, it is possible to read and write the parameters of SPA units over an IEC 60870-5-101 line. The same kind of mechanism can also be used for other purposes, for example, for exchanging free-format text messages between the master and the slave.

5.5.2.7 Command procedures for file transfer

In COM500*i* command procedures COM_USFT and COM_IEFT handle the file transmission. Command procedures check cyclically, if the source directory contains files to be transmitted. A source directory is defined with the cross-reference tool. When requested, the directory listing of transferable files is sent to the IEC 60870-5-101 master protocol. File transfer commands contain the ASDUs shown in Table 18.

Table 18: ASDUs for File Transfer

Type id	ASDU	Description
120	F_FR_NA_1	File Ready
121	F_SR_NA_1	Section ready
122	F_SC_NA_1	Call directory, select file
123	F_LS_NA_1	Last section, last segment
124	F_AF_NA_1	Ack file, ack section
125	F_SG_NA_1	Segment
126	F_DR_TA_1	Directory

5.5.2.8 Command procedure for the end of initialization message

In the IEC 60870-5-101 protocol, the slave station sends a specific message, end of initialization (ASDU 70), after the slave station is initialized (that is, when it is ready for normal operation after start-up).

An example of a command procedure sends the end of initialization message to the IEC master. This command procedure can be executed from, for example, the predefined command procedure APL_INIT_1. The execution should be timed so that the message is sent after all the process objects are updated from the process devices.

```
;send end of initialization (zero address and value)

#SET STA'COM_IEC_STA_NUMBER':SSD(1) = ("M_EI_NA_1",0,0,4,0,0,0)
```

5.6 Command procedures for parameter in control direction

In the IEC 60870-5-101 protocol, it is possible for the master to set and activate the parameters of information objects of the IEC slave. Setting the limits of a measured value are an example of this kind of action. The following ASDUs are provided by the protocol.

Table 19: ASDUs for parameters in control direction

Type id	ASDU	Description
110	P_ME_NA_1	Parameter of measured values, normalized value
111	P_ME_NB_1	Parameter of measured values, scaled value
112	P_ME_NC_1	Parameter of measured values, short floating point number.
113	P_AC_NA_1	Parameter activation

In case RM bit 7 is set to 0, these ASDUs can be received by the same bit stream process object as system commands, and their handling can be included in the same command procedure that handles the system commands.

By using the UNPACK_STR function, the parameter ASDU can be converted into a vector whose elements are the octets of the message. The position and number of the elements depend on the address lengths, that is, the value of the PL, SL and IL attributes of the IEC slave station.

In case RM bit 7 is set to 1, the incoming ASDUs 110-112 are handled as set-point commands. In this case the updated process object is of type analog input and the OA of the process object is defined by the information object address field of the incoming message. With this configuration, the bitstream process object is not updated.

Table 20 presents the information elements of an ASDU 110 message as decimal numbers. The IEC slave station has the following address length attributes: PL=1, SL=2, IL=3, CL=2.

Table 20: The information elements of an ASDU 110 message

Octet	Value	Description
1	243	Control byte
2	12	Link address
3	110	Type id of ASDU
4	1	Variable structure qualifier (one object)
5	6	Cause of transmission (activation)
6	0	Originator address
7	12	First byte of common address of ASDU
8	0	Second byte of common address of ASDU
9	123	First byte of information object address
10	0	Second byte of information object address
11	0	Third byte of information object address
12	111	First byte of the value of the parameter
13	0	Second byte of the value of the parameter
14	2	Qualifier of the parameter
15	108	Checksum of the message

The actions taken depend on the value of some of the octets. The values listed in Table 21 are possible.

Table 21: The values of information elements

Octet	Values
Type id of ASDU	110…113, see Table 19
Cause of transmission	6 = activate, 8 = deactivate
Value	ASDU 110: Parameter as normalized value ASDU 111: Parameter as scaled value ASDU 112: Parameter as short floating point number ASDU 113: Not included
Qualifier	ASDUs 110…112: 1=threshold value, 2=smoothing factor, 3=low limit, 4=high limit ASDU 113: 1=act/deact previously loaded parameters, 2=act/deact of the addressed object, 3=act/deact cyclic transmission

5.7 Signal engineering

The term signal engineering here means the engineering needed to establish communication to an NCC using the IEC protocol. The signal engineering in COM500*i* is made using the signal X-reference tool after the system configuration is completed and the process database for the related to process devices is ready. The signal engineering process has been described in detail in COM500*i* Users Guide manual of the SYS600 product. The principal sequence for the signal engineering is:

1. Make a list of all signals that are to be transferred between the master and the slave. Include at least the ASDU type, class and address.
2. Make sure that the same communication mode (balanced or unbalanced) is used in the master and slave.
3. Determine the need for redundancy and update the system configuration correspondingly.
4. Define the NCCs, communication mode and the cross-reference data using the Signal X-references tool of COM500*i*.
5. Test each signal.

5.8 Status codes

The status codes for the IEC 60870-5-101 Slave protocol are defined in the SYS600 Status Codes manual. Some typical reasons for some of the status codes are also given.

Status codes are sent as system messages which can be received by analog input project objects with a unit number (UN) 0 and an object address (OA) as determined by the MI attribute of the line or station, or alternatively, they are returned as a response to a SCIL command accessing a IEC station object.

5.9 Interoperability

5.9.1 Interoperability

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the common address of ASDUs, represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction, allow the specification of the complete set or subsets, as appropriate for the applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment stemming

from different manufacturers, it is necessary for all the partners to agree on the selected parameters.

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

☐	Function or ASDU is not used
☒	Function or ASDU is used as standardized (default)
R	Function or ASDU is used in reverse mode
B	Function or ASDU is used in standard and reverse mode
A	Function or ASDU may need some additional application level work

The possible selection (blank, X, R, or B) is specified for each specific clause or parameter.

In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

5.9.1.1 System or device

(system-specific parameter)

☐	System definition
☐	Controlling station definition (Master)
☒	Controlled station definition (Slave)

5.9.1.2 Network configuration

(network-specific parameter)

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

5.9.1.3 Physical layer

(network-specific parameter)

Transmission speed (control direction)

Unbalanced interchange Circuit V.24/V.28 Standard		Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s		Balanced interchange Circuit X.24/X.27			
☐	100 bit/s	☒	2400 bit/s	☒	2400 bit/s	☐	56000 bit/s
☐	200 bit/s	☒	4800 bit/s	☒	4800 bit/s	☐	64000 bit/s
☒	300 bit/s	☒	9600 bit/s	☒	9600 bit/s		
☒	600 bit/s			☒	19200 bit/s		
☒	1200 bit/s			☒	38400 bit/s		

Transmission speed (monitor direction)

Unbalanced interchange Circuit V.24/V.28 Standard		Unbalanced interchange Circuit V.24/V.28 Recommended if >1 200 bit/s		Balanced interchange Circuit X.24/X.27			
☐	100 bit/s	☒	2400 bit/s	☒	2400 bit/s	☐	56000 bit/s
☐	200 bit/s	☒	4800 bit/s	☒	4800 bit/s	☐	64000 bit/s
☒	300 bit/s	☒	9600 bit/s	☒	9600 bit/s		
☒	600 bit/s			☒	19200 bit/s		
☒	1200 bit/s			☒	38400 bit/s		

5.9.1.4 Link layer

(network-specific parameter)

Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.

Link transmission produce

☒	Balanced transmission
☒	Unbalanced transmission

Frame length

255	Maximum length L

Address field of the link

☒	Not present (balanced transmission only)
☒	One octet
☒	Two octets
A	Structured
☒	Unstructured

When using an unbalanced link layer, the following ASDU types are returned in class 2 messages (low priority) with the indicated causes of transmission:

☐ The standard assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission
9, 11, 13, 21	<1>

☐ The standard assignment of ASDUs to class 2 messages is used as follows:

Type identification	Cause of transmission

5.9.1.5 Application layer

Transmission mode for application data
Mode 1 (least significant octet first), as defined in 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

Common address of ASDU
(system-specific parameter)

☒ One octet
☒ Two octets

Information object address
(system-specific parameter)

☒ One octet
☒ Two octets
☒ Three octets
A Structured
☒ Unstructured

Cause of transmission
(system-specific parameter)

☒ One octet
☒ Two octets
Set to zero in case of no originator address

Selection of standard ASDUs
Process information in monitor direction

(station-specific parameter)

☒	<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

Table continues on next page

A	<7> := Bitstring of 32 bit	M_BO_NA_1
A	<8> := Bitstring of 32 bit with time tag	M_BO_TA_1
X	<9> := Measured value, normalized value	M_ME_NA_1
X	<10> := Measured value, normalized value with time tag	M_ME_TA_1
X	<11> := Measured value, scaled value	M_ME_NB_1
X	<12> := Measured value, scaled value with time tag	M_ME_TB_1
X	<13> := Measured value, short floating point value	M_ME_NC_1
X	<14> := Measured value, short floating point value with time tag	M_ME_TC_1
X	<15> := Integrated totals	M_IT_NA_1
X	<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 quality descriptor	M_ME_ND_1
X	<30> := Single-point information with time tag CP56Time2a	M_SP_TB_1
X	<31> := Double-point information with time tag CP56Time2a	M_DP_TB_1
X	<32> := Step position information with time tag CP56Time2a	M_ST_TB_1
X	<33> := Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
X	<34> := Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1
X	<35> := Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
X	<36> := Measured value, short floating point value with time tag CP56Time2a	M_ME_TF_1
X	<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
A	<128> := Parameter byte string	M_SR_NA_1
A	<130> := 101 Encapsulated SPA bus reply message	M_SB_NA_1

Process information in control direction
(station-specific parameter)

X	<45> := Single command	C_SC_NA_1
X	<46> := Double command	C_DC_NA_1
X	<47> := Regulating step command	C_RC_NA_1
X	<48> := Set point command, normalized value	C_SE_NA_1
X	<49> := Set point command, scaled value	C_SE_NB_1
X	<50> := Set point command, short floating point value	C_SE_NC_1
A	<51> := Bitstring of 32 bit	C_BO_NA_1
A	<131> := Parameter byte string	C_SR_NA_1
A	<133> := 101 Encapsulated SPA bus message	C_SB_NA_1

System information in monitor direction
(station-specific parameter)

X	<70> := End of initialization	M_EI_NA_1

System information in control direction
(station-specific parameter)

X	<100>:= Interrogation command	C_IC_NA_1
X	<101>:= Counter interrogation command	C_CI_NA_1
X	<102>:= Read command	C_RD_NA_1
X	<103>:= Clock synchronization command	C_CS_NA_1
X	<104>:= Test command	C_TS_NA_1
X	<105>:= Reset process command	C_RP_NA_1
	<106>:= Delay acquisition command	C_CD_NA_1

Parameter in control direction
(station-specific parameter)

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

File transfer
(station-specific parameter)

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

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

Shaded boxes are not required.

Blank = function or ASDU is not used.

Mark type identification/cause of transmission combinations:

"**X**" if used only in the standard direction

"**R**" if used only in the reverse direction

"**B**" if used in both directions

Type identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		A	X		X						A	A		X					
<2>	M_SP_TA_1			X		A						A	A							
<3>	M_DP_NA_1		A	X		X						A	A		X					
<4>	M_DP_TA_1			X		A						A	A							
<5>	M_ST_NA_1		A	X		X						A	A		X					
<6>	M_ST_TA_1			X		A						A	A							
<7>	M_BO_NA_1		A	A		A									A					
<8>	M_BO_TA_1			A		A														
<9>	M_ME_NA_1	A	A	X		X									X					

Table continues on next page

Type identification		Cause of transmission																		
<10>	M_ME_TA_1			X		A														
<11>	M_ME_NB_1	A	A	X		X									X					
<12>	M_ME_TB_1			X		A														
<13>	M_ME_NC_1	A	A	X		X									X					
<14>	M_ME_TC_1			X		A														
<15>	M_IT_NA_1			X												X				
<16>	M_IT_TA_1			X												A				
<17>	M_EP_TA_1																			
<18>	M_EP_TB_1																			
<19>	M_EP_TC_1																			
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X		A						A	A							
<31>	M_DP_TB_1			X		A						A	A							
<32>	M_ST_TB_1			X		A						A	A							
<33>	M_BO_TB_1			X		A														
<34>	M_ME_TD_1			X		A														
<35>	M_ME_TE_1			X		A														
<36>	M_ME_TF_1			X		A														
<37>	M_IT_TB_1			X												A				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X

Table continues on next page

Type identification		Cause of transmission																		
<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						A	A			A						A	A	A	A
<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	X	X
<102>	C_RD_NA_1					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																			
<110>	P_ME_NA_1						X	X							A		X	X	X	X
<111>	P_ME_NB_1						X	X							A		X	X	X	X
<112>	P_ME_NC_1						X	X							A		X	X	X	X
<113>	P_AC_NA_1						A	A	A	A							A	A	A	A
<120>	F_FR_NA_1													B			X	X	X	X
<121>	F_SR_NA_1													B			X	X	X	X
<122>	F_SC_NA_1													B			X	X	X	X
<123>	F_LS_NA_1													B			X	X	X	X
<124>	F_AF_NA_1													B			X	X	X	X

Table continues on next page

Type identification		Cause of transmission																		
<125>	F_SG _NA_1													B			X	X	X	X
<126>	F_DR _TA_1 *			X		X														
	* Blank or X only.																			

5.9.1.6 Basic application functions

Station initialization
(station-specific parameter)

[X] Remote initialization

Cyclic data transmission
(station-specific parameter)

[ ] Cyclic data transmission

Read procedure
(station-specific parameter)

[X] Read procedure

Spontaneous transmission
(station-specific parameter)

[X] Spontaneous transmission

Double transmission of information objects with cause of transmission spontaneous
(station-specific parameter)

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 and M_PS_NA_1

[ ] Double-point information M_DP_NA_1, M_DP_TA_1 and M_DP_TB_1

[ ] Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1

[ ] Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1 (if defined for a specific project)

[ ] Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1

[ ] Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1

[ ] Measured value, short floating point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

Station interrogation
(station-specific parameter)

X	global				
A	group 1	A	group 7	A	group 13
A	group 2	A	group 8	A	group 14
A	group 3	A	group 9	A	group 15
A	group 4	A	group 10	A	group 16
A	group 5	A	group 11	Information object addresses assigned to each group must be shown in a separate table	
A	group 6	A	group 12		

Clock synchronization
(station-specific parameter)

X	Clock synchronization

Command transmission
(station-specific parameter)

X	Direct command transmission
X	Direct set-point command transmission
X	Select and execute command
X	Select and execute set-point command
X	C_SE ACTTERM used
X	No additional definition
X	Short-pulse duration (duration determined by a system parameter in the outstation) [1)]
X	Long-pulse duration (duration determined by a system parameter in the outstation) [1)]
X	Persistent output [1)]

1) The qualifier field of the command updates the RB attribute of the process object. This value is freely used by the MicroSCADA application.

Transmission of integrated totals
(station-specific parameter)

- [A] Mode A: local freeze with spontaneous transmission
- [A] Mode B: local freeze with counter interrogation
- [A] Mode C: freeze and transmit by counter interrogation commands
- [A] Mode D: freeze by counter interrogation command, frozen values reported spontaneously

- [X] Counter read
- [A] Counter freeze without reset
- [A] Counter freeze with reset
- [A] Counter reset

- [X] General request counter
- [A] Request counter group 1
- [A] Request counter group 2
- [A] Request counter group 3
- [A] Request counter group 4

Parameter loading
(station-specific parameter)

- [A] Threshold value
- [A] Smoothing factor
- [A] Low limit for transmission of measured value
- [A] High limit for transmission of measured value

Parameter activation
(station-specific parameter)

- [ ] Act/deact of persistent cyclic or periodic transmission of the addressed object

Test procedure
(station-specific parameter)

- [X] Test procedure

File transfer
(station-specific parameter)

File transfer in monitor direction

- [x] Transparent file
- [ ] Transmission of disturbance data of protection equipment
- [ ] Transmission of sequences of events
- [ ] Transmission of sequences of recorded analogue values

File transfer in control direction

- [x] Transparent file

Background scan
(station-specific parameter)

- [ ] Background scan

Acquisition of transmission delay
(station-specific parameter)

- [ ] Acquisition of transmission delay

Appendix A Examples of communication system configuration

The following SCIL procedures make the communication system configuration which is related to the base system configuration example presented earlier in this document. The first procedure creates an IEC 60870-5-101 Slave line and two stations on this line.

```
;*************************************************************************
; INPUT PARAMETERS
@NET = 3 ; NODE NUMBER OF THE PC-NET
@MODE = 0 ; BALANCED MODE
@LINE = 1 ; LINE NUMBER
@STATIONS = (1,2) ; SLAVE STATION NUMBERS
@APPLIC = 1 ; APPLICATION NUMBER
;*************************************************************************
; CREATE A IEC 60870-5-101 LINE TO NET
#IF NET'NET':SPO'LINE'==0 #THEN #BLOCK
#IF %MODE==0 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 30 ;balanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'LINE' = 12 ;link type
 #SET NET'NET':SPD'LINE' = 5000;polling detection (ms)
#BLOCK_END
#ELSE_IF %MODE==1 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 29 ;unbalanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'LINE' = 4 ;link type
 #SET NET'NET':SPD'LINE' = 30000 ;polling detection (ms)
#BLOCK_END
 #SET NET'NET':SSD'LINE' = "COM1" ;system device name
 #SET NET'NET':SMS'LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'LINE' = %LINE+(6000+(%NET*100)) ;message identifier
 #SET NET'NET':SBR'LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'LINE' = 2 ;parity
 #SET NET'NET':SRD'LINE' = 8 ;receive bit count
 #SET NET'NET':STD'LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'LINE' = 1 ;stop bit count
 #SET NET'NET':SPS'LINE' = 20 ;buffer pool size
 #SET NET'NET':SDE'LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'LINE' = 0 ;transmission wait delay (ms)
 #SET NET'NET':SHT'LINE' = 2000 ;header timeout (ms)
 #SET NET'NET':STI'LINE' = 2 ;timeout interval (s)
 #SET NET'NET':SRK'LINE' = 0 ;RTS keep up padding
;characters
 #SET NET'NET':SRI'LINE' = 0 ;receiver disabling time (ms)
 #SET NET'NET':SEN'LINE' = 3 ;enquiry limit
#BLOCK_END
;*************************************************************************
; CREATE IEC 60870-5-101 STATIONS TO NET
#LOOP_WITH I = 1..LENGTH(%STATIONS)
@STA=%STATIONS(%I)
 #SET NET'NET':SDV(29) = (%STA,%LINE) ;create station to line
 #SET STA'STA':SAL = 1 ;allocated
 #SET STA'STA':SAS = %APPLIC ;allocated system
 #SET STA'STA':SMI = 1000+%STA ;message identification
 #SET STA'STA':SMS = %APPLIC ;message application
 #SET STA'STA':SSE = 1 ;system messages enabled
 #SET STA'STA':SSA = %STA ;station address
 #SET STA'STA':SSL = 2 ;station address length (bytes)
 #SET STA'STA':SPA = %STA ;polling address link address)
 #SET STA'STA':SPL = 1 ;polling address length (bytes)
```

```
 #SET STA'STA':SIL = 3 ;info addr. length (bytes)
 #SET STA'STA':SCL = 1 ;COT length (bytes)
 #SET STA'STA':STC = 0 ;time synchronisation (0,1,2,3)
 #SET STA'STA':SCA = 32000 ;command address
 #SET STA'STA':SPC = 0 ;data conf. (0=man, 1=auto)
 #SET STA'STA':SMT = 15 ;manual conf waiting time (s)
 #SET STA'STA':SST = 5000 ;SYS waiting time (ms)
 #SET STA'STA':SSU = 0 ;summer time (0=no, 1=yes)
 #SET STA'STA':SRW = 10 ;reply window size
 #SET STA'STA':SML = 230 ;max. message length
 #SET STA'STA':SDR = 0 ;direction
 #SET STA'STA':SSR = 0 ;single char response
 #SET STA'STA':SRM = 0 ;running mode
 #SET STA'STA':SIU = 1 ;set station in use
#LOOP_END
; Set line in use
 #SET NET'NET':SIU'LINE' = 1
```

The second example is for configuring one IEC station with redundant lines:

```
@NET = 3 ; NODE NUMBER OF THE PC-NET
@MODE = 1 ; UNBALANCED MODE
@LINE = 2 ; LINE NUMBER
@RED_LINE= 7 ; REDUNDANT LINE
@STA = 2 ; SLAVE STATION NUMBER
@APPLIC = 1 ; APPLICATION NUMBER
;*************************************************************************
; CREATE THE IEC 60870-5-101 MAIN LINE TO NET
#IF NET'NET':SPO'LINE'==0 #THEN #BLOCK
#IF %MODE==0 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 30 ;balanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'LINE' = 12 ;link type
 #SET NET'NET':SPD'LINE' = 5000;polling detection (ms)
#BLOCK_END
#ELSE_IF %MODE==1 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 29 ;unbalanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'LINE' = 4 ;link type
 #SET NET'NET':SPD'LINE' = 30000 ;polling detection (ms)
#BLOCK_END
 #SET NET'NET':SSD'LINE' = "COM3" ;system device name
 #SET NET'NET':SMS'LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'LINE' = %LINE+(6000+(%NET*100));message identifier
 #SET NET'NET':SBR'LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'LINE' = 2 ;parity
 #SET NET'NET':SRD'LINE' = 8 ;receive bit count
 #SET NET'NET':STD'LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'LINE' = 1 ;stop bit count
 #SET NET'NET':SPS'LINE' = 20 ;buffer pool size
 #SET NET'NET':SDE'LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'LINE' = 0 ;transmission wait delay (ms)
 #SET NET'NET':SHT'LINE' = 2000 ;header timeout (ms)
 #SET NET'NET':STI'LINE' = 2 ;timeout interval (s)
 #SET NET'NET':SRK'LINE' = 0 ;RTS keep up padding characters
 #SET NET'NET':SRI'LINE' = 0 ;receiver disabling time (ms)
 #SET NET'NET':SEN'LINE' = 3 ;enquiry limit
 #SET NET'NET':SRU'LINE' = %STA ;redundant line station definition
#BLOCK_END
;*************************************************************************
; CREATE THE IEC 60870-5-101 BACK-UP LINE TO NET
#IF NET'NET':SPO'RED_LINE'==0 #THEN #BLOCK
#IF %MODE==0 #THEN #BLOCK
 #SET NET'NET':SPO'RED_LINE' = 30 ;balanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'RED_LINE' = 12 ;link type
 #SET NET'NET':SPD'RED_LINE' = 5000 ;polling detection (ms)
#BLOCK_END
```

```
#ELSE_IF %MODE==1 #THEN #BLOCK
 #SET NET'NET':SPO'RED_LINE' = 29 ;unbalanced IEC 60870-5-101
;slave
 #SET NET'NET':SLK'RED_LINE' = 4 ;link type
 #SET NET'NET':SPD'RED_LINE' = 30000 ;polling detection (ms)
#BLOCK_END
 #SET NET'NET':SSD'RED_LINE' = "COM8" ;system device name
 #SET NET'NET':SMS'RED_LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'RED_LINE' = %LINE+(6000+(%NET*100));message
;identifier
 #SET NET'NET':SBR'RED_LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'RED_LINE' = 2 ;parity
 #SET NET'NET':SRD'RED_LINE' = 8 ;receive bit count
 #SET NET'NET':STD'RED_LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'RED_LINE' = 1 ;stop bit count
 #SET NET'NET':SPS'RED_LINE' = 20 ;buffer pool size
 #SET NET'NET':SDE'RED_LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'RED_LINE' = 0 ;transmission wait delay (ms)
 #SET NET'NET':SHT'RED_LINE' = 2000 ;header timeout (ms)
 #SET NET'NET':STI'RED_LINE' = 2 ;timeout interval (s)
 #SET NET'NET':SRK'RED_LINE' = 0 ;RTS keep up padding
;characters
 #SET NET'NET':SRI'RED_LINE' = 0 ;receiver disabling time (ms)
 #SET NET'NET':SEN'RED_LINE' = 3 ;enquiry limit
 #SET NET'NET':SRU'RED_LINE' = %STA ;redundant line station
;definition
#BLOCK_END
;*********************************************************************
; CREATE IEC 60870-5-101 SLAVE STATION TO NET
 #SET NET'NET':SDV(29) = (%STA,%LINE) ;create station to line
 #SET STA'STA':SLI(2) = %RED_LINE ;redundant line number
 #SET STA'STA':SAL = 1 ;allocated
 #SET STA'STA':SAS = %APPLIC ;allocated system
 #SET STA'STA':SMI = 1000+%STA ;message identification
 #SET STA'STA':SMS = %APPLIC ;message application
 #SET STA'STA':SSE = 1 ;system messages enabled
 #SET STA'STA':SSA = %STA ;station address
 #SET STA'STA':SSL = 2 ;station address length (bytes)
 #SET STA'STA':SPA = %STA ;polling address link address)
 #SET STA'STA':SPL = 1 ;polling address length (bytes)
 #SET STA'STA':SIL = 3 ;info addr. length (bytes)
 #SET STA'STA':SCL = 2 ;COT length (bytes)
 #SET STA'STA':STC = 0 ;time synchronisation (0,1,2,3)
 #SET STA'STA':SCA = 32000 ;command address
 #SET STA'STA':SPC = 0 ;data conf. (0=man, 1=auto)
 #SET STA'STA':SMT = 15 ;manual conf waiting time (s)
 #SET STA'STA':SST = 5000 ;SYS waiting time (ms)
 #SET STA'STA':SSU = 0 ;summer time (0=no, 1=yes)
 #SET STA'STA':SRW = 10 ;reply window size
 #SET STA'STA':SML = 230 ;max. message length
 #SET STA'STA':SDR = 0 ;direction
 #SET STA'STA':SSR = 0 ;single char response
 #SET STA'STA':SRM = 0 ;running mode
 #SET STA'STA':SIU = 1 ;set station in use
; Set lines in use
 #SET NET'NET':SIU'LINE' = 1
 #SET NET'NET':SIU'RED_LINE' = 1
```

The third example is for an IEC 60870-5-101 slave line with dial-up and two IEC stations.

```
;*********************************************************************
; INPUT PARAMETERS
@NET = 3 ; NODE NUMBER OF THE PC-NET
@MODE = 1 ; 0 = BALANCED MODE, 1 = BALANCED MODE
@LINE = 2 ; LINE NUMBER
@STATIONS = (1,2) ; MASTER STATION NUMBERS
@APPLIC = 1 ; APPLICATION NUMBER
;*********************************************************************
```

```
; CREATE A IEC 60870-5-101 SLAVE DIAL-UP LINE TO NET
 #IF NET'NET':SPO'LINE'==0 #THEN #BLOCK
 #IF %MODE==0 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 32 ;balanced IEC 60870-5-101
;master
 #SET NET'NET':SLK'LINE' = 12 ;link type
 #SET NET'NET':SPD'LINE' = 5000;polling delay (ms)
 #SET NET'NET':SRI'LINE' = 0 ;receiver disabling time (ms)
#BLOCK_END
#ELSE_IF %MODE==1 #THEN #BLOCK
 #SET NET'NET':SPO'LINE' = 31 ;unbalanced IEC 60870-5-101
;master
 #SET NET'NET':SLK'LINE' = 1 ;link type
 #SET NET'NET':SPD'LINE' = 500 ;polling delay (ms)
 #SET NET'NET':SRI'LINE' = 5 ;receiver disabling time (ms)
 #SET NET'NET':SPL'LINE' = 10 ;polling limit
#BLOCK_END
 #SET NET'NET':SPP'LINE' = 10 ;polling period(ms)
 #SET NET'NET':SRP'LINE' = 1 ;reply polling
 #SET NET'NET':SMS'LINE' = %APPLIC ;message application
 #SET NET'NET':SMI'LINE' = %LINE+(6000+(%NET*100));message identifier
 #SET NET'NET':SBR'LINE' = 9600 ;baud rate
 #SET NET'NET':SPY'LINE' = 2 ;parity
 #SET NET'NET':SRD'LINE' = 8 ;receive bit count
 #SET NET'NET':STD'LINE' = 8 ;transmit bit count
 #SET NET'NET':SSB'LINE' = 1 ;stop bit count
 #SET NET'NET':SPS'LINE' = 20 ;buffer pool size
 #SET NET'NET':SDE'LINE' = 50 ;CTS delay (ms)
 #SET NET'NET':STW'LINE' = 0 ;transmission wait delay (ms)
 #SET NET'NET':SHT'LINE' = 50 ;header timeout (ms)
 #SET NET'NET':STI'LINE' = 50 ;timeout interval (ms)
 #SET NET'NET':SRK'LINE' = 0 ;RTS keep up padding characters
 #SET NET'NET':SEN'LINE' = 3 ;enquiry limit
 #SET NET'NET':SAC'LINE' = 1 ;autocaller enabled
 #SET NET'NET':SIU'LINE' = 1 ;set line in use, initialize modem
#BLOCK_END
;*************** Configure Autocaller *********************************
 #SET NET'NET':SIU'LINE' = 0
 #SET NET'NET':SCL'LINE' = 0 ;connection time limited
 #SET NET'NET':SCT'LINE' = 0 ;connection time
 #SET NET'NET':SDD'LINE' = 0 ;radio disc. delay
 #SET NET'NET':SPU'LINE' = 0 ;pulse dialing
 #SET NET'NET':SRC'LINE' = 0 ;remote calls enabled
 #SET NET'NET':SRW'LINE' = 0 ;radio connecton wait time
 #SET NET'NET':SIU'LINE' = 1
;*********************************************************************
; CREATE IEC 60870-5-101 SLAVE STATIONS TO NET
#LOOP_WITH I = 1..LENGTH(%STATIONS)
@STA=%STATIONS(%I)
 #SET NET'NET':SDV(29) = (%STA,%LINE) ;create station to line
 #SET STA'STA':SAL = 1 ;allocated
 #SET STA'STA':SAS = %APPLIC ;allocating application
 #SET STA'STA':SMI = 1000+%STA ;message identification
 #SET STA'STA':SMS = %APPLIC ;message application
 #SET STA'STA':SSE = 1 ;system messages enabled
 #SET STA'STA':SSA = %STA ;station address
 #SET STA'STA':SSL = 2 ;station address length (bytes)
 #SET STA'STA':SPA = %STA ;polling address link address)
 #SET STA'STA':SPL = 1 ;polling address length (bytes)
 #SET STA'STA':SIL = 3 ;info addr. length (bytes)
 #SET STA'STA':SCL = 1 ;COT length (bytes)
 #SET STA'STA':SCA = 32000 ;command address
 #SET STA'STA':SST = 5000 ;SYS waiting time (ms)
 #SET STA'STA':SRT = 10 ;application reply timeout (s)
 #SET STA'STA':SCT = 60 ;application term. timeout (s)
 #SET STA'STA':SSU = 0 ;summer time (0=no, 1=yes)
 #SET STA'STA':SRW = 10 ;reply window size
 #SET STA'STA':SML = 230 ;max. message length
```

```
 #SET STA'STA':SDR = 0 ;direction
 #SET STA'STA':SSR = 0 ;single char response
 #SET STA'STA':SRM = 0 ;running mode
 #SET STA'STA':SIU = 1 ;set station in use
#LOOP_END
;dial number 123456789
 #SET NET'NET':SCN'LINE' = "123456789" ;connection
```

Index

Hitachi ABB Power Grids
Grid Automation Products
PL 688
65101 Vaasa, Finland

https://hitachiabb-powergrids.com/microscadax

Scan this QR code to visit our website

1MRK 511 490-UEN