

Control FPWIN Pro7

## M\_IEC60870\_PRO7 Programming Library

Reference Manual



## Copyright and liability

---

Copyright and liability information relating to this documentation.

Copyright © 2025 Panasonic Industry Europe GmbH

Caroline-Herschel-Strasse 100, 85521 Ottobrunn, Germany

Last modified on: 2025-02-02

This documentation and everything described in it are copyrighted. You may not copy this documentation, in whole or part, without written consent of Panasonic Industry Europe GmbH.

Panasonic Industry Europe pursues a policy of continuous improvement of the design and performance of its products. Therefore we reserve the right to change the documentation/product without notice. In no event will Panasonic Industry Europe be liable for direct, special, incidental, or consequential damage resulting from any defect in the product or its documentation, even if advised of the possibility of such damages.

For technical support, please contact the [Panasonic hotline](#).

# Table of contents

<b>1 General information.....</b>	<b>6</b>
1.1 Hardware and software requirements.....	6
1.2 Recommended PLC types.....	6
1.3 Version history.....	7
1.4 Global variables and useful constants.....	10
1.5 Advanced system description information.....	12
1.6 System registers and serial ports.....	13
1.7 Glossary of terms.....	16
1.8 Frequently Asked Questions (FAQs).....	18
<b>2 PLC examples and basic programs.....</b>	<b>21</b>
2.1 Example for standard stations.....	21
2.2 Example for stations without class 1/class 2 buffer management.....	22
<b>3 Debugging and monitoring.....</b>	<b>23</b>
3.1 Detecting process commands with <b>IEC_Debug</b> .....	23
3.2 Monitoring with <b>IEC_Bits</b> .....	24
3.3 Checking the buffer in the monitor direction.....	26
3.4 Checking the PLC's real-time clock.....	28
<b>4 Function blocks of this library.....</b>	<b>29</b>
4.1 IEC 60870 main function blocks.....	29
4.1.1 <b>IEC60870</b> .....	29
4.1.2 <b>IEC60870_ASDU</b> .....	30
4.1.3 <b>IEC60870_EM1</b> .....	31
4.2 Process information in monitor direction.....	33
4.2.1 <b>IEC_M_BO_NA_1</b> .....	33
4.2.2 <b>IEC_M_BO_TA_1</b> .....	34
4.2.3 <b>IEC_M_BO_TB_1</b> .....	36
4.2.4 <b>IEC_M_BO_TB_EM1</b> .....	37
4.2.5 <b>IEC_M_DP_NA_1</b> .....	39
4.2.6 <b>IEC_M_DP_NA_2</b> .....	41
4.2.7 <b>IEC_M_DP_NA_EM1</b> .....	42
4.2.8 <b>IEC_M_DP_TA_1</b> .....	44
4.2.9 <b>IEC_M_DP_TB_1</b> .....	46

4.2.10 IEC_M_DP_TB_1EM1.....	47
4.2.11 IEC_M_DP_TB_2.....	49
4.2.12 IEC_M_IT_NA_1.....	51
4.2.13 IEC_M_IT_NA_1EM1.....	52
4.2.14 IEC_M_IT_NA_2.....	54
4.2.15 IEC_M_IT_NA_2EM1.....	55
4.2.16 IEC_M_IT_TA_1.....	57
4.2.17 IEC_M_IT_TA_2.....	59
4.2.18 IEC_M_IT_TB_1.....	60
4.2.19 IEC_M_IT_TB_1EM1.....	62
4.2.20 IEC_M_IT_TB_2.....	63
4.2.21 IEC_M_IT_TB_2EM1.....	65
4.2.22 IEC_M_ME_NA_1.....	67
4.2.23 IEC_M_ME_NA_EM1.....	68
4.2.24 IEC_M_ME_NB_1.....	70
4.2.25 IEC_M_ME_NB_EM1.....	72
4.2.26 IEC_M_ME_NC_1.....	74
4.2.27 IEC_M_ME_NC_EM1.....	75
4.2.28 IEC_M_ME_TA_1.....	77
4.2.29 IEC_M_ME_TB_1.....	79
4.2.30 IEC_M_ME_TC_1.....	81
4.2.31 IEC_M_ME_TD_1.....	83
4.2.32 IEC_M_ME_TD_EM1.....	84
4.2.33 IEC_M_ME_TE_1.....	86
4.2.34 IEC_M_ME_TE_EM1.....	88
4.2.35 IEC_M_ME_TF_1.....	90
4.2.36 IEC_M_ME_TF_EM1.....	92
4.2.37 IEC_M_SP_NA_1.....	93
4.2.38 IEC_M_SP_NA_EM1.....	94
4.2.39 IEC_M_SP_TA_1.....	96
4.2.40 IEC_M_SP_TB_1.....	97
4.2.41 IEC_M_SP_TB_EM1.....	98
4.3 Process information in control direction.....	99
4.3.1 IEC_C_BO_NA_1.....	99
4.3.2 IEC_C_BO_TA_1.....	100

4.3.3 IEC_C_DC_NA_1.....	102
4.3.4 IEC_C_DC_NA_2.....	103
4.3.5 IEC_C_DC_TA_1.....	105
4.3.6 IEC_C_DC_TA_2.....	107
4.3.7 IEC_C_SC_NA_1.....	108
4.3.8 IEC_C_SC_NA_2.....	110
4.3.9 IEC_C_SC_TA_1.....	111
4.3.10 IEC_C_SC_TA_2.....	113
4.3.11 IEC_C_SE_NA_1.....	114
4.3.12 IEC_C_SE_NB_1.....	115
4.3.13 IEC_C_SE_NC_1.....	116
4.3.14 IEC_C_SE_TA_1.....	117
4.3.15 IEC_C_SE_TB_1.....	119
4.3.16 IEC_C_SE_TC_1.....	120
4.4 Other useful function blocks.....	121
4.4.1 IEC_FP0_A80.....	121
4.4.2 IEC_FPX_ReadAD80.....	123
4.4.3 IEC60870_M_Diag.....	124
4.4.4 IEC60870_RTC_DST.....	126
4.4.5 IEC60870_RTC_NoDST.....	128
4.4.6 IEC60870_Simulation.....	130
<b>5 Panasonic hotline.....</b>	<b>131</b>

# 1 General information

---

The library can be used for IEC 60870-5 remote control using the FP-I4C or FPWEB2 unit.

Supported PLC types: FP0H, FP7, FP-XH, FP0R (tested with T32CP), FP2SH, FP7, FPΣ, FP-X (tested with C30 and C14)


Read this information carefully to avoid malfunctions.

## 1.1 Hardware and software requirements

---

The programming library is designed for FP series PLCs and dedicated communication units.

- The library has been developed for the programming software Control FPWIN Pro7 and can be used with our FPWEB2 and FP-I4C communication units.

Control FPWIN Pro7 is the programming software for all FP series PLCs of [Panasonic Industry](#) . It enables programming according to the International Electronic Commission standard IEC 61131-3.

The user library **M\_IEC60870R** can be used with versions 5.31 to 6.14.

The user library **M\_IEC60870\_PRO7** can be used for version 7.11 or higher.

- These PLC types have been tested with the library: FP0H, FP0R-T32CP, FP2SH, FP7, FPΣ, FP-X C30, FP-X C14, FP-XH.
- IEC 60870-5-101 serial communication is only supported by FPWEB2 units. IEC 60870-5-104 is supported by FPWEB2 and FP-I4C units. Please refer to the interoperability lists installed with the library for details about the supported features.

## 1.2 Recommended PLC types

---

Select a PLC type that is suitable for your application.

1. Ethernet, long-term storage and class 1/class 2 data separation are important.  
Basic PLC memory: FP0H (set to 65k DT) or FP7 (set to 999k DT)  
Memory used for IEC 60870 buffers: 262k words, 29k blocks  
Examples included in Control FPWIN Pro7 project file.  
Recommended PLC type: FP0H or FP7
2. Ethernet line with redundant communication to the main station. PLC used as a gateway to other substations, e.g. via GPRS or Modbus.  
Basic PLC memory: FP7 set to 144k or 999k DT, depending on the size of the PLC program  
Memory used for IEC 60870 buffers: 16k words, 2 x 300 blocks, free memory used for additional functions

Examples available on request.

Recommended PLC type: FP7

## 1.3 Version history

---

This documentation covers version 3.2.2.0.

### Version 3.2.2.0 2024-04

New functions:


- Cause of transmission P/N for spontaneous data telegrams
- New function blocks **IEC\_C\_SC\_NA\_2**, **IEC\_C\_SC\_TA\_2**, **IEC\_C\_DC\_NA\_2**, **IEC\_C\_DC\_TA\_2** with separate short and long pulse parameters.
- General interrogation with full time stamp (via **IEC\_Bits.GA\_Timestamp**). Implemented only for the long time stamp format.

Improvements and bug fixes:

- **IEC\_C\_xC\_xA\_2**: New variant with separate short and long pulse duration settings.
- **M\_XRB...**: Changed from INT to UINT (and in some internal cases to DINT) to support DT addresses up to DT65533.
- **Elem\_ofArray2D** needs DINT for large arrays.
- **EM1\_BUFFERSIZE** increased from 9 to 10 to be able to handle REAL values with a valid local time stamp. This may affect the deletion of non-transmitted data from existing systems.

Existing applications should update their **IEC870\_MonitorEM1**, **IEC870\_MonitorEM2** ... buffers. The right index size of the two dimensional array must be changed from 0..8 to 0..9, e.g. `ARRAY [0..500,0..8] OF INT` → `ARRAY [0..500,0..9] OF INT`. These arrays are declared outside of the library.

Other improvements:

- Re-designed documentation from CHM to HTML format. The new help is available in the InfoHub at <https://infohub.industry.panasonic.eu/>  and as a PDF file which is included in the ZIP file.
- New examples
- All function blocks **IEC\_C\_xxx\_xx\_EM\_1** have been removed. If one of these function blocks was used in an existing application, it should be replaced by the corresponding function blocks with the same name without "EM\_1" at the end.
- The library is available as a ZIP file without setup. Add the contents to a project folder of your choice.

### Version 3.1 2016-05

New functions:

- IEC60870\_ASDU for Gateway applications.

Improvements and bug fixes:

- Improved RBM buffer organization (initializing, start delay).
- EM1 buffer handling for redundancy connections. Buffers could drop data even if the corresponding RBM is not connected.

### **Version 3.0 2014-07**

New functions:

- The library is ready to run with FP7 and FPWIN Pro 7.
- Change of library name into M\_IEC60870\_PRO7.sul
- The library supports up to 4 redundancy buffers with IP lock to bind the main station IP address to a specified buffer.
- The buffer can be split to share class1 and class2 data. The FP7 can use with this buffer organization all its DT registers above the address 32767.

Improvements and bug fixes:

- The main concept is optimized and the number of function blocks (for different PLC types) is reduced. This helps to minimize the engineering if a PLC type is changed for your future projects.
- The HTML pages and the standard samples are updated for using the FP7.

### **Version 2.6 2014-02**

New functions:

- The library is ready to run with up to four connections. This function needs the FP Web Configurator in version V2.6 or newer.

Improvements and bug fixes:

- The HTML pages do not need array elements in the csv export file. This helps to minimize the update process of the FPWEB unit.
- The HTML pages and the standard samples are updated for using the FP7.

### **Version 2.42 2013-12**

New functions:

- New function blocks for different daylight saving rules.

Improvements and bug fixes:

- The invalid bit for monitoring data can be modified by the user program.
- Updated examples with ready-to-run IEC60870 RTU solutions.
- Real time clock handling in IEC60870\_RTU
- RTU user mapping of monitoring data is using the REAL format



- Small changed in the RTU HTML configuration pages

## Version 2.4 2012-04

New functions:

- New solution IEC60870 RTU: no programming tool is necessary. All project files including the PLC program can be uploaded to the FPWEB/PLC unit via the FP Web Configurator Tool. The RTU can be configure via HTML pages.
- New examples with ready-to-run IEC60870 RTU solutions.
- New examples for redundancy connections.
- PLC and FP Web-Server sources of the new IEC60870 RTU solution are available to customize the unit.
- Code optimization for FPWIN Pro 6.3 with conditional compiling.
- Improved performance in all function blocks.
- New functions IEC\_M\_xx\_NA\_LOG (using the log buffer management for IEC60870 functions, which does not contain time stamp information).
- New combi function blocks used in the IEC60870 RTU.

## Version 2.2 2010-06

New functions:

- New license concept: all FPWEB2 units can be used without hardware-related licenses; the license is related to the FP Web Configuration Tool V2.2 or higher
- New time stamp concept with high accuracy (ms)
- Code optimization for FPWIN Pro 6.1
- Improved performance of all function blocks
- Communication redundancy mechanism (communicating with two supervisory control systems)
- New function block: **IEC60870\_RTC**
- Improved function blocks: **IEC\_M\_DP\_NA\_1**, **IEC\_M\_DP\_TB\_1**, **IEC\_M\_DP\_TB\_1EM1**
- Completion of all types in monitoring direction for the different communication modes
- Viewing STARTDT state of the IEC60870 as flag in the PLC.
- IEC60870 system variable names changed, e.g. IEC870\_RBM --> IEC870\_RBM1, IEC870\_Monitor --> IEC870\_Monitor1

## Version 1.2 2005-08

New functions:

- New function block **IEC\_M\_BO\_TB\_LOG**
- Improved performance in all function blocks

- Code optimization for FPWIN Pro 5.0, service release 2
- Communication redundancy mechanism (communicating with two supervisory control systems)
- Disabling of Cycle\_max in the measurement function blocks possible
- For the commands in control direction, the telegram Activation on Termination has been added in accordance with the international standard IEC60870.

## 1.4 Global variables and useful constants

The global variable lists in the sample programs supplied with the libraries contain data buffers, data arrays, and system flags.

Comments for IEC 60870 system information start with a number to make sorting easier.

Global variables of the sample programs:

	Class	Identifier	FP address	IEC address	Type	Initial	A...	Comment
1	VAR_GLOBAL_CONSTANT	IEC60870_REDUNDANCY			INT	1	<input type="checkbox"/>	1000 IEC60870 Buffer definition: Set 0 if no redundancy is used
2	VAR_GLOBAL_RETAIN	IEC870_RBM1	DT8000	%MW5.8000	IEC60870_RBuffer		<input type="checkbox"/>	1000 IEC60870 Header for monitoring buffer
3	VAR_GLOBAL_RETAIN	IEC870_RBM2	DT10600	%MW5.10600	IEC60870_RBuffer		<input type="checkbox"/>	1000 IEC60870 Header for monitoring buffer (used for redun...
4	VAR_GLOBAL_RETAIN	IEC870_RBM1_EM1			IEC60870_R_EM1		<input type="checkbox"/>	1000 IEC60870 Header for monitoring buffer (used for redun...
5	VAR_GLOBAL_RETAIN	IEC870_RBM2_EM1			IEC60870_R_EM1		<input type="checkbox"/>	1000 IEC60870 Header for monitoring buffer (used for redun...
6	VAR_GLOBAL	IEC870_Control1	DT250	%MW5.250	ARRAY [1..1,0..25] O...	[26(0)]	<input type="checkbox"/>	1002 IEC60870 Control buffer for incoming telegrams
7	VAR_GLOBAL	IEC870_Control2			ARRAY [1..1,0..25] O...	[26(0)]	<input type="checkbox"/>	1002 IEC60870 Control buffer for incoming telegrams (used f...
8	VAR_GLOBAL_RETAIN	IEC870_Monitor1	DT8026	%MW5.8026	ARRAY [0..98,0..25] ...		<input type="checkbox"/>	1001 IEC60870 Monitoring buffer for send telegrams
9	VAR_GLOBAL_RETAIN	IEC870_Monitor2	DT10626	%MW5.10626	ARRAY [0..98,0..25] ...		<input type="checkbox"/>	1001 IEC60870 Monitoring buffer for send telegrams (used for r...
10	VAR_GLOBAL_RETAIN	IEC870_MonitorEM1			ARRAY [0..11000,0.....	[110010(0)]	<input type="checkbox"/>	1001 IEC60870 Extended monitoring buffer for send telegrams ...
11	VAR_GLOBAL_RETAIN	IEC870_MonitorEM2			ARRAY [0..11000,0.....	[110010(0)]	<input type="checkbox"/>	1001 IEC60870 Extended monitoring buffer for send telegrams ...
12	VAR_GLOBAL_RETAIN	IEC870_FCounter	DDT5000	%MD5.5000	ARRAY [1..8] OF DINT		<input type="checkbox"/>	3000 IEC60870 Integrated totals (counter) e.g. for high speed c...
13	VAR_GLOBAL_RETAIN	IEC870_SP_Buffer	DT6000	%MW5.6000	ARRAY [1..64,1..3] O...		<input type="checkbox"/>	4000 IEC60870 Database for IEC60870-Elements (System)
14	VAR_GLOBAL_RETAIN	IEC870_ME_Buffer	DDT6500	%MD5.6500	ARRAY [1..48,1..4] O...		<input type="checkbox"/>	4000 IEC60870 Database for IEC60870-Elements (System)
15	VAR_GLOBAL_RETAIN	IEC870_IT_Buffer	DDT7000	%MD5.7000	ARRAY [1..8,1..3] OF...		<input type="checkbox"/>	4000 IEC60870 Database for IEC60870-Elements (System)
16	VAR_GLOBAL						<input type="checkbox"/>	
17	VAR_GLOBAL_RETAIN	g_iErrorAddress	DT31500	%MW5.31500	INT		<input type="checkbox"/>	9990 Source address of the visualized error
18	VAR_GLOBAL_RETAIN	g_iError	DT31501	%MW5.31501	INT		<input type="checkbox"/>	9991 Paging of errors: 1 = newest entry
19	VAR_GLOBAL_RETAIN	g_sErrorCode	DT31502	%MW5.31502	STRING[16]		<input type="checkbox"/>	9992 Visualized message text of a history buffer log
20	VAR_GLOBAL_RETAIN	g_sErrorTimestamp	DT31512	%MW5.31512	STRING[18]		<input type="checkbox"/>	9993 Visualized timestamp of the message text of a history buf...
21	VAR_GLOBAL_RETAIN	g_dutIEC60870_RTUSet...	DT31527	%MW5.31527	IEC60870_UnixTime...		<input type="checkbox"/>	9993 IEC60870 Set time
22	VAR_GLOBAL_RETAIN	g_awLogError	DT31530	%MW5.31530	ARRAY [0..99,0..4] O...		<input type="checkbox"/>	9995 Error logger archive
23	VAR_GLOBAL	g_bLogErrorUp	R0	%MX0.0.0	BOOL	FALSE	<input type="checkbox"/>	9995 Page up: Get previous (older) error
24	VAR_GLOBAL	g_bLogErrorDown	R1	%MX0.0.1	BOOL	FALSE	<input type="checkbox"/>	9996 Page down: Get next (newer) error
25	VAR_GLOBAL_RETAIN	g_dutDiagnosis	WR210	%MW0.210	dut26WORD		<input type="checkbox"/>	9998 FP Web Server Diagnosis status.htm

The addresses of the buffers can be different for each RTU type. To ensure correct operation, synchronize the buffer addresses in the PLC and the IEC 60870 communication unit.

### NOTICE

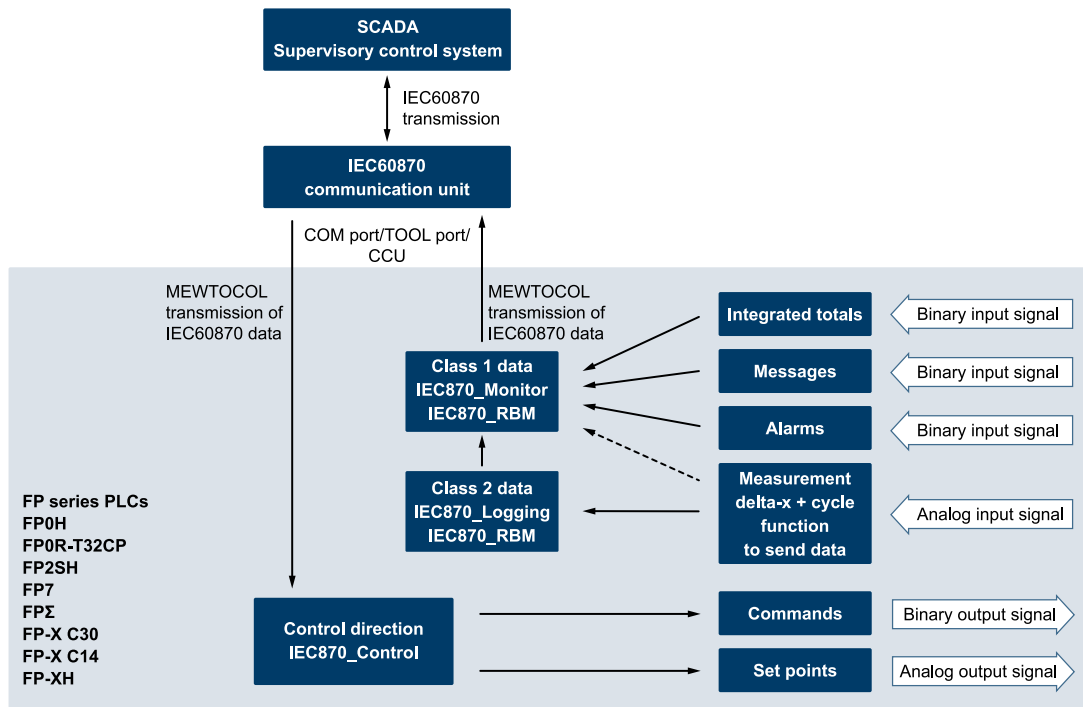
Please note that changing any other variable can lead to serious malfunctions. If you need assistance with assigning individual addresses, please contact our hotline.

Several useful variables, many of which can be used for conditional compilation, are included in the M\_IEC60870\_PRO7 Library or in the sample programs. The constant values are used as inputs for the IEC 60870 main function blocks.

Variable	Comment
<b>VERSION_IEC60870</b>	Returns the date and version number of the library specified in the comment of the variable declaration. (This is a global library variable.)
<b>IEC60870_REDUNDANCY</b>	<p>The library supports up to 4 communication lines (IEC 60870-5-104 only). This constant optimizes program size and runtime behavior.</p> <p>The value of the constant defines the main system function of the buffer:</p> <ul style="list-style-type: none"> <li>• 0: Used for non-redundant systems, modem configurations, or the FPG-EM1 unit (uses RBM1, RBM2_EM1 or RBM2).</li> <li>• 1: Used for standard redundancy with two lines (uses RBM1, RBM2).</li> <li>• 2: Used for communication with three lines (uses RBM1, RBM2, RBM3).</li> <li>• 3: Used for communication with four lines (uses RBM1, RBM2, RBM3, RBM4).</li> </ul> <p>If the main function block <b>IEC60870_EM1</b> is used, the additional buffers <b>IEC870_RBM1_EM1</b>, <b>IEC870_RBM2_EM1</b>, <b>IEC870_RBM3_EM1</b>, and <b>IEC870_RBM4_EM1</b> are used for the large class 2 buffers.</p>
<b>M_PLC_FP0</b> <b>M_PLC_FP0H</b> <b>M_PLC_FP2</b> <b>M_PLC_FP_SIGMA</b> <b>M_PLC_FP10SH</b> <b>M_PLC_FP7</b>	Constant variable to calculate normalized measurement values for each PLC type.
<b>M_PLC_NO_NORM</b>	Use this variable if measurement values are not normalized.
<b>M_PLC_FP0_0_20mA</b> <b>M_PLC_FP0H_0_20mA</b> <b>M_PLC_FPX_0_20mA</b> <b>M_PLC_FP_SIGMA_0_20mA</b> <b>M_PLC_FP2_0_20mA</b> <b>M_PLC_FP7_0_20mA</b>	<p>Constant variable to calculate normalized measurement values for each PLC type.</p> <p>Use these constants for sensors with a measurement range of 0 to 20mA instead of 4 to 20mA.</p>

## 1.5 Advanced system description information

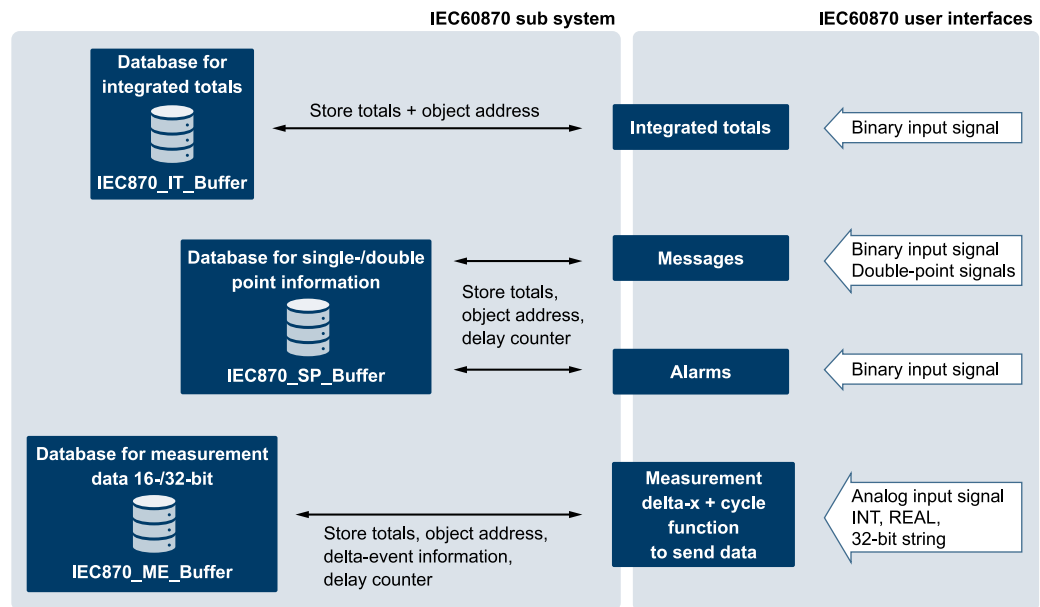
The function blocks for IEC 60870 communication are linked together in a complex structure that does not need to be understood in detail. For advanced programmers, we have provided block diagrams of the data flow and storage mechanism.



This diagram shows the data flow between the data point (object) via the IEC 60870 communication unit and the supervisory control system.

If an event is logged, the information will be stored in the buffer for the monitor direction until the communication unit reads it and sends it to the supervisory control system. If the communication unit is not running or communication is disconnected, all data will be held in the monitoring buffer and the logging buffer. These buffers are declared as `VAR_GLOBAL_RETAIN` in the global variable list. Even if the program is updated, no data will be lost.

In the control direction, the communication unit provides set points and commands to the user interfaces, i.e. the library's function blocks.



This diagram shows the links between the user interfaces (e.g. **IEC\_M\_SP\_NA\_1**) and the databases which contain temporary information on each object.

Each information point requires temporary information. All these signals are stored in the three different databases: **IEC870\_IT\_Buffer**, **IEC870\_SP\_Buffer**, and **IEC870\_ME\_Buffer**. The databases handle the temporary signals with multiple instances of the function blocks.

Related topics

[Debugging and monitoring](#) (page 23)

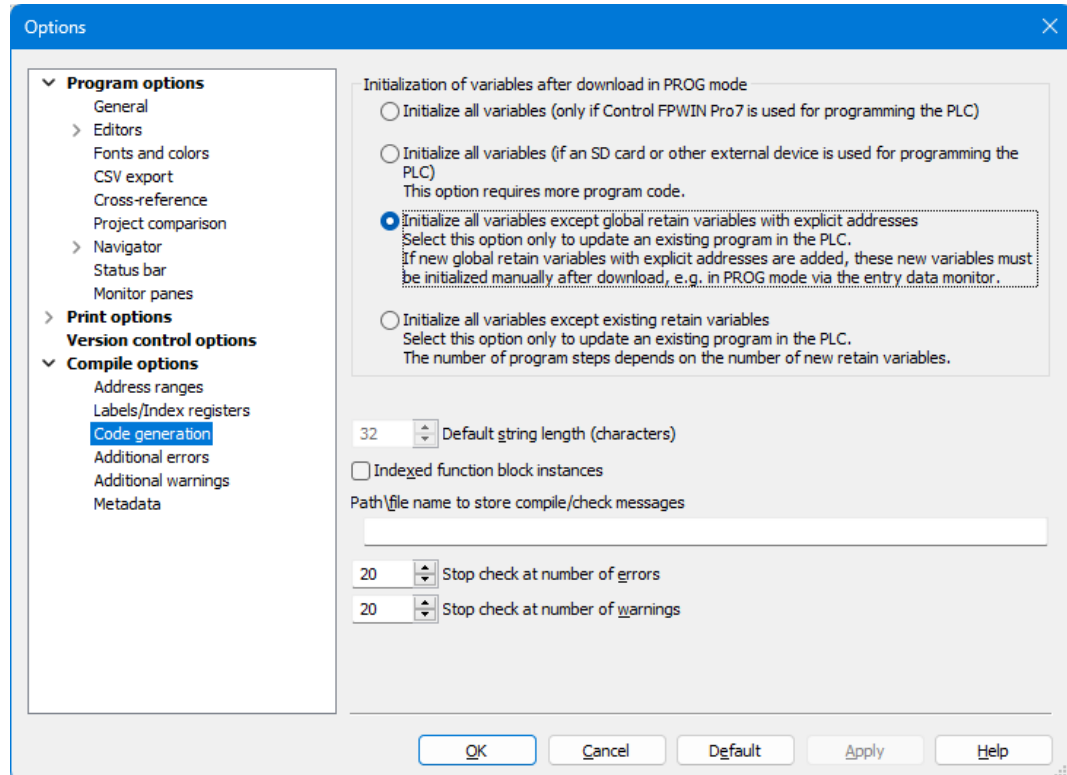
## 1.6 System registers and serial ports

Communication settings must be made in the system registers of the PLC and in the IEC 60870 communication unit.

Each sample project contains the correct settings for the system registers and serial ports.

### Note

- Changing the PLC type will reset all system registers and the COM/TOOL port parameters.
- The code generation settings under “Extras” > “Options” > “Compile options” > “Code generation” are very important. Make sure you select “Initialize all variables except global retain variables with explicit addresses” and deactivate “Indexed function block instances” (only available for certain PLC types).



- The following communication settings are valid for all PLCs:

“Communication mode”	MEWTOCOL-COM master/slave
“Data length”	8 bits
“Parity”	Odd
“Stop bits”	1 bit
“End code/reception done condition”	CR
“Start code”	No-STX
“Station number”	1
“Receive buffer starting address”	0
“Receive buffer capacity”	0

- The same parameters can be used by the Multi Communication Unit (MCU), Serial Communication Unit (FP7-SCU) or the TOOL port.

- The baud rate is not specified. With the FP7 it is possible to preset up to 230400 baud. For this baud rate we recommend to use a short communication cable.
- All parameters must be synchronized with the settings of the IEC 60870 communication unit.
- The following figure shows the typical settings for an FP2SH PLC:

COM1 X					
No	Item name	Data	Dimension	Range	Additional information
412	Communication mode	MEWTOCOL-COM slave [Computer link]		Unused	The PLC is a MEWTOCOL-COM slave unit. The slave receives c...
415	Station number	1		1 to 99	Station numbers are the numbers to identify the different PLC...
414	Baud rate	115200	baud	115200	Specifies the baud rate of the port.
413	Data length	8 bits		8 bits	Selects the data length.
413	Parity	Odd		None	Selects the parity check.
413	Stop bits	1 bit		1 bit	Specifies the number of stop bits.
413	Start code	No-STX		No-STX	Selects the start code.
413	End code/reception done condition	CR		CR	Selects the end code.
417	Receive buffer starting address	0		0 to 10239	Specifies the starting address of the receive buffer (available in...
418	Receive buffer capacity	0		0 to 1024	Specifies the number of words to be used for data reception (a...
416	Modem connection	Disabled		Disabled	Specifies if a modem is connected.

- The system registers for the retain variables must be specified in the “Hold on/off” section.
- The only important registers are “Data register hold area start address” and “Data register hold area start address” (FP2 and FP10SH).
- Only FP2 and FP10SH support FL registers. For all other PLC types, store all messages and object information in DT retain (hold) registers. In this case, approximately 80% of the memory can be allocated to the hold area (default: from DT8000).
- The PLC types FP2SH and FP10SH contain file registers. To store messages and object information, all IEC 60870 system data is linked to the file registers. In this case, approximately 95% of the file memory can be set to the hold area (default: starting at FL14000). The DT area is not used for user-specified (IEC 60870 specific) memory. All registers can be used by both the compiler and other non-IEC 60870 programs.
- Select “Extras” > “Options” > “Compile options” > “Address ranges” to set the required address ranges.

PLC (FPWIN Pro V6.1)	Dialog
FPΣ	
FP2SH	

If the settings are not suitable for the defined retain variables in the global variable list or the hold on/off settings in Control FPWIN Pro7, the compiler will stop and yield errors.

1.7 Glossary of terms

Find key terms and concepts explained.

ASDU

Application Service Data Unit. Data block transmitted between a supervisory control system and a remote terminal unit such as the FP-I4C unit using the IEC60870-5-104 protocol.



## **Automatic numbering**

For easy parameterization, it is very useful to use the internal numbering of the library. It will automatically set a unique number for each object.

Please take into account that if you add a new object between two other elements, automatic numbering will recreate the database and some data could be lost. Usually, the system will operate normally after the new automatic numbering. If not, reset all system buffers of the PLC.

## **Control direction**

IEC 60870 defines control direction as data transport from the supervisory control system to the controlled substation (RTU).

## **Class 1 and class 2 data**

The IEC 60870 standard defines the priority of sending data as class 1 (important) and class 2 (less important).

Typically, alarms should be transmitted as class 1 data, as this information is more important than cyclically transmitted measurement values.

The library declares the following process information as class 1:

- Alarms (single-point, double-point)
- General interrogation (single-point, double-point, measurement values, all data without time stamp)
- Counter interrogation (integrated totals)
- Acknowledgment of commands
- Acknowledgment of set points

## **IEC 60870 communication unit**

This refers to the units FP-I4C and FPWEB2 which support the IEC 60870 international standard for data transfer.

## **IEC 60870 international standard**

This standard provides a communication profile for sending basic telecontrol messages between a central telecontrol station and a telecontrol outstation, which uses permanent directly connected data circuits between the central station and the individual outstations.

## **Monitor direction**

IEC 60870 defines monitor direction as data transport from the substation (RTU) to the supervisory control system.

### Multi-instance function blocks

To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

You should not use indexing of function blocks, which is available for some PLC types.

### RTU (substation)

Normally an RTU (Remote Terminal Unit, Remote Telecontrol Unit, or Remote Telemetry Unit) is a substation with little or no PLC functionality. Together with an IEC 60870 communication unit, our PLCs combine RTU and PLC functions to create an intelligent outstation.

### Supervisory control system

This system can be used to control large water and sewage treatment plants, gas and energy suppliers. Also called controlling station.

## 1.8 Frequently Asked Questions (FAQs)

---

This page provides brief answers to common questions that may arise when using this library.

### The system runs well for a while but then no new signals are transferred to the supervisory control system.

Many data telegrams are sent in the monitor direction and the counter value in the monitor buffer increases until it reaches the maximum.

It probably is a performance problem. Please check your measurement values' **Cycle\_Min**, **Cycle\_Max**, and **Delta-Event** inputs. Use optimized values to prevent too many telegrams in the monitor direction.

### How can I check the data transfer of IEC 60870 communication?

There is a wide range of analyzer software available. Very useful serial testers (analyzers) are the freeware [www.wireshark.org](http://www.wireshark.org) and the special simulators from [Frontline Test Equipment](#).

### What happens if the monitoring buffer is full?

The system will stop creating new events until the buffer is cleared or space is freed due to data transfer.

**How fast are the FP0's counters?**

The FP0 and FPΣ can operate with hardware interrupts to count up the integrated totals. This feature allows counter frequencies up to 50kHz. To use this mode, some parameters in the system registers of the PLC (in Control FFWIN Pro7) must be checked.

**My integrated totals (counters) are not counting correctly. It seems that some pulses are not recognized.**

Check the parameters in the high-speed counter section of the system registers in Control FFWIN Pro7. The first four PLC inputs must be specified as interrupt triggers. It is also necessary to disable all other high-speed counter modes, e.g. pulse-catch input or high-speed counting (all channels).

**My integrated total (counter) is not counting correctly. The PLC counter indicates higher values than the hardware counter.**

If you use a relay contact, the high-speed counter will also recognize bouncing signals. Relay signals are low-speed counters. It is recommended to disable the high-speed counter mode for this input and to use a standard input signal with a positive edge.

**The signal input/output status of the function blocks in the POU IEC60870\_User is incorrect in online mode.**

This happens because there are multiple instances of the function blocks. The status you see is the status of the last function block call. Only the last state can be identified because monitor data is read at the end of the PLC cycle. Ignore this status information: only the associated input and output signals are relevant.

**How can I reset the PLC's entire memory?**

The easiest way to do this is to toggle the RUN/STOP switch on the front of the PLC three times. This will clear the monitoring buffer, the logging buffer, and the IEC 60870 internal functions.

**How long are the measurement values stored in the PLC until they must be transferred to the supervisory control system?**

It depends on the PLC type, the number of measurement values, events and memory installed. The FP0 has 16k of data memory. A maximum of 12k can be used as a logging buffer. At three events per minute, this memory provides a history of approximately 10 hours.

An FPΣ PLC with a memory unit (512KB) provides a history of 162 hours (more than six days).

See also [Advanced system description information](#) (page 12), [Recommended PLC types](#) (page 6).

### **How can I resolve the compilation error "C5059 Initial value missing"?**

Go to "Extras" > "Options" > "Compile options" > "Code generation" and select "Initialize all variables except global retain variables with explicit addresses".

### **How can I resolve compilation errors?**

If the error "F1101 Wrong address access size identifier" or "VAR\_GLOBAL\_RETAIN does not match the address in the non-hold area" occurs, the settings under "Compile options" or the address links between identifiers and data registers in the global variable list are incorrect.

If you use the PLC's COM port for active data transfer, e.g. for leased lines, the defined receive buffer of this COM port may be in the memory area used by the compiler. Check the compile options and the COM port settings in the system registers. If the error is not caused by the COM port settings, refer to the programming examples installed with the library.

### **How can I resolve negative acknowledgments from the PLC?**

The PLC and IEC 60870 communication unit are correctly installed, the program has been compiled without errors and is running in the PLC. However, the PLC responds to telegrams from the supervisory control system with NACK.

In this case, the communication between the PLC and the IEC 60870 communication unit is not working. Check the wiring between the units and the baud rate as well as the system register settings for the PLC's serial port.

## 2 PLC examples and basic programs

Two basic programming examples are part of the library installation and can be adapted to your needs.

The sample programs can be used with the following PLC types:

FP0H, FP7, FP-XH, FP0R-T32CP

Discontinued: FPΣ, FP2SH, FP-X, FP0-T32CP

Start with the FP0H example (IEC60870\_FP0H\_Redundancy\_EN.pce) for small PLC types and do not require special buffer management for class 1 and class 2 data.

You can add your own POU's to the example as needed.

Station type	File name	Comment
Standard station	IEC60870_FP0H_Class1-Class2_EN.pce IEC60870_FP7_Class1-Class2_EN.pce	Recommended for optimal buffer management and splitting of class 1 and class 2 data.
Station with a single buffer	IEC60870_FP0H_Redundancy_EN.pce	Optimized for using a single (or redundancy) buffer without splitting of class 1 and class 2 data. All data must be located between DT0 and DT65533. Not recommended for newer installations via IEC60870-5-104.

### Note

The following applies to all examples.

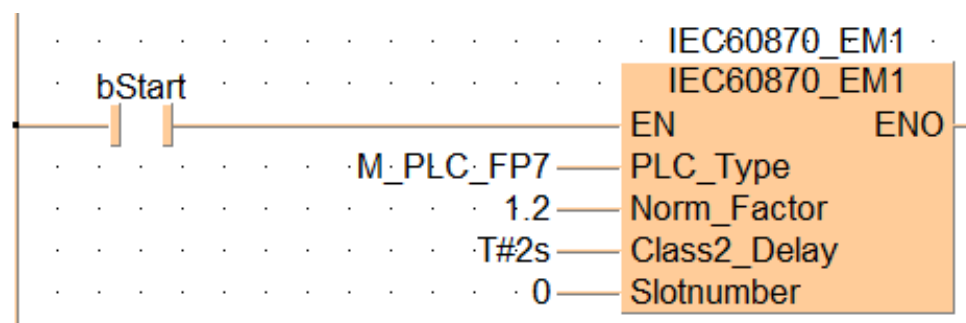
Be careful when changing DT addresses in the PLC. If the address is changed for the variables **IEC870\_RBM...**, **IEC870\_Monitor...**, or **IEC870\_Control...**, the corresponding setting in the IEC60870 communication unit must also be updated.

### 2.1 Example for standard stations

Use the examples **IEC60870\_FP0H\_Class1-Class2\_EN.pce** and **IEC60870\_FP7\_Class1-Class2\_EN.pce** if you want to use the large buffers of the PLC. The measurement and counter values are split into class 1 and class 2 data and stored in separate buffers before being transferred to the main station.

Add your own POU's as needed.

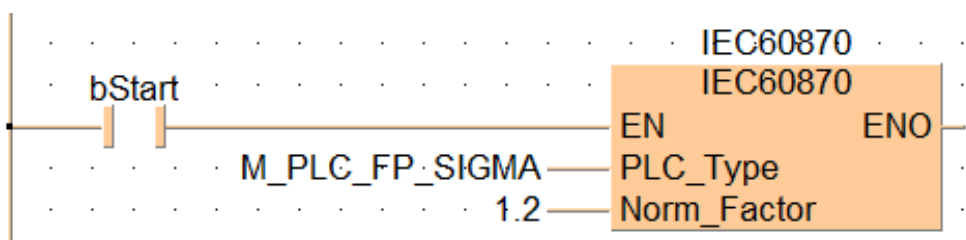
The IEC 60870 main function block handles the communication between the PLC and the IEC communication unit.



## 2.2 Example for stations without class 1/class 2 buffer management

Use this example **IEC60870\_FP0H\_Redundancy\_EN.pce** if your substations are directly connected to the main station and you only need one buffer to store the data in the PLC before sending it to the main station.

The IEC 60870 main function block handles the communication between the PLC and the IEC communication unit.



In the POU's subsequent networks, define the tags you wish to transmit to the master. The example programs are complete RTUs. Add or delete tags as needed.

### 3 Debugging and monitoring

If the system is not working properly or you want to monitor your application, the M\_IEC 60870\_PRO7 library provides several ways to check your system and application.

#### 3.1 Detecting process commands with IEC\_Debug

The library's global variable list includes the DUT **IEC\_Debug** that can be used to detect commands from the supervisory control system.

Use the online monitoring function of Control FPGWIN Pro7 to view the status of the DUT elements.

	Identifier	Value	FP address	Comment
1	IEC_Debug			Debug info to fix hotline problems
2	Reset	FALSE	R290	Reset from main station recognized
3	InterrogationA	FALSE	R291	General interrogation from main station recognized
4	InterrogationC	FALSE	R292	Gen.interrogation of integrated totals from main station recognized
5	TestCommand	FALSE	R293	Test command from main station recognized
6	Clocksyc	FALSE	R294	Time synchronisation from main station recognized
7	Command	FALSE	R295	Single/double command from main station recognized
8	Setpoint	FALSE	R296	Setpoint/bitstring command from main station recognized
9	ProcessReset	FALSE	R297	Process reset command from main station recognized
10	ReadData	FALSE	R298	Read command from main station recognized

The binary information in the DUT is static, i.e. whenever the PLC detects a command from the supervisory control system (e.g. InterrogationC for counters, InterrogationA for general interrogation), the corresponding bit will be set. A reset cannot be triggered automatically by internal functions. To repeat a test, you can manually reset any of the DUT's bits (**do not use the force function!**). Simply select a line, type FALSE (or 0) and press <Enter>.

Identifier	Description
<b>Reset</b>	TRUE after a cold or warm start of the PLC.
<b>InterrogationA</b>	TRUE if a general interrogation command (type 100) has been detected.
<b>InterrogationC</b>	TRUE if a general interrogation command of integrated totals (type 101) has been detected.
<b>TestCommand</b>	TRUE if a test command (type 104 or 107) has been detected.
<b>Clocksyc</b>	TRUE if a clock synchronization command (type 103) has been detected.
<b>Command</b>	TRUE if a single-point or double-point command (type 45, 46, 58, 59) has been detected.
<b>Setpoint</b>	TRUE if a set point command (type 48, 49, 50, 51, 61, 62, 63, 64) has been detected.
<b>ProcessReset</b>	TRUE if a process reset command (type 105) has been detected.
<b>ReadData</b>	TRUE if a read data command (type 102) has been detected.

## 3.2 Monitoring with IEC\_Bits

The DUT **IEC\_Bits** in the library's global variable list provides information about the running system and some control flags.

The status of only a few flags can be changed manually. All others are read-only as they are used by the internal system functions of the M\_IEC60870\_PRO7 library.

	Identifier	Value	FP address	Comment
1	IEC_Bits			Internal data communication
2	Reset	FALSE	R250	Reset all buffers
3	Trigger1Sec	TRUE	R251	Time trigger to count delays of measurements
4	Init_PLCStart	FALSE	R252	Information to the main station in end_of_init command
5	Init_ok	TRUE	R253	Unit ready
6	GA_active	FALSE	R254	General interrogation is running
7	GA_start	FALSE	R255	General interrogation is started
8	GA_IT_active	FALSE	R256	General interrogation of integrated totals is running
9	Control_Checked	FALSE	R257	Command recognized (delete first control buffer data set)
10	Control_New	FALSE	R258	New Command in buffer
11	GA_SP_not_ready	FALSE	R259	General interrogation is running (set by all user functions, if GA is recognized)
12	GA_ME_not_ready	FALSE	R25A	General interrogation is running (set by all user functions, if GA is recognized)
13	GA_IT_not_ready	FALSE	R25B	General interrogation of integrated totals is running (set by all user functions, if GA is recognized)
14	CP24HourChange	FALSE	R25C	Change of hour detected (next time used telegram send a sync. telegram first)
15	TimeValid	TRUE	R25D	PLC time is valid (BCD-format test)
16	DaylightSavingTime	TRUE	R25E	Daylight saving time (summer)
17	ProcessReset	FALSE	R25F	Process Reset from main station is running
18	BufferError	FALSE	R260	Wrong buffer status
19	LoggingBuffer	FALSE	R261	Second logging buffer used
20	EM1Used	TRUE	R262	Second logging buffer used with EM1 unit
21	LoggingOut	FALSE	R263	Remove data from logging buffer (modem support)
22	OnlineStarted	TRUE	R264	Modem has new connection (modem support)
23	Class1Data1	TRUE	R265	Will be set from single point only
24	Class1Data2	TRUE	R266	Will be set from single point only
25	StartDT1	FALSE	R267	Communication -104 is fully running (task1)
26	StartDT2	FALSE	R268	Communication -104 is fully running (task2)
27	SP_Invalid	FALSE	R269	Set the quality invalid bit for each value
28	ME_Invalid	FALSE	R26A	Set the quality invalid bit for each value
29	BO_Invalid	FALSE	R26B	Set the quality invalid bit for each value
30	IT_Invalid	FALSE	R26C	Set the quality invalid bit for each value
31	SP_NotTopical	FALSE	R26D	Set the quality not topical bit for each value
32	SP_Trigger	FALSE	R26E	Force send, be carefully to prevent buffer overflow! Reset in IEC60870 main function blocks
33	ME_NotTopical	FALSE	R26F	Set the quality not topical bit for each value
34	BO_NotTopical	FALSE	R270	Set the quality not topical bit for each value
35	Cause_P_N	FALSE	R271	Set the cause of transmission individually positive (FALSE) or negative (TRUE). Default value: FALSE
36	GA_Timestamp	FALSE	R272	Set the option to use general interrogation with time stamped values
37	StartDT			Communication -104 is fully running (task1-4)
38	[0]	TRUE	R280	Communication -104 is fully running (task1-4)
39	[1]	TRUE	R281	Communication -104 is fully running (task1-4)
40	[2]	FALSE	R282	Communication -104 is fully running (task1-4)
41	[3]	FALSE	R283	Communication -104 is fully running (task1-4)
42	Config_ok			Buffer is initialized
43	[0]	TRUE	R288	Buffer is initialized
44	[1]	TRUE	R289	Buffer is initialized
45	[2]	TRUE	R28A	Buffer is initialized
46	[3]	TRUE	R28B	Buffer is initialized

Useful **read-only** bits are listed below:

Identifier	Description
Init_ok	TRUE after a cold or warm start.
TimeValid	TRUE if the real-time clock has a valid time format.



Identifier	Description
<b>DaylightSavingTime</b>	TRUE if the daylight saving time in Central Europe has been set with <b>M_IEC60870_RTC</b> .
<b>StartDT1</b>	Obsolete: TRUE if an IEC60870-5-104 Ethernet communication (task 1) has been established.
<b>StartDT2</b>	Obsolete: TRUE if an IEC60870-5-104 Ethernet communication (task 2) has been established.
<b>startDT[]</b>	Array for up to 4 connections. Each bit shows the connection status. TRUE if an IEC60870-5-104 Ethernet communication has been established.
<b>Config_ok[]</b>	Array for RBM buffers. Each bit shows the configuration status. TRUE if the configuration is ok.
<b>Class1Data1</b>	TRUE if the buffer contains class 1 data (supported by EM1 function blocks). Will never be reset internally.
<b>Class1Data2</b>	TRUE if the buffer contains class 1 data (supported by EM1 function blocks). Will never be reset internally.

Useful **write access** bits:

Identifier	Description
<b>SP_Invalid</b>	<p>FALSE if the single-point values are valid. Turn to TRUE if the physical input unit is damaged.</p> <p>This flag is used to monitor all single-point values. To use the flag individually for each process information, set <b>IEC_Bits.SP_Invalid</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b>.</p>
<b>ME_Invalid</b>	<p>FALSE if the measurement values are valid. Turn to TRUE if the physical analog input or the unit is damaged.</p> <p>This flag is used to monitor measurement values. To use the flag individually for each process information, set <b>IEC_Bits.ME_Invalid</b> just before calling a monitoring function block such as <b>IEC_M_ME_NC_1</b>.</p>
<b>BO_Invalid</b>	<p>FALSE if the bit string data is valid. Turn to TRUE if the physical data source is damaged.</p> <p>This flag is used to monitor bit string data. To use the flag individually for each process information, set <b>IEC_Bits.BO_Invalid</b> just before calling a monitoring function block such as <b>IEC_M_BO_NA_1</b>.</p>
<b>IT_Invalid</b>	<p>FALSE if the integrated totals (counter values) are valid. Turn to TRUE if the physical input unit is damaged.</p> <p>This flag is used to monitor integrated totals. To use the flag individually for each process information, set <b>IEC_Bits.IT_Invalid</b> just before calling a monitoring function block such as <b>IEC_M_IT_TB_1</b>.</p>
<b>SP_NotTopical</b>	<p>FALSE if the most recent update of the single-point value was successful. Turn to TRUE if the value is outdated.</p> <p>This flag is used to monitor all single-point values. To use the flag individually for each process information, set <b>IEC_Bits.SP_Topical</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b>.</p>

Identifier	Description
<b>ME_NotTopical</b>	FALSE if the most recent update of the measurement value was successful. Turn to TRUE if the value is outdated.  This flag is used to monitor all measurement values. To use the flag individually for each process information, set <b>IEC_Bits.ME_Topical</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b> .
<b>BO_NotTopical</b>	FALSE if the most recent update of the string data was successful. Turn to TRUE if the value is outdated.  This flag is used to monitor all string data. To use the flag individually for each process information, set <b>IEC_Bits.BO_Topical</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b> .
<b>Cause_P_N</b>	FALSE if the command was confirmed. Turn to TRUE if it was not confirmed.  To use the flag individually for each process information, set <b>IEC_Bits.Cause_P_N</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b> .
<b>GA_Timestamp</b>	FALSE if no timestamp is transmitted with general interrogations. Turn to TRUE if a timestamp is required.  To use the flag individually for each process information, set <b>IEC_Bits.GA_Timestamp</b> just before calling a monitoring function block such as <b>IEC_M_SP_NA_1</b> .

All other information is dynamic and cannot be monitored very well.

Related topics

[Advanced system description information](#)

### 3.3 Checking the buffer in the monitor direction

The buffer can be monitored with the DUTs **IEC870\_RBM1** and **IEC870\_RBM2**, which can be found in the global variable list of the programming examples.

Use the online monitoring function to view the status of the DUT elements.

	Identifier	Value	FP address	Comment
1	IEC870_RBM1			1000 IEC60870 Header for monitoring buffer
2	Identification	8081	DT8000	Identification (buffer initialized)
3	CheckedPosition	12126	DT8001	Feedback pointer if IEC-Communicator
4	NextPosition	12828	DT8002	Pointer to the next used buffer position
5	LastChecked	12126	DT8003	Pointer to recognize overflows
6	StartPosition	12126	DT8004	First address of the buffer (to set the first data set)
7	LastPosition	14674	DT8005	Last address (+1) of the used buffer (one data set is following for special use)
8	Area	5	DT8006	Data area of the buffer DT/FL
9	Overflow	0	DT8007	Buffer overflow (to calculate pointers of the ring buffer)
10	Blocklen	26	DT8008	Length of a data set (default = 25)
11	ResetTimer	15	DT8009	Setpoint for the buffer reset delay
12	ResetCountdown	1	DT8010	Countdown to finish the buffer reset
13	Modemstatus	16#0000	DT8011	Information from IEC-Communicator
14	ASDU0	0	DT8012	1 Byte for ASDU (for multiple stations)
15	BufferSize	99	DT8013	Buffer size (to recognize changes)
16	Counter	2	DT8014	Count of data sets, they are not transferred with the IEC-Communicator
17	CounterMax	27	DT8015	Drag indicator for debugging (to detect the peaks)
18	LogOutLimit	0	DT8016	For FPG-EM1 used only: For IEC870_Logging buffered data only (default = 15)
19	Spare			Not used but must be clear for future changes

The status of integer values and word addresses is dynamic, i.e. their values can change during communication.

The only value you are allowed to change is **Identification**. The value of this variable must be 16#8080 (version 1.x) or 16#8081 (from version 2.x). Otherwise, the system is not running or the IEC 60870 main function block is not installed. If you manually overwrite the value of **Identification**, the buffer will be reset. This reset can solve transmission problems in newly installed systems. After writing a value, 16#8080 or 16#8081 will immediately be displayed in the data register.

#### Note

Changing the value of any other variable can lead to serious malfunctions.

Function	Explanation
<b>Identification</b>	Identification of the active buffer
<b>CheckedPosition</b>	Internal pointer, corresponds to <b>NextPosition</b> (see note)
<b>NextPosition</b>	Internal pointer, corresponds to next <b>CheckedPosition</b> (see note)
<b>LastChecked</b>	Internal pointer
<b>StartPosition</b>	Start address of the monitor buffer (must be identical to the DT/FL address in the global variable list)
<b>LastPosition</b>	Last address of the monitor buffer. Calculated value for the last data set
<b>Area</b>	Data area DT=5, FL=9
<b>Overflow</b>	Internal information about the buffer state. If -1, it is not an error, but some data is at the end position of the linear buffer and some telegrams are at the start position.
<b>Blocklen</b>	Data set length (default = 26)

Function	Explanation
<b>ResetTimer</b>	If the buffer is to be restored, a delay time will run before it is ready. For IEC60870-5-104, use a delay time $\geq 15$ s. This time is related to the -104 time value t2.
<b>ResetCountdown</b>	Elapsed time (when ready: 1)
<b>Counter</b>	Number of data sets in the queue for transmission to the supervisory control system
<b>CounterMax</b>	Static. Highest count value since the last buffer reset. May be set to 0.

#### Note

If the system is running well and IEC 60870 communication is established, **CheckedPosition** and **NextPosition** should have the same value and the counter should decrease to 0.

## 3.4 Checking the PLC's real-time clock

You can check the PLC's real-time clock with Control FPWIN Pro7.

1. Change to online mode.
2. Go to "Monitor" > "Special flags and registers" > "Clock/calendar functions".  
A new window with the current date and time information opens. You can overwrite the values here.

#### Note

The system variable **sys\_wClockCalendarHourMin** (available for some PLCs only) is read-only .

You can also set the date and time from the supervisory control system with the IEC60870-5-101 telegram type 103 or with the NTP protocol (more details can be found in the FP Web Configurator tool).

#### Related topics

[Advanced system description information](#)

## 4 Function blocks of this library

Find detailed descriptions of all input and output parameters.

### 4.1 IEC 60870 main function blocks

Every program must start with one of these function blocks, as they are responsible for the memory handling and system command processing.

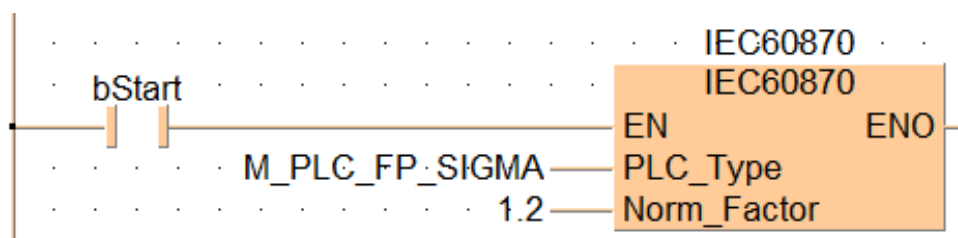
#### 4.1.1 IEC60870

Function block to control the data transfer to the IEC 60870 communication unit.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

The complete memory handling and processing of system commands (e.g. interrogation, clock synchronization) takes place inside this function block.

The number of used buffers must be configured with the constant global variable **IEC60870\_REDUNDANCY**.



#### Parameters

Input

**EN** (BOOL)

Activates the function block.

**PLC\_Type** (INT)

Set the PLC type for the calculation of normalized values. Use a constant to specify the PLC type.

Values:

**M\_PLC\_FP0**  
**M\_PLC\_FP0\_0\_20mA**  
**M\_PLC\_FP0H**  
**M\_PLC\_FP0H\_0\_20mA**

**M\_PLC\_FP\_SIGMA**  
**M\_PLC\_FP\_SIGMA\_0\_20mA**  
**M\_PLC\_FPX**  
**M\_PLC\_FPX\_0\_20mA**  
**M\_PLC\_FP10SH**  
**M\_PLC\_FP2**  
**M\_PLC\_FP2\_0\_20mA**  
**M\_PLC\_7**  
**M\_PLC7\_0\_20mA**  
**M\_PLC\_NO\_NORM**

#### **Norm\_Factor (REAL)**

Set a system-specific factor for the calculation of normalized values (default = 1.2).

#### **Related topics**

[Global variables and useful constants](#) (page 10)

### **4.1.2 IEC60870\_ASDU**

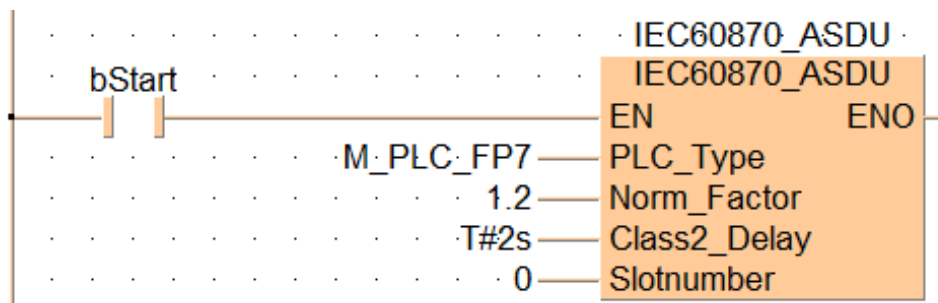
Function block to control the data transfer to the IEC 60870 communication unit via a gateway.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

This function block handles additional ASDU information used by gateway applications (the ASDU is often used to divide the data of different logical stations within the gateway).

The complete memory handling and processing of system commands (e.g. interrogation, clock synchronization) takes place inside this function block.

The number of used buffers must be configured with the constant global variable **IEC60870\_REDUNDANCY**.



#### **Parameters**

Input

**EN (BOOL)**

Activates the function block.

**PLC\_Type (INT)**

Set the PLC type for the calculation of normalized values. Use a constant to specify the PLC type.

Values:

**M\_PLC\_FP0**  
**M\_PLC\_FP0\_0\_20mA**  
**M\_PLC\_FP0H**  
**M\_PLC\_FP0H\_0\_20mA**  
**M\_PLC\_FP\_SIGMA**  
**M\_PLC\_FP\_SIGMA\_0\_20mA**  
**M\_PLC\_FPX**  
**M\_PLC\_FPX\_0\_20mA**  
**M\_PLC\_FP10SH**  
**M\_PLC\_FP2**  
**M\_PLC\_FP2\_0\_20mA**  
**M\_PLC\_7**  
**M\_PLC7\_0\_20mA**  
**M\_PLC\_NO\_NORM**

**Norm\_Factor (REAL)**

Set a system-specific factor for the calculation of normalized values (default = 1.2).

**Class2\_Delay (TIME)**

Set the delay to move the buffered class 2 data to the class 1 buffer. The main class 1 buffers are **IEC870\_RBM...**/**IEC870\_Monitor...** (default: T#2s).

**Slotnumber (INT)**

Set the slot number of the FPG-EM1 unit (default value 0; input is ignored if unit is not used).

## Related topics

[Global variables and useful constants](#) (page 10)

### 4.1.3 IEC60870\_EM1

---

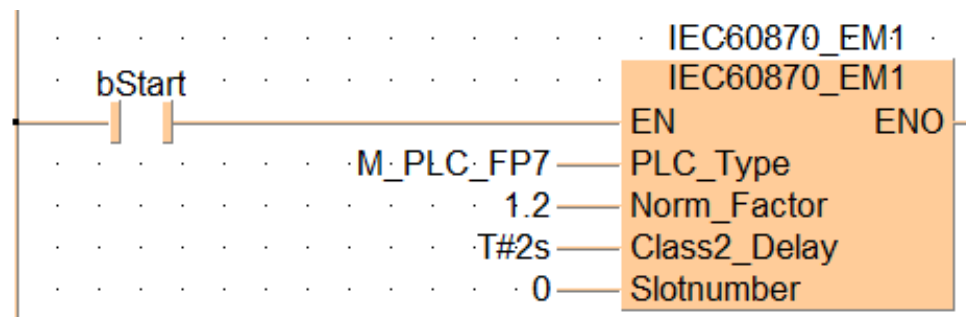
Function block to control the data transfer to the IEC 60870 communication unit for PLCs with a large buffer, allowing class 1/class 2 buffer management.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

This function block is recommended for all new installations.

The complete memory handling and processing of system commands (e.g. interrogation, clock synchronization) takes place inside this function block.

The number of used buffers must be configured with the constant global variable **IEC60870\_REDUNDANCY**.



## Parameters

Input

**EN** (BOOL)

Activates the function block.

**PLC\_Type** (INT)

Set the PLC type for the calculation of normalized values. Use a constant to specify the PLC type.

Values:

**M\_PLC\_FP0**  
**M\_PLC\_FP0\_0\_20mA**  
**M\_PLC\_FP0H**  
**M\_PLC\_FP0H\_0\_20mA**  
**M\_PLC\_FP\_SIGMA**  
**M\_PLC\_FP\_SIGMA\_0\_20mA**  
**M\_PLC\_FPX**  
**M\_PLC\_FPX\_0\_20mA**  
**M\_PLC\_FP10SH**  
**M\_PLC\_FP2**  
**M\_PLC\_FP2\_0\_20mA**  
**M\_PLC\_7**  
**M\_PLC7\_0\_20mA**  
**M\_PLC\_NO\_NORM**

**Norm\_Factor** (REAL)

Set a system-specific factor for the calculation of normalized values (default = 1.2).



**Class2\_Delay** (TIME)

Set the delay to move the buffered class 2 data to the class 1 buffer. The main class 1 buffers are **IEC870\_RBM.../IEC870\_Monitor...** (default: T#2s).

**Slotnumber** (INT)

Set the slot number of the FPG-EM1 unit (default value 0; input is ignored if unit is not used).

Related topics

[Global variables and useful constants](#) (page 10)

**4.2 Process information in monitor direction**

These function blocks send process information, such as measurements and events, from the substation (RTU) to the supervisory control system.

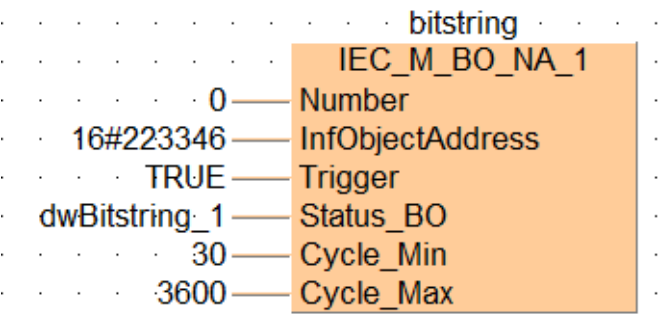
**4.2.1 IEC\_M\_BO\_NA\_1**

Function block for type identifier 7 to send 32-bit strings in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes, a new message is sent to the IEC 60870 communication unit.



**Parameters**

Input

**Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### **InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### **Trigger** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### **Status\_BO** (DWORD)

Set a bit string of 32 bits.

### **Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

### **Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

---

## 4.2.2 IEC\_M\_BO\_TA\_1

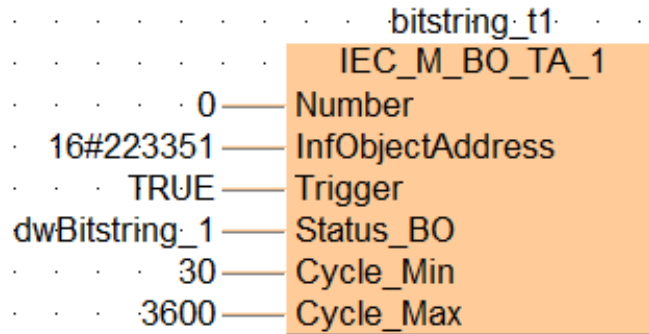
---

Function block for type identifier 8 to send 32-bit strings with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### Status\_BO (DWORD)

Set a bit string of 32 bits.

### Cycle\_Min (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

### Cycle\_Max (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

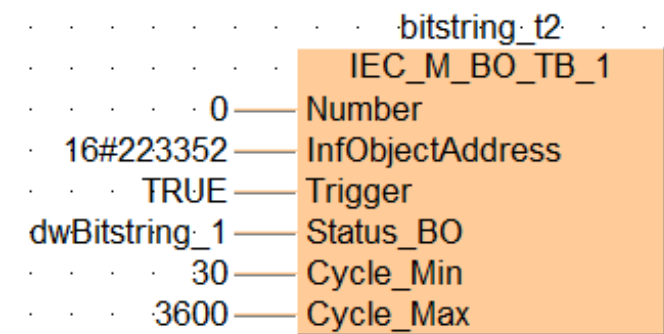
4.2.3 IEC\_M\_BO\_TB\_1

Function block for type identifier 33 to send 32-bit strings with a CP56Time2A time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes, a new message is sent to the IEC 60870 communication unit.



Parameters

Input

**Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_BO (DWORD)**

Set a bit string of 32 bits.

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

#### 4.2.4 IEC\_M\_BO\_TB\_EM1

---

Function block for type identifier 33 to send 32-bit strings with a CP56Time2A time tag in monitor direction (with buffer management).

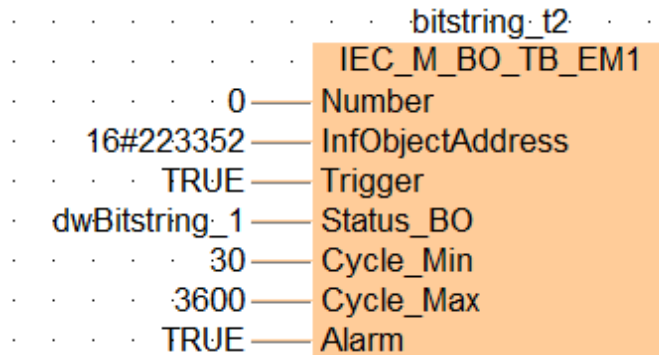
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time

tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### Status\_BO (DWORD)

Set a bit string of 32 bits.

### Cycle\_Min (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

### Cycle\_Max (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

### Alarm (BOOL)

Set the alarm mode.

Values:

TRUE: When a bit at **Status\_BO** changes, a new class 1 message is sent to the supervisory control system, even if a connection has to be established. Use this mode for alarm signals.

FALSE: When a bit at **Status\_BO** changes, a new message in the buffer is created, which is not sent immediately but the next time a connection is established. Use this mode for status signals of actuators and switches but not for alarms.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.


### Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.5 IEC\_M\_DP\_NA\_1

Function block for type identifier 3 to send double-point information in monitor direction.

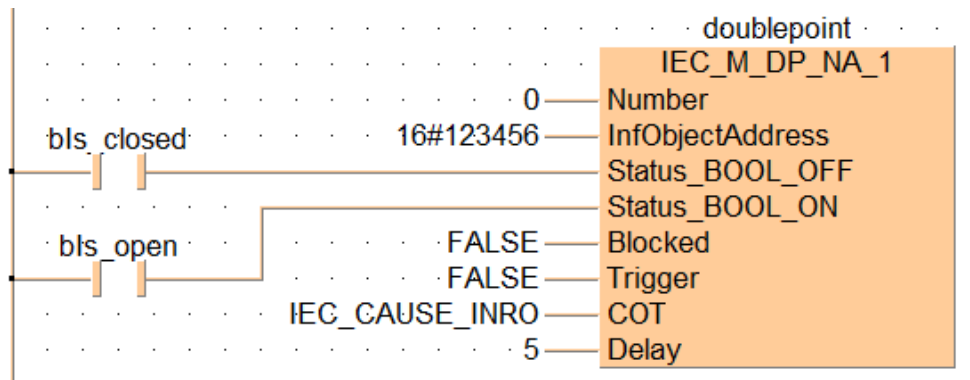
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library .

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL\_OFF (BOOL)

OFF state signal.

### Status\_BOOL\_ON (BOOL)

ON state signal.

### Blocked (BOOL)

Blocked state signal. The command was cancelled.

### Trigger (BOOL)

Turn to TRUE to update the status (e.g. output is blocked). This input is not needed in normal operation.

### COT (WORD)

Cause of transmission, e.g. IEC\_CAUSE\_INRO (value 12)

### Delay (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)



### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.6 IEC\_M\_DP\_NA\_2

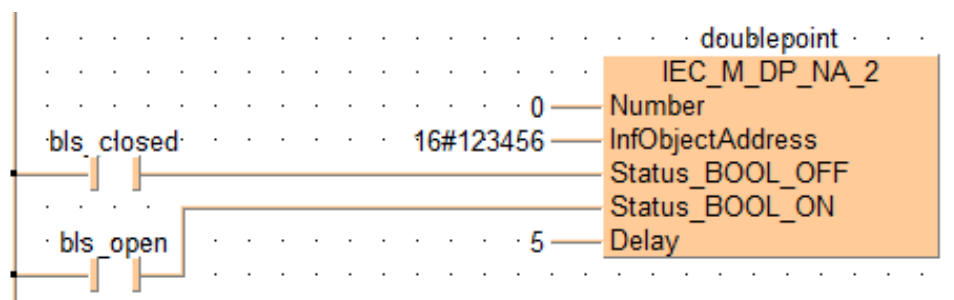
Function block for type identifier 3 to send double-point information in monitor direction (without **Blocked**, **Trigger**, and **COT** inputs).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

- A message is sent if **Status\_BOOL\_OFF** is TRUE.
- A message is sent if **Status\_BOOL\_ON** is TRUE.
- A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
- No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).



### Parameters

#### Input

##### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Status\_BOOL\_OFF** (BOOL)

OFF state signal.

**Status\_BOOL\_ON** (BOOL)

ON state signal.

**Delay** (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)

## Note

- Do not use triggers for the input signals **Status\_BOOL\_OFF** or **Status\_BOOL\_ON**.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.7 IEC\_M\_DP\_NA\_EM1

---

Function block for type identifier 3 to send double-point information in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

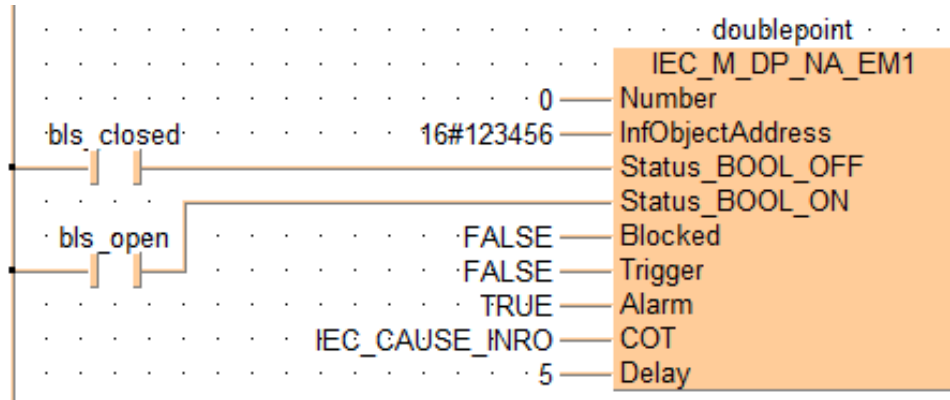
You must always start your program with **IEC60870\_FPG\_EM1**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL\_OFF (BOOL)

OFF state signal.

### Status\_BOOL\_ON (BOOL)

ON state signal.

### Blocked (BOOL)

Blocked state signal. The command was cancelled.

### Trigger (BOOL)

Turn to TRUE to update the status (e.g. output is blocked). Use a rising edge! This input is not needed in normal operation.

### Alarm (BOOL)

Set the alarm mode.

### Values:

TRUE: When a bit at **Status\_BO** changes, a new class 1 message is sent to the supervisory control system, even if a connection has to be established. Use this mode for alarm signals.

FALSE: When a bit at **Status\_BO** changes, a new message in the buffer is created, which is not sent immediately but the next time a connection is established. Use this mode for status signals of actuators and switches but not for alarms.

### COT (WORD)

Cause of transmission, e.g. IEC\_CAUSE\_INRO (value 12)

### Delay (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.8 IEC\_M\_DP\_TA\_1

---

Function block for type identifier 4 to send double-point information with a time tag in monitor direction.

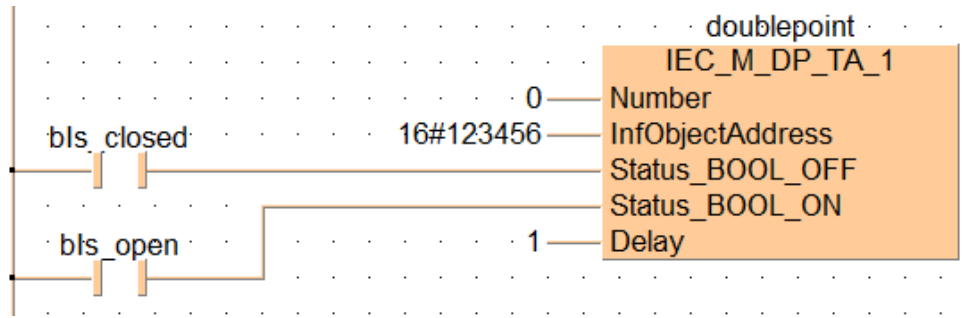
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL\_OFF (BOOL)

OFF state signal.

### Status\_BOOL\_ON (BOOL)

ON state signal.

### Delay (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)

## Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.2.9 IEC\_M\_DP\_TB\_1

Function block for type identifier 31 to send double-point information with a CP56Time2A time tag in monitor direction.

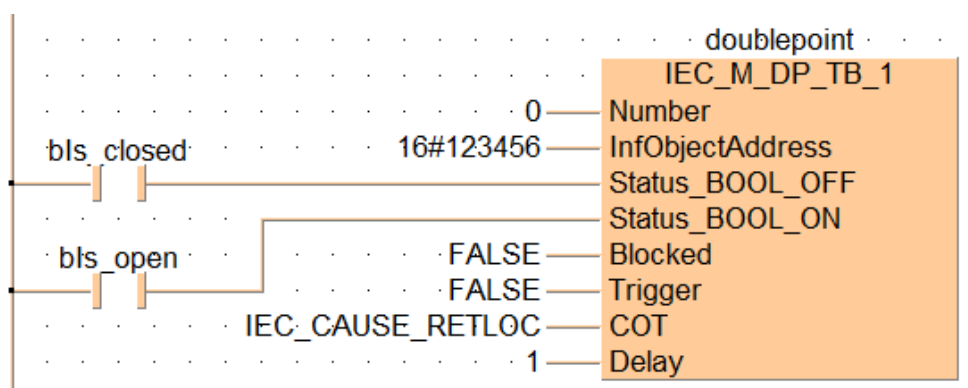
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

##### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Status\_BOOL\_OFF (BOOL)

OFF state signal.

**Status\_BOOL\_ON** (BOOL)

ON state signal.

**Blocked** (BOOL)

Blocked state signal. The command was cancelled.

**Trigger** (BOOL)

Turn to TRUE to update the status (e.g. output is blocked). Use a rising edge! This input is not needed in normal operation.

**COT** (WORD)

Cause of transmission, e.g. IEC\_CAUSE\_INRO (value 12)

**Delay** (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.10 IEC\_M\_DP\_TB\_1EM1

---

Function block for type identifier 31 to send double-point information with a CP56Time2A time tag in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

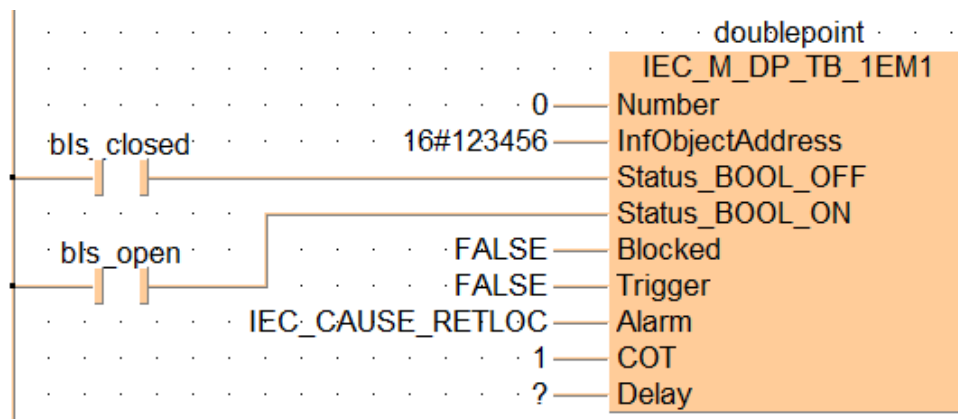
Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.

4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL\_OFF (BOOL)

OFF state signal.

### Status\_BOOL\_ON (BOOL)

ON state signal.

### Blocked (BOOL)

Blocked state signal. The command was cancelled.

### Trigger (BOOL)

Turn to TRUE to update the status (e.g. output is blocked). Use a rising edge! This input is not needed in normal operation.



**Alarm (BOOL)**

Set the alarm mode.

Values:

TRUE: When a bit at **Status\_BO** changes, a new class 1 message is sent to the supervisory control system, even if a connection has to be established. Use this mode for alarm signals.

FALSE: When a bit at **Status\_BO** changes, a new message in the buffer is created, which is not sent immediately but the next time a connection is established. Use this mode for status signals of actuators and switches but not for alarms.

**COT (WORD)**

Cause of transmission, e.g. IEC\_CAUSE\_INRO (value 12)

**Delay (BOOL)**

Set a debounce time.

Values: 0...127s (0: no debouncing)

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870\\_EM1](#)

[Glossary of terms](#)

**4.2.11 IEC\_M\_DP\_TB\_2**

Function block for type identifier 31 to send double-point information with a CP56Time2A time tag in monitor direction.

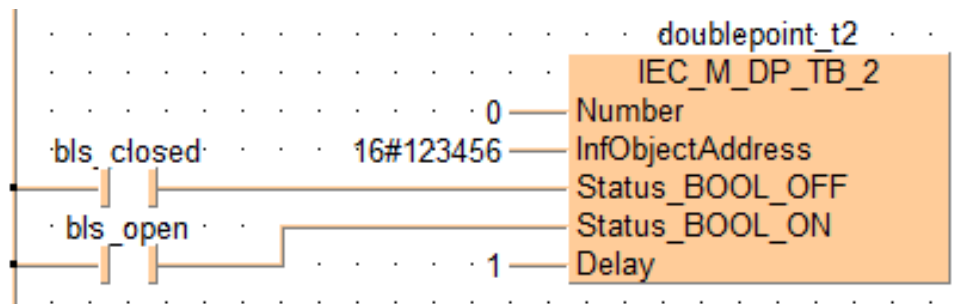
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double-point information can assume four states and can be used to transmit states more reliably than single-point information, for example to control switches and valves.

1. A message is sent if **Status\_BOOL\_OFF** is TRUE.
2. A message is sent if **Status\_BOOL\_ON** is TRUE.
3. A message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are FALSE.
4. No message is sent if **Status\_BOOL\_OFF** and **Status\_BOOL\_ON** are TRUE (invalid status).

Additionally, the output signal can be blocked using the **Blocked** input.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL\_OFF (BOOL)

OFF state signal.

### Status\_BOOL\_ON (BOOL)

ON state signal.

### Delay (BOOL)

Set a debounce time.

Values: 0...127s (0: no debouncing)

Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

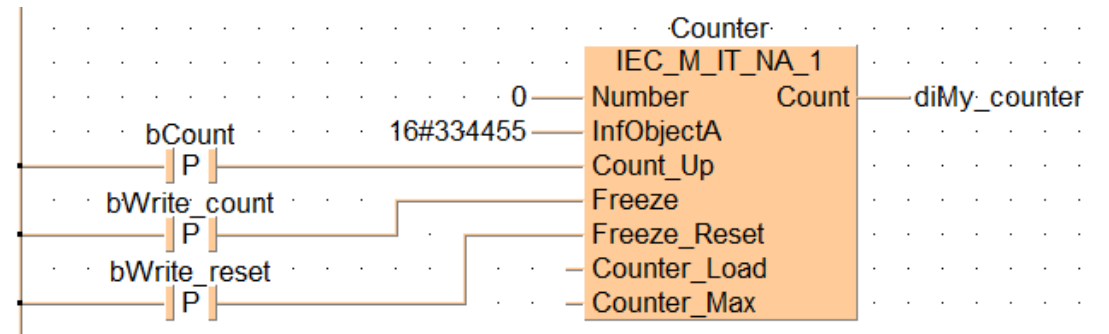
[Glossary of terms](#)

### 4.2.12 IEC\_M\_IT\_NA\_1

Function block for type identifier 15 to send integrated totals in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

##### InfObjectA (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Count\_Up (BOOL)

Turn to TRUE to increment the counter value. Use a rising edge!

##### Freeze (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

##### Freeze\_Reset (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

##### Counter\_Load (DINT)

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

**Counter\_Max** (DINT)

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- This function block is designed for low-speed counters < 2Hz.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

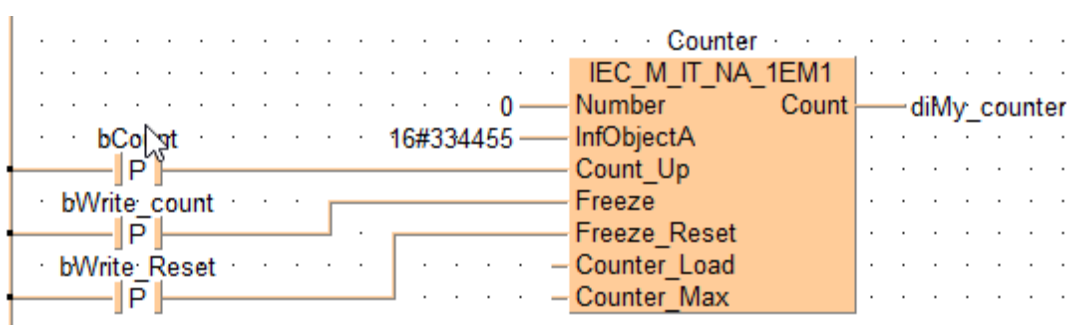
### 4.2.13 IEC\_M\_IT\_NA\_1EM1

Function block for type identifier 15 to send integrated totals in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



## Parameters

### Input

#### **Number** (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

#### **InfObjectA** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### **Count\_Up** (BOOL)

Turn to TRUE to increment the counter value. Use a rising edge!

#### **Freeze** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

#### **Freeze\_Reset** (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

#### **Counter\_Load** (DINT)

Set the counter start value after reset.  
Values: 0...2147483646 (default: 0)

#### **Counter\_Max** (DINT)

Set the maximum counter value.  
Values: 0...2147483646 (default: 2147483647)

### Output

#### **Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

### Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- This function block is designed for low-speed counters < 2Hz.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

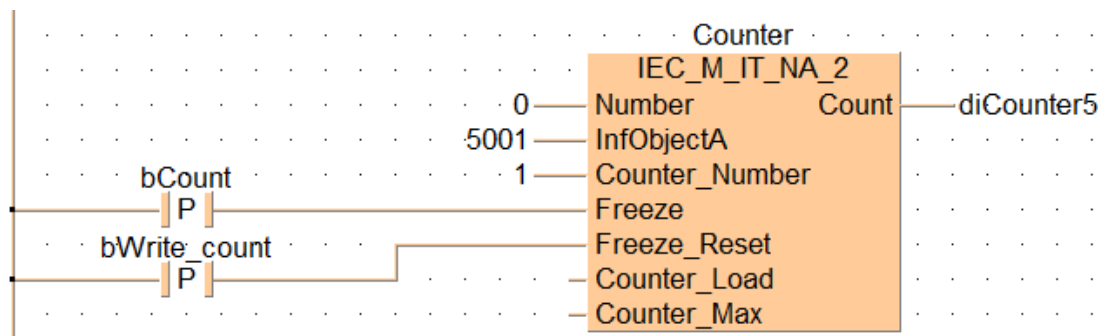
#### 4.2.14 IEC\_M\_IT\_NA\_2

Function block for type identifier 15 to send integrated totals in monitor direction (for high-speed counters).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870**, **IEC60870\_ASDU**, or **IEC60870\_FPG\_EM1**.

The function block can be used with the high-speed counters of the FP0R, FP-X, and FPΣ PLCs. The first four PLC inputs must be specified as interrupt triggers; the program's interrupt tasks require the POU's Counter1 to Counter4. The counter values are stored in the global variable **iEC870\_FCounter[ ]**.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

##### InfObjectA (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Counter\_Number (INT)

Set a unique high-speed counter number.

Values: 1...x (x depends on the size of the global variable **iEC870\_FCounter[ ]**)

**Freeze (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge!

**Freeze\_Reset (BOOL)**

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

**Counter\_Load (DINT)**

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

**Counter\_Max (DINT)**

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count (DINT)**

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.15 IEC\_M\_IT\_NA\_2EM1

---

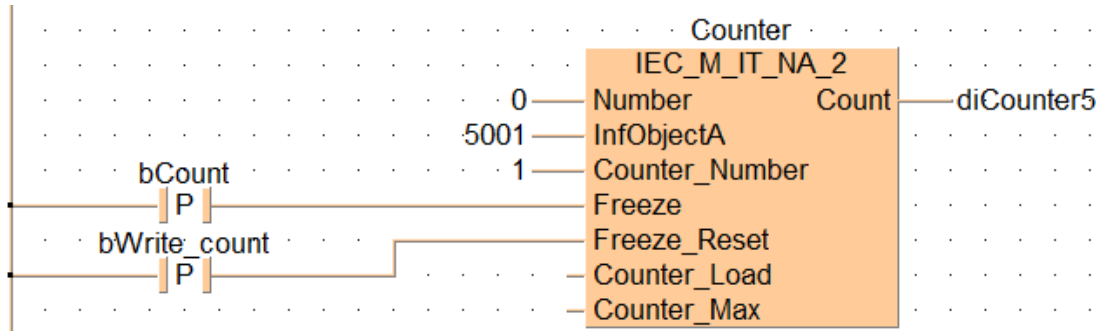
Function block for type identifier 15 to send integrated totals in monitor direction (for high-speed counters, with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with **IEC60870**, **IEC60870\_ASDU**, or **IEC60870\_FPG\_EM1**.

The function block can be used with the high-speed counters of the FP0R, FP-X, and FPΣ PLCs. The first four PLC inputs must be specified as interrupt triggers; the program's interrupt tasks require the POU's Counter1 to Counter4. The counter values are stored in the global variable **IEC870\_FCounter[ ]**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



## Parameters

### Input

#### Number (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

#### InfObjectA (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### Counter\_Number (INT)

Set a unique high-speed counter number.

Values: 1...x (x depends on the size of the global variable **IEC870\_FCounter[ ]**)

#### Freeze (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

#### Freeze\_Reset (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

#### Counter\_Load (DINT)

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)



**Counter\_Max** (DINT)

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[IEC60870\\_EM1](#)

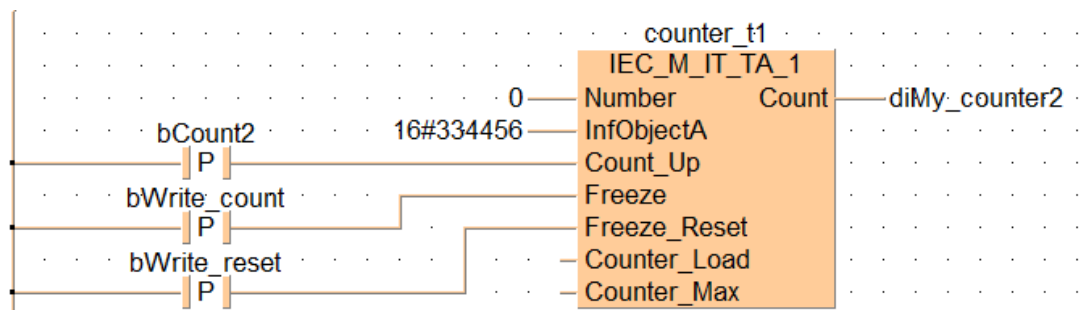
[Glossary of terms](#)

## 4.2.16 IEC\_M\_IT\_TA\_1

Function block for type identifier 16 to send integrated totals with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



### Parameters

## Input

**Number (INT)**

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

**InfObjectA (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Count\_Up (BOOL)**

Turn to TRUE to increment the counter value. Use a rising edge!

**Freeze (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge!

**Freeze\_Reset (BOOL)**

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

**Counter\_Load (DINT)**

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

**Counter\_Max (DINT)**

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count (DINT)**

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- This function block is designed for low-speed counters < 2Hz.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

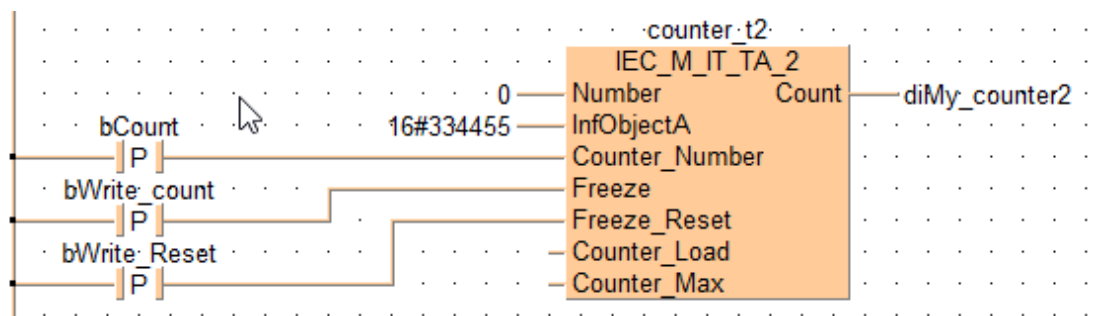
### 4.2.17 IEC\_M\_IT\_TA\_2

Function block for type identifier 16 to send integrated totals with a time tag in monitor direction (for high-speed counters).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870**, **IEC60870\_ASDU**, or **IEC60870\_FPG\_EM1**.

The function block can be used with the high-speed counters of the FP0R, FP-X, and FPΣ PLCs. The first four PLC inputs must be specified as interrupt triggers; the program's interrupt tasks require the POU's Counter1 to Counter4. The counter values are stored in the global variable **iEC870\_FCounter[ ]**.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

##### InfObjectA (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Counter\_Number (INT)

Set a unique high-speed counter number.

Values: 1...x (x depends on the size of the global variable **iEC870\_FCounter[ ]**)

##### Freeze (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

##### Freeze\_Reset (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

**Counter\_Load** (DINT)

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

**Counter\_Max** (DINT)

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[IEC60870\\_EM1](#)

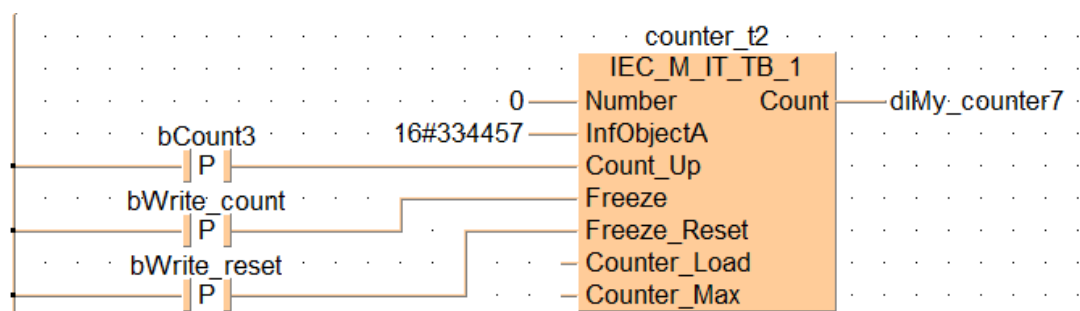
[Glossary of terms](#)

### 4.2.18 IEC\_M\_IT\_TB\_1

Function block for type identifier 37 to send integrated totals with a CP56Time2A time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



## Parameters

### Input

#### **Number** (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

#### **InfObjectA** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### **Count\_Up** (BOOL)

Turn to TRUE to increment the counter value. Use a rising edge!

#### **Freeze** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

#### **Freeze\_Reset** (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

#### **Counter\_Load** (DINT)

Set the counter start value after reset.  
Values: 0...2147483646 (default: 0)

#### **Counter\_Max** (DINT)

Set the maximum counter value.  
Values: 0...2147483646 (default: 2147483647)

### Output

#### **Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

### Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- This function block is designed for low-speed counters < 2Hz.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

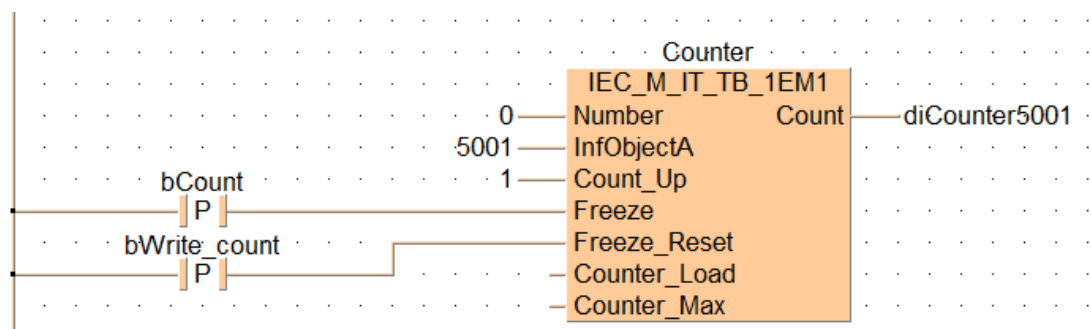
#### 4.2.19 IEC\_M\_IT\_TB\_1EM1

Function block for type identifier 37 to send integrated totals with a CP56Time2A time tag in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



#### Parameters

Input

##### **Number** (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

##### **InfObjectA** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### **Count\_Up** (BOOL)

Turn to TRUE to increment the counter value. Use a rising edge!

**Freeze (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge!

**Freeze\_Reset (BOOL)**

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

**Counter\_Load (DINT)**

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

**Counter\_Max (DINT)**

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

## Output

**Count (DINT)**

Returns the counter value (can be used for diagnosis and further processing).

## Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- This function block is designed for low-speed counters < 2Hz.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.20 IEC\_M\_IT\_TB\_2

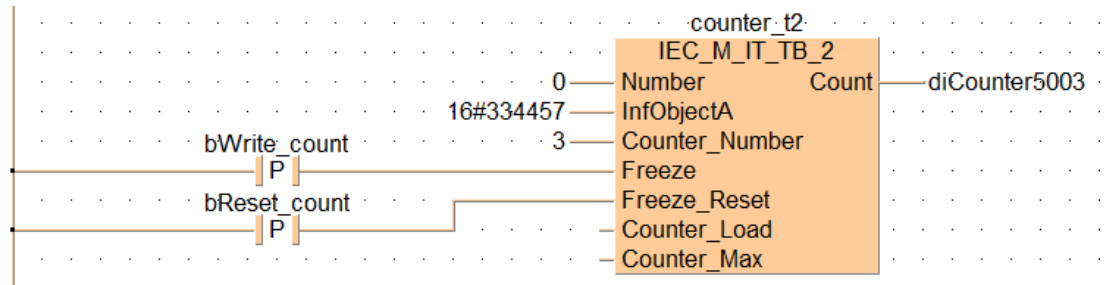
Function block for type identifier 37 to send integrated totals with a CP56Time2A time tag in monitor direction (for high-speed counters).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with **IEC60870**, **IEC60870\_ASDU**, or **IEC60870\_FPG\_EM1**.

The function block can be used with the high-speed counters of the FP0R, FP-X, and FPΣ PLCs. The first four PLC inputs must be specified as interrupt triggers; the program's

interrupt tasks require the POU's Counter1 to Counter4. The counter values are stored in the global variable **IEC870\_FCounter[ ]**.



## Parameters

### Input

#### Number (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

#### InfObjectA (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### Counter\_Number (INT)

Set a unique high-speed counter number.

Values: 1...x (x depends on the size of the global variable **IEC870\_FCounter[ ]**)

#### Freeze (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

#### Freeze\_Reset (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

#### Counter\_Load (DINT)

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

#### Counter\_Max (DINT)

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

### Output



**Count (DINT)**

Returns the counter value (can be used for diagnosis and further processing).

**Note**

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[IEC60870\\_EM1](#)

[Glossary of terms](#)

[Global variables and useful constants](#) (page 10)

**4.2.21 IEC\_M\_IT\_TB\_2EM1**

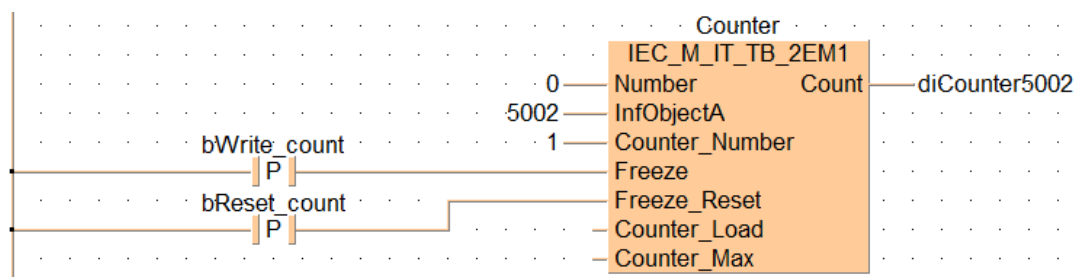
Function block for type identifier 37 to send integrated totals with a CP56Time2A time tag in monitor direction (for high-speed counters, with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[ \]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

The function block can be used with the high-speed counters of the FP0R, FP-X, and FPΣ PLCs. The first four PLC inputs must be specified as interrupt triggers; the program's interrupt tasks require the POU's Counter1 to Counter4. The counter values are stored in the global variable **iEC870\_FCounter[ ]**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.



## Parameters

### Input

#### **Number** (INT)

Set a unique number for this element in the integrated totals database. Use 0 for automatic numbering (recommended).

#### **InfObjectA** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### **Counter\_Number** (INT)

Set a unique high-speed counter number.

Values: 1...x (x depends on the size of the global variable **IEC870\_FCounter[ ]**)

#### **Freeze** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge!

#### **Freeze\_Reset** (BOOL)

Turn to TRUE to immediately create a send telegram and reset the counter value. Use a rising edge!

#### **Counter\_Load** (DINT)

Set the counter start value after reset.

Values: 0...2147483646 (default: 0)

#### **Counter\_Max** (DINT)

Set the maximum counter value.

Values: 0...2147483646 (default: 2147483647)

### Output

#### **Count** (DINT)

Returns the counter value (can be used for diagnosis and further processing).

### Note

- Use a rising edge at the **Freeze** and **Freeze\_Reset** inputs.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

### 4.2.22 IEC\_M\_ME\_NA\_1

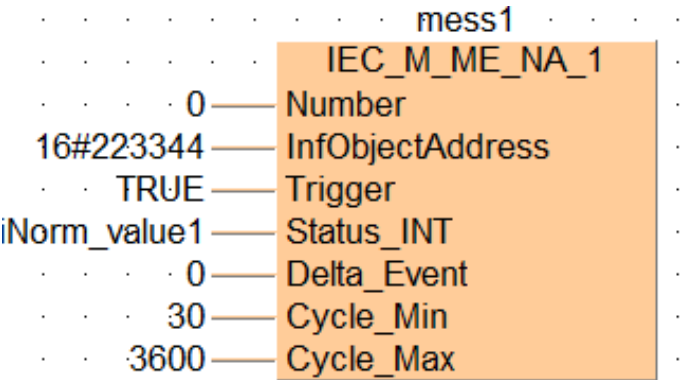
Function block for type identifier 9 to send normalized measurement values in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

##### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### **Status\_INT** (INT)

Measurement value (direct signal from the analog 4...20mA input)

Values: -32768...32767

### **Delta\_Event** (INT)

Set the deviation as a percentage of the last telegram sent.

Values: 1...100%

### **Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

### **Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.23 IEC\_M\_ME\_NA\_EM1

---

Function block for type identifier 9 to send normalized measurement values in monitor direction (with buffer management).

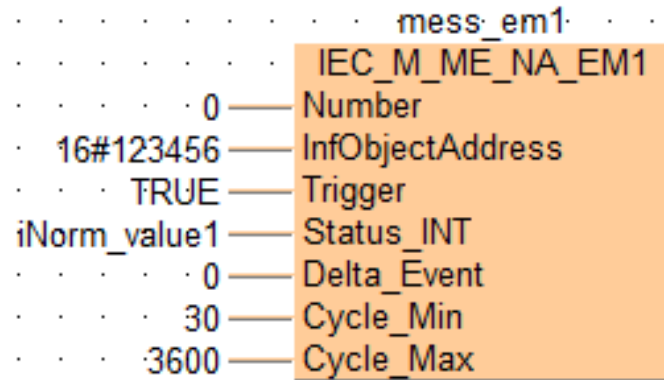
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected.

The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### Status\_INT (INT)

Measurement value (direct signal from the analog 4...20mA input)  
Values: -32768...32767

### Delta\_Event (INT)

Set the deviation as a percentage of the last telegram sent.  
Values: 1...100%

### Cycle\_Min (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

- Values:
- 0...32767s
  - 1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

- [IEC60870\\_EM1](#)
- [Glossary of terms](#)

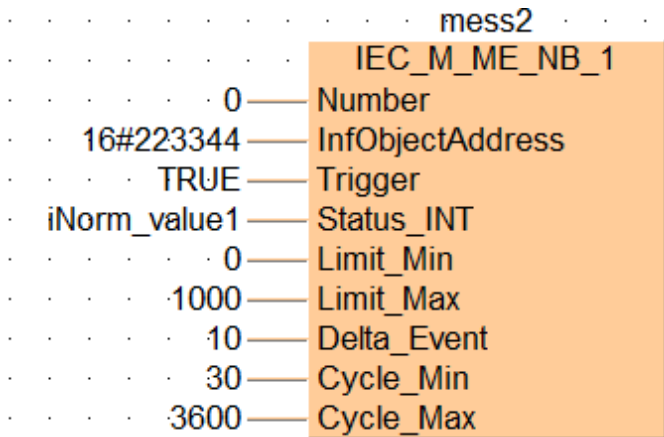
**4.2.24 IEC\_M\_ME\_NB\_1**

Function block for type identifier 11 to send scaled measurement values in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



**Parameters**

Input

**Number (INT)**

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

**InfObjectAddress (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT (INT)**

Scaled measurement value  
Values: -32768...32767

**Limit\_Min (INT)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (INT)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event (INT)**

Set the deviation in digits from the last telegram sent.  
Values: 0...32767

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.  
Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.  
Values:  
0...32767s  
-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

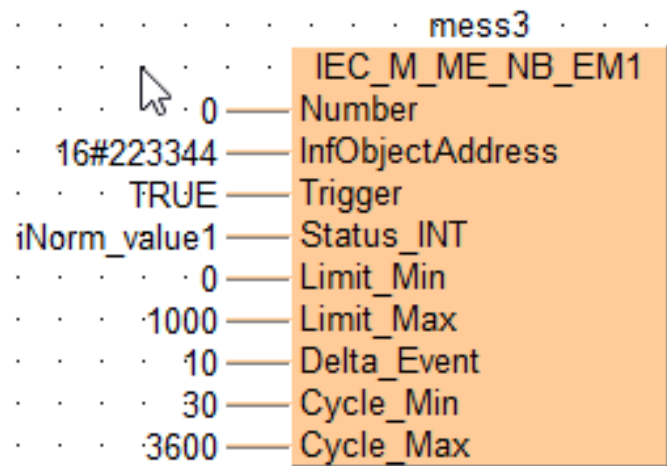
4.2.25 IEC\_M\_ME\_NB\_EM1

Function block for type identifier 11 to send scaled measurement values in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



Parameters

Input

**Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.



**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT (INT)**

Scaled measurement value

Values: -32768...32767

**Limit\_Min (INT)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (INT)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event (INT)**

Set the deviation as a percentage of the last telegram sent.

Values: 1...100%

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870\\_EM1](#)

[Glossary of terms](#)

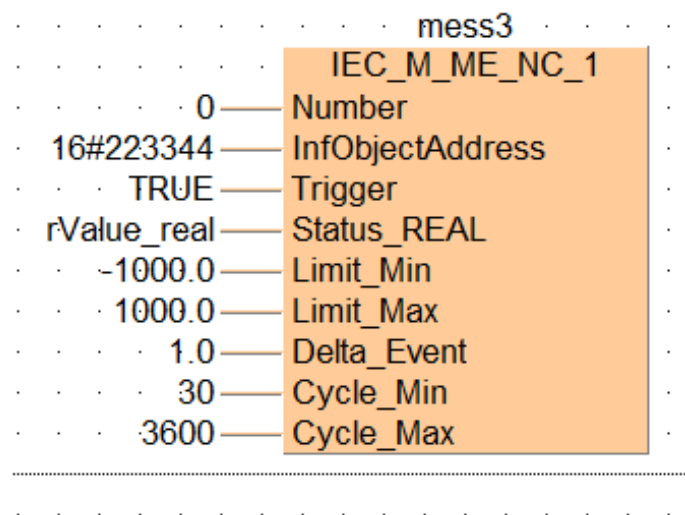
#### 4.2.26 IEC\_M\_ME\_NC\_1

Function block for type identifier 13 to send short floating point values in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

## Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_REAL** (REAL)

Short floating point measurement value

Values: 3.0e38...3.0e38

**Delta\_Event (REAL)**

Set the deviation in digits from the last telegram sent.

Values: 3.0e38...3.0e38

**Limit\_Min (REAL)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (REAL)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

**4.2.27 IEC\_M\_ME\_NC\_EM1**

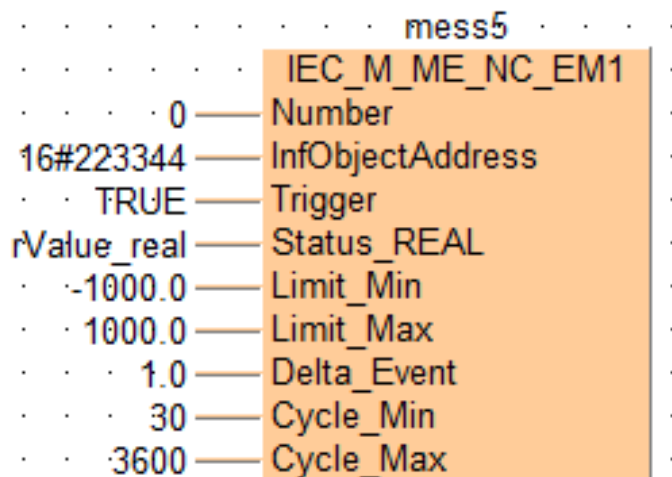
Function block for type identifier 13 to send short floating point values in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected.

The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### Status\_REAL (REAL)

Short floating point measurement value  
Values: 3.0e38...3.0e38

### Delta\_Event (REAL)

Set the deviation in digits from the last telegram sent.  
Values: 3.0e38...3.0e38

### Limit\_Min (REAL)

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max** (REAL)

Set the highest accepted value. Values above this limit will set an invalid flag.

**Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.28 IEC\_M\_ME\_TA\_1

---

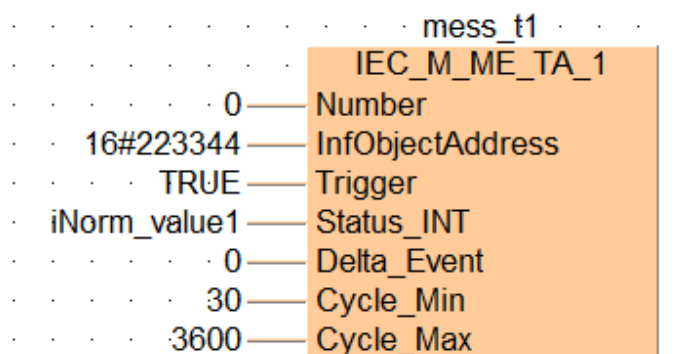
Function block for type identifier 10 to send normalized measurement values with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[Z\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

### Status\_INT (INT)

Measurement value (direct signal from the analog 4...20mA input)

### Delta\_Event (INT)

Set the deviation as a percentage of the last telegram sent.  
Values: 1...100%

### Cycle\_Min (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.  
Values: 0...32767s

### Cycle\_Max (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.  
Values:  
0...32767s

-1: Function disabled

Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

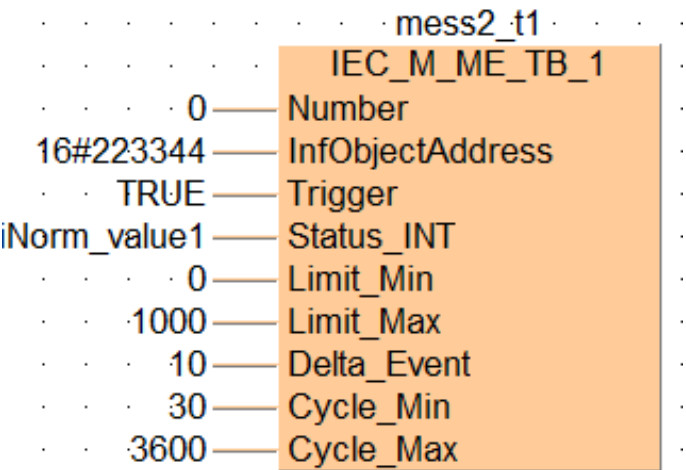
4.2.29 IEC\_M\_ME\_TB\_1

Function block for type identifier 12 to send scaled measurement values with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



Parameters

Input

**Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT** (INT)

Scaled measurement value

Values: -32768...32767

**Limit\_Min** (INT)

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max** (INT)

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event** (INT)

Set the deviation in digits from the last telegram sent.

Values: 0...32767

**Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.



Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

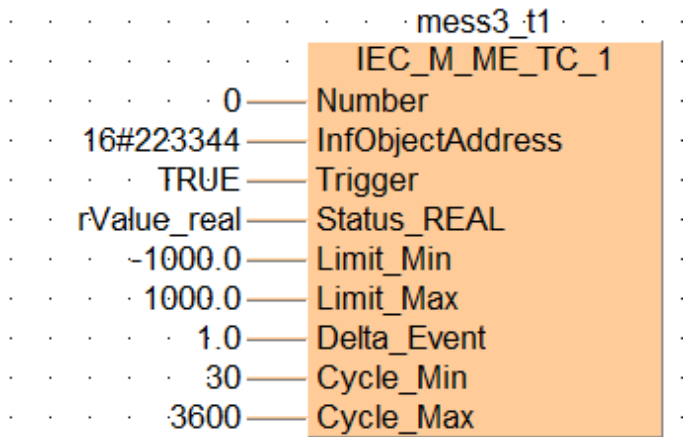
### 4.2.30 IEC\_M\_ME\_TC\_1

Function block for type identifier 14 to send short floating point measurement values with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



#### Parameters

Input

##### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

##### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_REAL** (REAL)

Short floating point measurement value

Values: 3.0e38...3.0e38

**Limit\_Min** (REAL)

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max** (REAL)

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event** (REAL)

Set the deviation in digits from the last telegram sent.

Values: 3.0e38...3.0e38

**Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.2.31 IEC\_M\_ME\_TD\_1

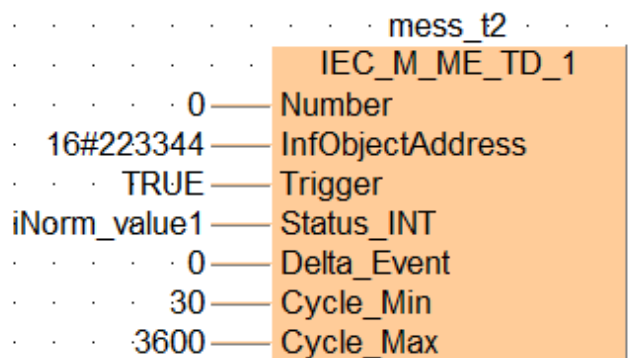
Function block for type identifier 34 to send normalized measurement values with a CP56Time2A time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



#### Parameters

Input

##### **Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

##### **InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### **Trigger** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

##### **Status\_INT** (INT)

Measurement value (direct signal from the analog 4...20mA input)

##### **Delta\_Event** (INT)

Set the deviation as a percentage of the last telegram sent.

Values: 1...100%

### **Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

### **Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.32 IEC\_M\_ME\_TD\_EM1

---

Function block for type identifier 34 to send normalized measurement values with a CP56Time2A time tag in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.

		.mess2log
		IEC_M_ME_TD_EM1
0	—	Number
2001	—	InfObjectAddress
TRUE	—	Trigger
iFP0A80Mess1	—	Status_INT
5	—	Delta_Event
30	—	Cycle_Min
60	—	Cycle_Max

## Parameters

Input

## Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT** (INT)

Measurement value (direct signal from the analog 4...20mA input)  
Values: -32768...32767

## Delta\_Event (INT)

Set the deviation as a percentage of the last telegram sent.  
Values: 1...100%

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

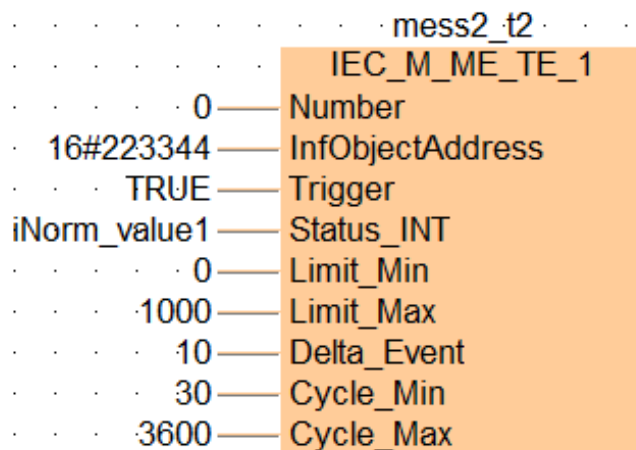
**Related topics**[IEC60870\\_EM1](#)[Glossary of terms](#)**4.2.33 IEC\_M\_ME\_TE\_1**

Function block for type identifier 35 to send scaled measurement values with a CP56Time2A time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.

**Parameters****Input****Number (INT)**

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

**InfObjectAddress (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT (INT)**

Scaled measurement value

Values: -32768...32767

**Limit\_Min (INT)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (INT)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event (INT)**

Set the deviation in digits from the last telegram sent.

Values: 0...32767

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

4.2.34 IEC\_M\_ME\_TE\_EM1

Function block for type identifier 35 to send scaled measurement values with a CP56Time2A time tag in monitor direction (with buffer management).

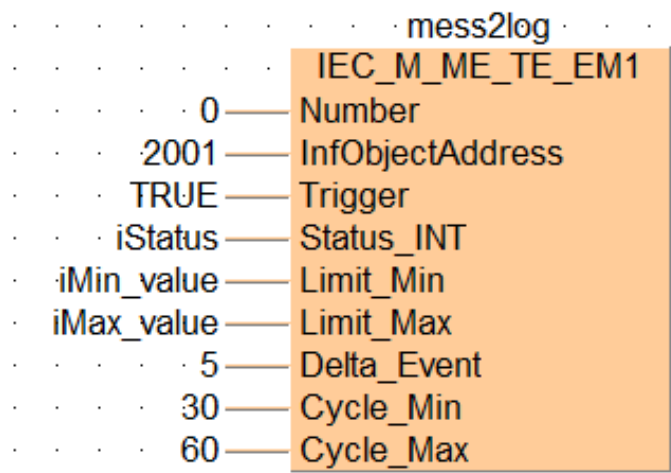
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

The function block normalizes the unscaled measurement values at the analog inputs of the PLCs.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



Parameters

Input

**Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).



**InfObjectAddress (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Trigger (BOOL)**

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_INT (INT)**

Scaled measurement value

Values: -32768...32767

**Limit\_Min (INT)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (INT)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Delta\_Event (INT)**

Set the deviation in digits from the last telegram sent.

Values: 0...32767

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, use the same instance of this function block every time, even though it is programmed to use multiple instances. The programming design differs from that for indexed function block instances.

Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

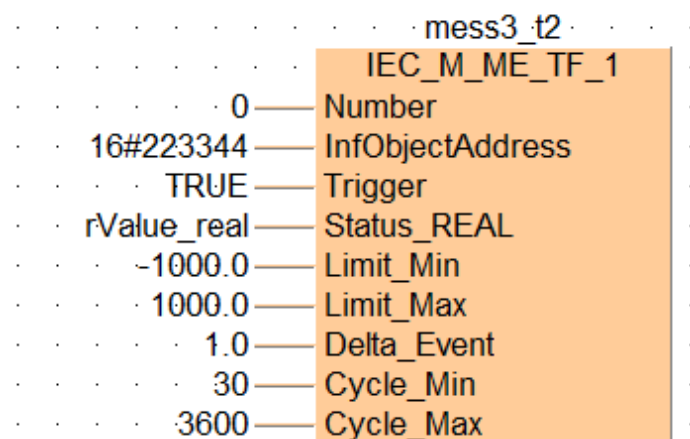
### 4.2.35 IEC\_M\_ME\_TF\_1

Function block for type identifier 36 to send short floating point values with a CP56Time2Ain time tag monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



### Parameters

Input

#### Number (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

#### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### Trigger (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

**Status\_REAL** (REAL)

Short floating point measurement value

Values: 3.0e38...3.0e38

**Delta\_Event** (REAL)

Set the deviation in digits from the last telegram sent.

Values: 3.0e38...3.0e38

**Limit\_Min** (REAL)

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max** (REAL)

Set the highest accepted value. Values above this limit will set an invalid flag.

**Cycle\_Min** (INT)

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max** (INT)

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

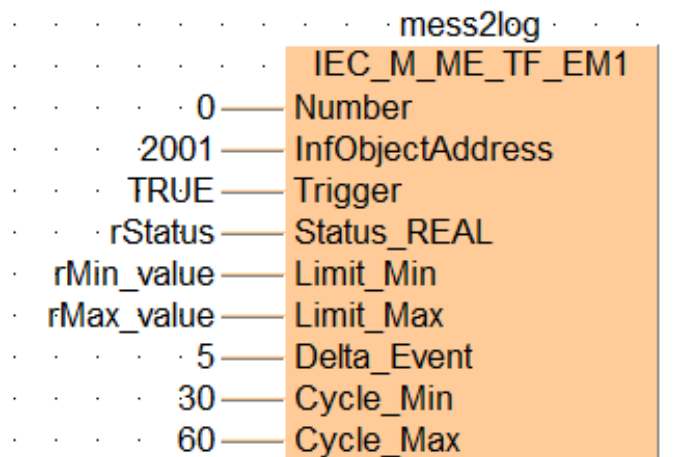
### 4.2.36 IEC\_M\_ME\_TF\_EM1

Function block for type identifier 36 to send short floating point values with a CP56Time2Ain time tag monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Whenever a bit of the input value changes so that it is outside the delta event range, a new message is sent to the IEC 60870 communication unit.



#### Parameters

Input

##### **Number** (INT)

Set a unique number for this element in the 16-/32-bit measurement data database. Use 0 for automatic numbering (recommended).

##### **InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### **Trigger** (BOOL)

Turn to TRUE to immediately create a send telegram. Use a rising edge! This input is not needed in normal operation.

##### **Status\_REAL** (REAL)

Short floating point measurement value

Values: 3.0e38...3.0e38

**Delta\_Event (REAL)**

Set the deviation in digits from the last telegram sent.

Values: 3.0e38...3.0e38

**Limit\_Min (REAL)**

Set the lowest accepted value. Values below this limit will set an invalid flag.

**Limit\_Max (REAL)**

Set the highest accepted value. Values above this limit will set an invalid flag.

**Cycle\_Min (INT)**

Set a time delay to protect the memory buffer and minimize data transfer. If a second change occurs before the time has elapsed, this change is ignored and no telegram is created and sent.

Values: 0...32767s

**Cycle\_Max (INT)**

Set a time value for the send cycle. A telegram is sent after this time, even if no change has occurred.

Values:

0...32767s

-1: Function disabled

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.2.37 IEC\_M\_SP\_NA\_1

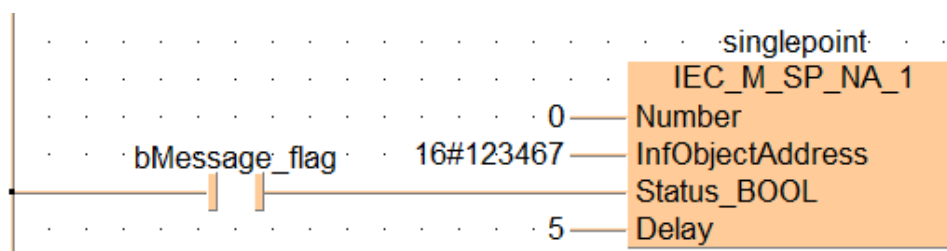
---

Function block for type identifier 1 to send single-point information in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single-point information can assume two states (TRUE or FALSE) and can be used to transmit fault or process conditions, for example.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL (BOOL)

Status signal.

### Delay (INT)

Set a debounce time.

Values: 0...127s (0: no debouncing)

Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.2.38 IEC\_M\_SP\_NA\_EM1

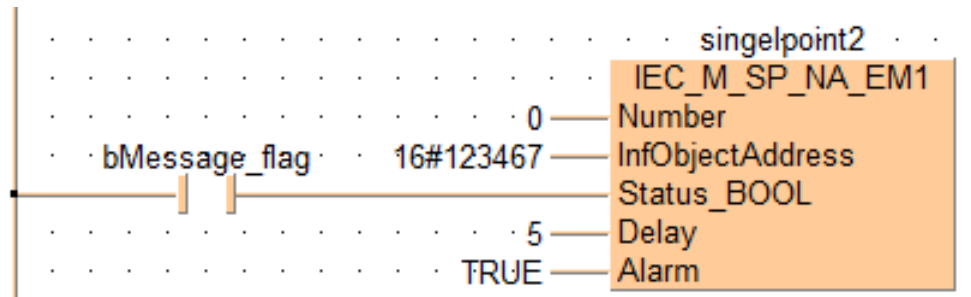
Function block for type identifier 1 to send single-point information in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Use this function block for modem support or if you need a large buffer (FPG-EM1, FP7 or FP0H with DT registers above the address 32767) for times when the unit is disconnected. The function block stores the measurement values in the second data buffer with time tags and sends them to the supervisory control system when the connection has been established.

Single-point information can assume two states (TRUE or FALSE) and can be used to transmit fault or process conditions, for example.



## Parameters

Input

### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### Status\_BOOL (BOOL)

Status signal.

### Delay (INT)

Set a debounce time. Set to zero if not used. Valid values 0...127 seconds.

### Alarm (BOOL)

Set the alarm mode.

Values:

**TRUE**: When a bit at **Status\_BO** changes, a new class 1 message is sent to the supervisory control system, even if a connection has to be established. Use this mode for alarm signals.

**FALSE**: When a bit at **Status\_BO** changes, a new message in the buffer is created, which is not sent immediately but the next time a connection is established. Use this mode for status signals of actuators and switches but not for alarms.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

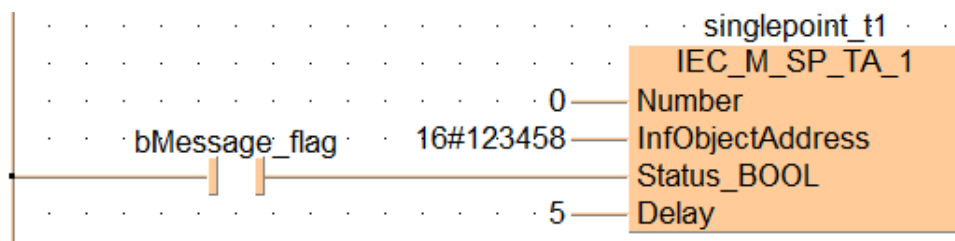
## 4.2.39 IEC\_M\_SP\_TA\_1

Function block for type identifier 2 to send single-point information with a time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single-point information can assume two states (TRUE or FALSE) and can be used to transmit fault or process conditions, for example.



### Parameters

#### Input

##### Number (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

##### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### Status\_BOOL (BOOL)

Status signal.



**Delay (INT)**

Set a debounce time. Set to zero if not used. Valid values 0...127 seconds.

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

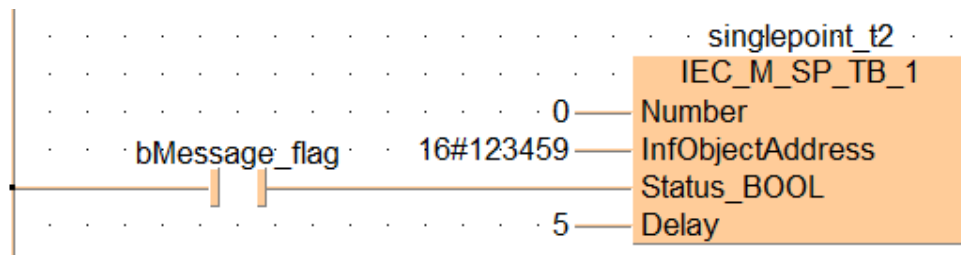
**4.2.40 IEC\_M\_SP\_TB\_1**

Function block for type identifier 30 to send double-point information with a CP56Time2A time tag in monitor direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[ \]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single-point information can assume two states (TRUE or FALSE) and can be used to transmit fault or process conditions, for example.

**Parameters****Input****Number (INT)**

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

**InfoObjectAddress (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Status\_BOOL** (BOOL)

Status signal.

**Delay** (INT)

Set a debounce time. Set to zero if not used. Valid values 0...127 seconds.

Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

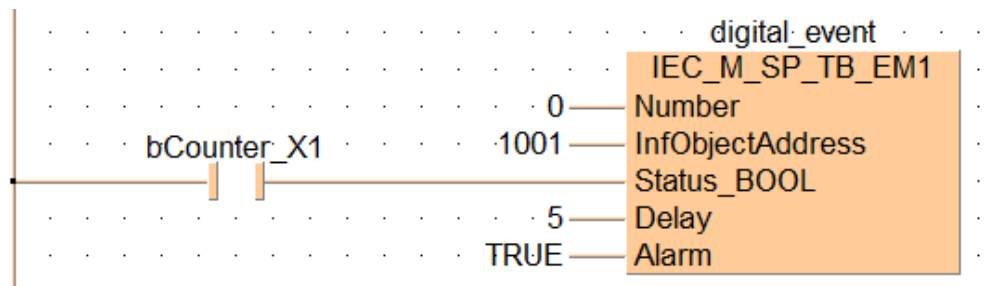
**4.2.41 IEC\_M\_SP\_TB\_EM1**

Function block for type identifier 30 to send single-point information with a CP56Time2A time tag in monitor direction (with buffer management).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with **IEC60870\_FPG\_EM1**.

Single-point information can assume two states (TRUE or FALSE) and can be used to transmit fault or process conditions, for example.



**Parameters**

Input

**Number** (INT)

Set a unique number for this element in the single-/double-point database. Use 0 for automatic numbering (recommended).

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Status\_BOOL** (BOOL)

Status signal.

**Delay** (INT)

Set a debounce time. Set to zero if not used. Valid values 0...127 seconds.

**Alarm** (BOOL)

Set the alarm mode.

Values:

TRUE: When a bit at **Status\_BO** changes, a new class 1 message is sent to the supervisory control system, even if a connection has to be established. Use this mode for alarm signals.

FALSE: When a bit at **Status\_BO** changes, a new message in the buffer is created, which is not sent immediately but the next time a connection is established. Use this mode for status signals of actuators and switches but not for alarms.

## Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870\\_EM1](#)

[Glossary of terms](#)

## 4.3 Process information in control direction

---

These function blocks send control instructions and setpoints from the supervisory control system to the controlled substation (RTU).

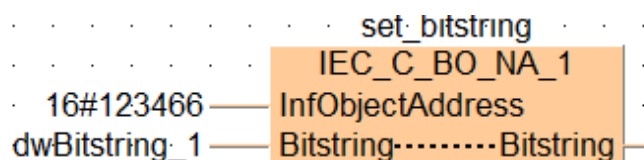
### 4.3.1 IEC\_C\_BO\_NA\_1

---

Function block for type identifier 51 to send 32-bit strings in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



## Parameters

Input

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

Input/output

### Bitstring (DWORD)

The output signal for the user can be a data register (DDT, DFL, %MD5, %MD9).

Output values: 0...4294967295 (i.e. 2E32 -1).

Note

- If the setpoint is to be a static signal, use variables of type VAR\_GLOBAL\_RETAIN with fixed DDT or DFL addresses in the user area. This is necessary so that the setpoints are retained when a cold start, a warm start or a program change is made in the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

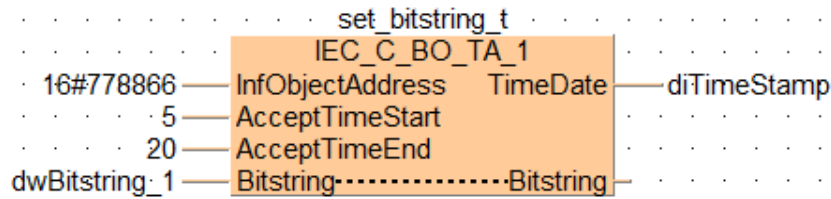
[Glossary of terms](#)

## 4.3.2 IEC\_C\_BO\_TA\_1

Function block for type identifier 64 to send 32-bit strings with a time tag in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



## Parameters

### Input

#### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### AcceptTimeStart (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

#### AcceptTimeEnd (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

### Input/output

#### Bitstring (DWORD)

The output signal for the user can be a data register (DDT, DFL, %MD5, %MD9).

Output values: 0...4294967295 (i.e. 2E32 -1).

### Note

- If the setpoint is to be a static signal, use variables of type VAR\_GLOBAL\_RETAIN with fixed DDT or DFL addresses in the user area. This is necessary so that the setpoints are retained when a cold start, a warm start or a program change is made in the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.3.3 IEC\_C\_DC\_NA\_1

Function block for type identifier 46 to send double commands in control direction.

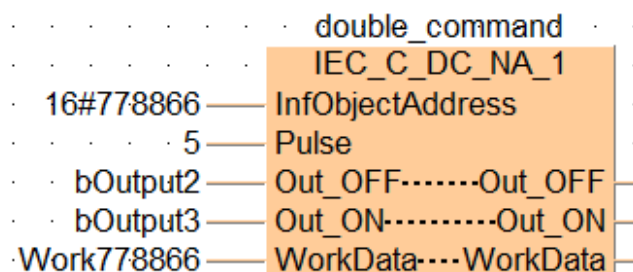
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double commands have two bits and can be used to control devices that have two stable states, such as switches or valves, more reliably than single commands.

1. A command is sent if **Out\_OFF** is TRUE.
2. A command is sent if **Out\_ON** is TRUE.
3. No command is sent if **Out\_OFF** and **Out\_ON** are FALSE.
4. No command is sent if **Out\_OFF** and **Out\_ON** are TRUE (invalid status).

The function block outputs the double commands as static or pulse signals, depending on the quality information from the controlling station.



#### Parameters

Input

##### **InfoObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### **Pulse** (INT)

Set the pulse duration for pulse commands.

Values: 0...32767s

Input/output

**Out\_OFF** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**Out\_ON** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

## Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

#### 4.3.4 IEC\_C\_DC\_NA\_2

---

Function block for type identifier 46 to send double commands with a time tag in control direction (short and long pulse inputs).

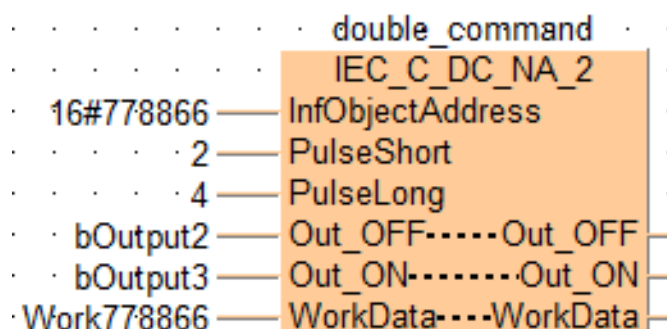
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double commands have two bits and can be used to control devices that have two stable states, such as switches or valves, more reliably than single commands.

1. A command is sent if **Out\_OFF** is TRUE.
2. A command is sent if **Out\_ON** is TRUE.
3. No command is sent if **Out\_OFF** and **Out\_ON** are FALSE.
4. No command is sent if **Out\_OFF** and **Out\_ON** are TRUE (invalid status).

The function block outputs the double commands as static or pulse signals, depending on the quality information from the controlling station.



## Parameters

### Input

#### **InfoObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### **PulseShort** (INT)

Set the duration of a short output pulse.

Values: 1..32767s

#### **PulseLong** (INT)

Set the duration of a long output pulse.

Values: 1..32767s

### Input/output

#### **Out\_OFF** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

#### **Out\_ON** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

#### **WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.



Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.3.5 IEC\_C\_DC\_TA\_1

Function block for type identifier 59 to send double commands with a time tag in control direction.

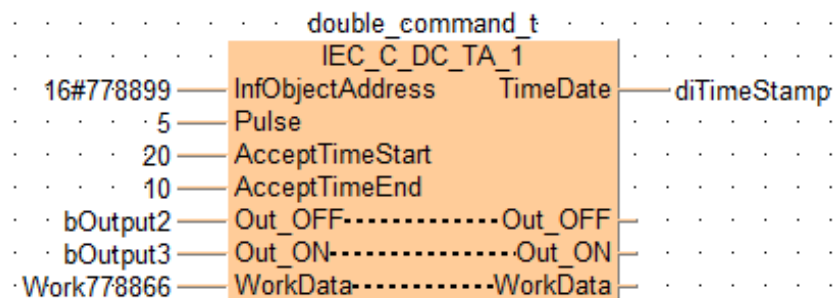
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double commands have two bits and can be used to control devices that have two stable states, such as switches or valves, more reliably than single commands.

1. A command is sent if **Out\_OFF** is TRUE.
2. A command is sent if **Out\_ON** is TRUE.
3. No command is sent if **Out\_OFF** and **Out\_ON** are FALSE.
4. No command is sent if **Out\_OFF** and **Out\_ON** are TRUE (invalid status).

The function block outputs the double commands as static or pulse signals, depending on the quality information from the controlling station.



#### Parameters

Input

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**Pulse** (INT)

Set the pulse duration for pulse commands.

Values: 0...32767s

**AcceptTimeStart** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

**AcceptTimeEnd** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

## Input/output

**Out\_OFF** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**Out\_ON** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

## Output

**TimeDate** (DINT)

Returns a time stamp from the supervisory control system.

## Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.3.6 IEC\_C\_DC\_TA\_2

Function block for type identifier 59 to send double commands with a time tag in control direction (with short and long pulse inputs).

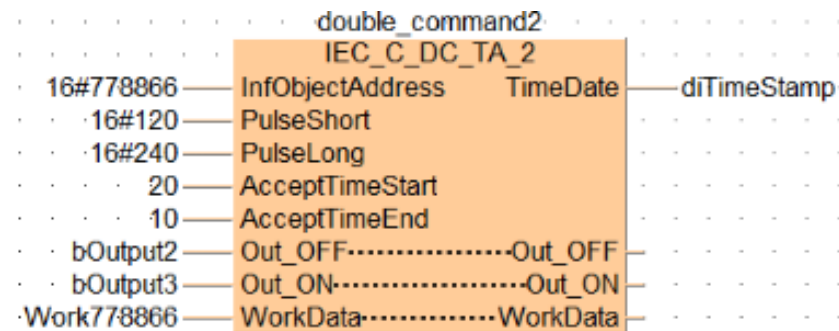
This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Double commands have two bits and can be used to control devices that have two stable states, such as switches or valves, more reliably than single commands.

1. A command is sent if **Out\_OFF** is TRUE.
2. A command is sent if **Out\_ON** is TRUE.
3. No command is sent if **Out\_OFF** and **Out\_ON** are FALSE.
4. No command is sent if **Out\_OFF** and **Out\_ON** are TRUE (invalid status).

The function block outputs the double commands as static or pulse signals, depending on the quality information from the controlling station.



#### Parameters

##### Input

##### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### PulseShort (INT)

Set the duration of a short output pulse.

##### PulseLong (INT)

Set the duration of a long output pulse.

##### AcceptTimeStart (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time

frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

### **AcceptTimeEnd** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Values: 0...32767s

### Input/output

### **Out\_OFF** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

### **Out\_ON** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

### **WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

### Output

### **TimeDate** (DINT)

Returns a time stamp from the supervisory control system.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

---

## 4.3.7 IEC\_C\_SC\_NA\_1

---

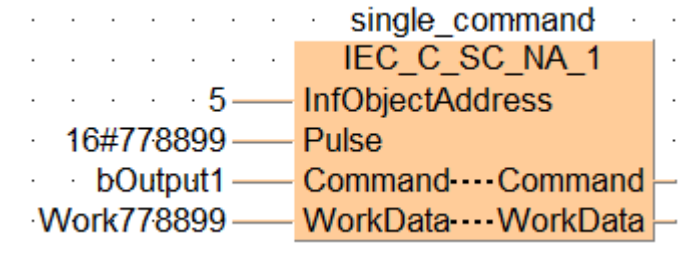
Function block for type identifier 45 to send single commands in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single commands use one bit to switch binary devices on and off.

The function block outputs the single commands as static or pulse signals, depending on the quality information from the controlling station.



## Parameters

### Input

#### **InfoObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

#### **Pulse** (INT)

Set the pulse duration for pulse commands.

Values: 0...32767s

### Input/output

#### **Command** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

#### **WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.3.8 IEC\_C\_SC\_NA\_2

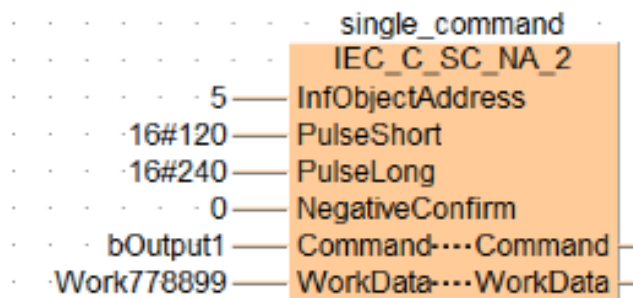
Function block for type identifier 45 to send single commands with a time tag in control direction (short and long pulse inputs).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single commands use one bit to switch binary devices on and off.

The function block outputs the single commands as static or pulse signals, depending on the quality information from the controlling station.



#### Parameters

##### Input

##### **InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### **PulseShort** (INT)

Set the duration of a short output pulse.

Values: 1..32767s

##### **PulseLong** (INT)

Set the duration of a long output pulse.

Values: 1..32767s

##### **NegativeConfirm** (BOOL)

Turn to TRUE to send a negative confirmation.

##### Input/output

##### **Command** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**WorkData (DUT)**

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

**Note**

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

**Related topics**

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

### 4.3.9 IEC\_C\_SC\_TA\_1

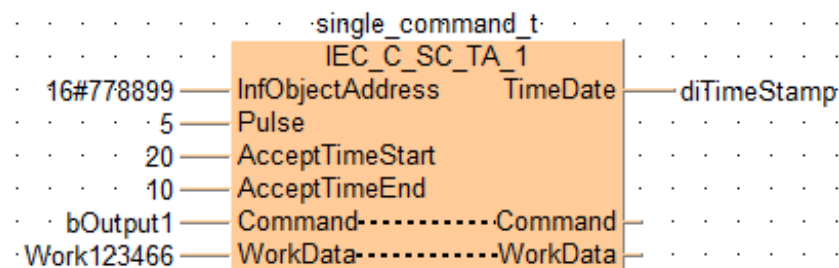
Function block for type identifier 58 to send single commands with a CP56Time2A time tag in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single commands use one bit to switch binary devices on and off.

The function block outputs the single commands as static or pulse signals, depending on the quality information from the controlling station.

**Parameters****Input****InfObjectAddress (DINT)**

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

### **Pulse** (INT)

Set the pulse duration for pulse commands.

Values: 0...32767s

### **AcceptTimeStart** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

### **AcceptTimeEnd** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

### Input/output

#### **Command** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

#### **WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

### Output

#### **TimeDate** (DINT)

Returns a time stamp from the supervisory control system.

### Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)



### 4.3.10 IEC\_C\_SC\_TA\_2

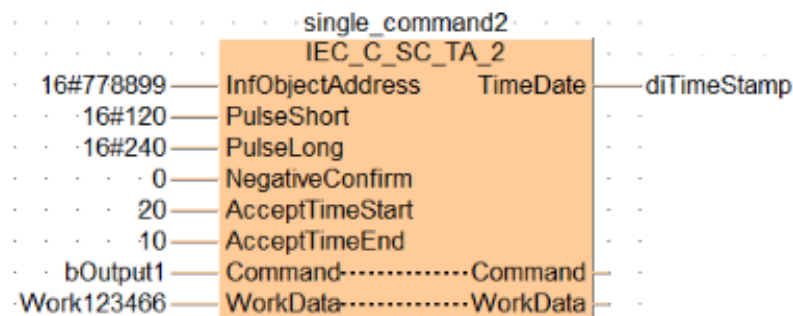
Function block for type identifier 58 to send single commands with a CP56Time2A time tag in control direction (short and long pulse inputs, negative confirmation input).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

Single commands use one bit to switch binary devices on and off.

The function block outputs the single commands as static or pulse signals, depending on the quality information from the controlling station.



#### Parameters

Input

##### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### PulseShort (INT)

Set the duration of a short output pulse.

##### PulseLong (INT)

Set the duration of a long output pulse.

##### NegativeConfirm (BOOL)

Turn to TRUE to send a negative confirmation.

##### AcceptTimeStart (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

**AcceptTimeEnd** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Input/output

**Command** (BOOL)

Turns an output (Y, %QX) or a flag (R, %MX) to TRUE or to FALSE.

**WorkData** (DUT)

Set the DUT **IEC60870\_C\_inst** specifying important temporary values for the function block. The variable name must be unique for each command.

Output

**TimeDate** (DINT)

Returns a time stamp from the supervisory control system.

## Note

- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

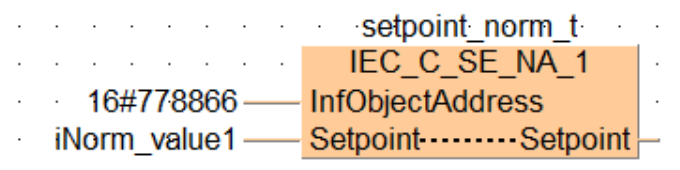
[Glossary of terms](#)

### 4.3.11 IEC\_C\_SE\_NA\_1

Function block for type identifier 48 to send normalized setpoint values in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



## Parameters

Input

### InfObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

Input/output

### Setpoint (INT)

The output signal for the user can be an external analog output (WY, %QW) or data register (DT, FL, %MW5, %MW9). The output signal is not normalized.

Values: -32768...32767s

## Note

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

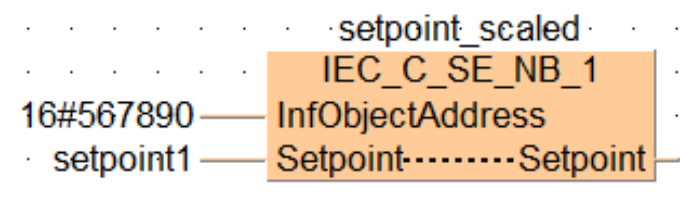
[Glossary of terms](#)

### 4.3.12 IEC\_C\_SE\_NB\_1

Function block for type identifier 49 to send scaled setpoint values in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



## Parameters

Input

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be set in hex format, e.g. 16#123456, or as a decimal value.

Input/output

**Setpoint** (INT)

The output signal for the user can be an external analog output (WY, %QW) or data register (DT, FL, %MW5, %MW9). The output signal is not scaled.

Values: -32768...32767s

**Note**

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

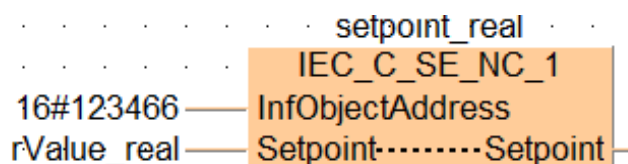
[Glossary of terms](#)

**4.3.13 IEC\_C\_SE\_NC\_1**

Function block for type identifier 50 to send short floating point setpoint values in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[?\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

**Parameters**

Input

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

Input/output

**Setpoint** (REAL)

The output signal for the user can be a data register (DDT, FL, %MW5, %MW9).

Values: -3.0e38... 3.0e38

**Note**

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

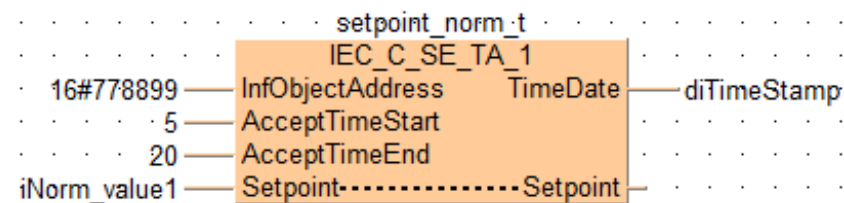
[Glossary of terms](#)

#### 4.3.14 IEC\_C\_SE\_TA\_1

Function block for type identifier 61 to send normalized setpoint values with a time tag in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.

**Parameters**

Input

**InfObjectAddress** (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

**AcceptTimeStart** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

**AcceptTimeEnd** (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

## Input/output

**Setpoint** (INT)

The output signal for the user can be an external analog output (WY, %QW) or data register (DT, FL, %MW5, %MW9). The output signal is not normalized.

Values: -32768...32767s

## Output

**TimeDate** (DINT)

Returns a time stamp from the supervisory control system.

## Note

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

## Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

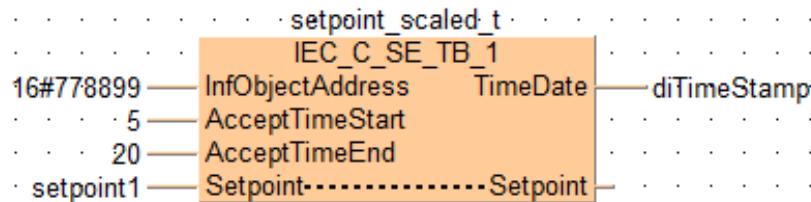
[Glossary of terms](#)

### 4.3.15 IEC\_C\_SE\_TB\_1

Function block for type identifier 62 to send scaled setpoint values a time tag in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



#### Parameters

Input

##### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be specified in hex format, e.g. 16#123456, or as a decimal value.

##### AcceptTimeStart (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

##### AcceptTimeEnd (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Input/output

##### Setpoint (INT)

The output signal for the user can be an external analog output (WY, %QW) or data register (DT, FL, %MW5, %MW9). The output signal is not scaled.

Values: -32768...32767s

Output

##### TimeDate (DINT)

Returns a time stamp from the supervisory control system.

### Note

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

### Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

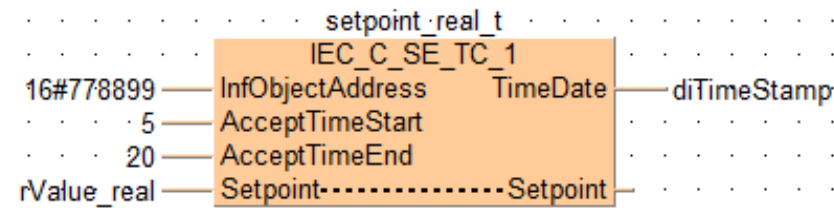
[Glossary of terms](#)

## 4.3.16 IEC\_C\_SE\_TC\_1

Function block for type identifier 63 to send short floating point setpoint values in control direction.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

You must always start your program with the function block **IEC60870** or **IEC60870\_ASDU**.



### Parameters

#### Input

##### InfoObjectAddress (DINT)

Set the unique information object address specified in the supervisory control system. The address can be set in hex format, e.g. 16#123456, or as a decimal value.

##### AcceptTimeStart (INT)

Set the lifetime of the command. The lifetime is the time between *PLC real time - AcceptTimeStart* and *PLC real time + AcceptTimeEnd*. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.



**AcceptTimeEnd (INT)**

Set the lifetime of the command. The lifetime is the time between *PLC real time* - **AcceptTimeStart** and *PLC real time* + **AcceptTimeEnd**. Commands outside of this time frame are not accepted. There may be a lifetime error if the time is not synchronized correctly.

Input/output

**Setpoint (REAL)**

The output signal for the user can be a data register (DDT, FL, %MW5, %MW9).  
Values: -3.0e38... 3.0e38

Output

**TimeDate (DINT)**

Returns a time stamp from the supervisory control system.

**Note**

- If the setpoint value should be a static signal, use VAR\_GLOBAL\_RETAIN variables with fixed DT or FL addresses in the user area. This is necessary for the system to hold the setpoint values during a cold start, warm start and for program updates of the PLC.
- To save memory, always use the same instance of this function block. All temporary information is stored in a linked database.

Related topics

[IEC60870](#)

[IEC60870\\_ASDU](#)

[Glossary of terms](#)

## 4.4 Other useful function blocks

---

The library contains some additional function blocks, e.g. for accessing analog units or for handling daylight saving time.

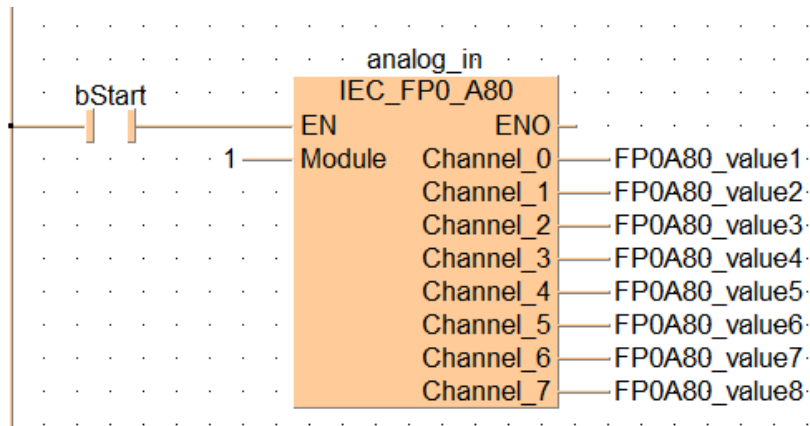
### 4.4.1 IEC\_FP0\_A80

---

Function block to read from an FP0-A80 unit.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).

The function block gets all measurement values from the analog module.



## Parameters

Input

### EN (BOOL)

Turn to TRUE to start the operation.

### Module (INT)

Set the slot number of the analog unit.

Values: 1...3

Output

### Channel0 to Channel7 (INT)

Returns the converted digital data from the analog unit by channel.

Values: 0 ... +4000

### Note

- The channel is configured using the DIP switches on the unit. For details, refer to the manual.
- This function block does not support multiple instances. For each analog unit, a separate instance name is required.
- A comprehensive set of system instructions for analog units can be found in the "Input, output, and unit access instructions" section.

### Related topics

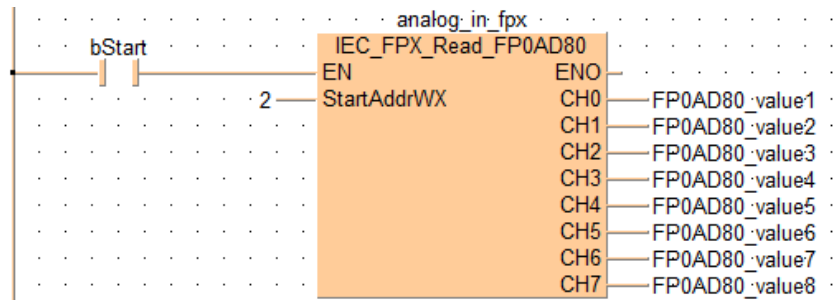
[Input, output, and unit access instructions](#) 

### 4.4.2 IEC\_FPX\_ReadAD80

Function block to read from an FP0-A80 unit (when using an adapter between CPU and analog unit).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

The function block gets all measurement values from the analog module.



#### Parameters

##### Input

##### EN (BOOL)

Turn to TRUE to start the operation.

##### Module (INT)

Set the slot number of the analog unit. The offset depends on the adapter position and the unit position relative to the adapter.

Values: 30...104

##### Output

##### Channel0 to Channel7 (INT)

Returns the converted digital data from the analog unit by channel.

Values: 0 ... +4000

##### Note


- The channel is configured using the DIP switches on the unit. For details, refer to the manual.
- This function block does not support multiple instances. For each analog unit, a separate instance name is required.
- A comprehensive set of system instructions for analog units can be found in the "Input, output, and unit access instructions" section.

Related topics

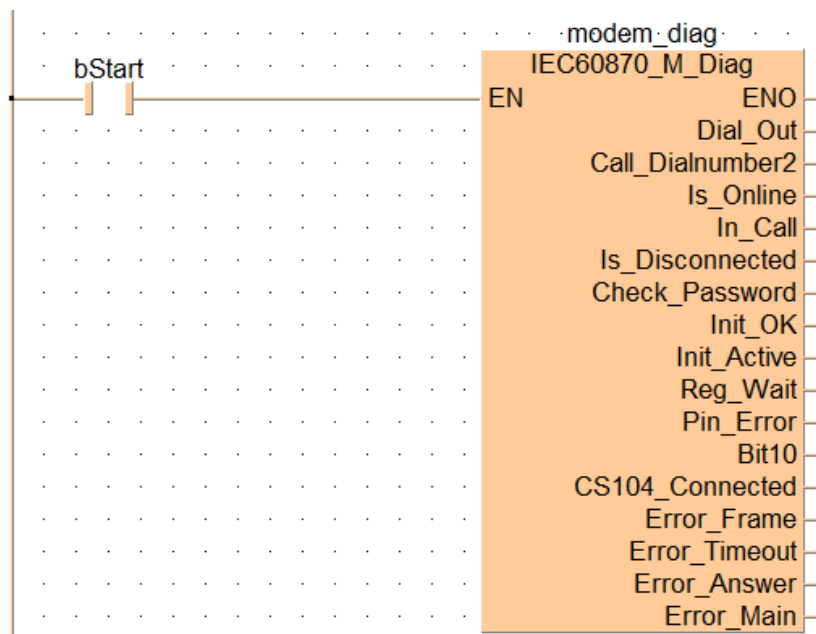
[Input, output, and unit access instructions](#) 

### 4.4.3 IEC60870\_M\_Diag

Function block to return modem status information.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library .

Modem communication is handled by the IEC 60870 communication unit, not by the PLC. If a modem is connected to the communication unit, this is the only way to obtain status information.



#### Parameters

Input

**EN** (BOOL)

Activates the function block.

Output

**Dial\_Out** (BOOL)

TRUE if the IEC 60870 communication unit tries to call the main station.

**Call\_Dialnumber2** (BOOL)

TRUE if the second phone number is used for dialing out.

**Is\_Online** (BOOL)

TRUE if the modem connection has been established.

**In\_Call** (BOOL)

TRUE if an incoming call has been detected.

**Is\_Disconnected** (BOOL)

TRUE after a successful disconnection.

**Check\_Password** (BOOL)

TRUE if password checking is active.

**Init\_OK** (BOOL)

TRUE if the modem has been initialized.

**Init\_Active** (BOOL)

TRUE if the mode is being initialized.

**Reg\_Wait** (BOOL)

TRUE if the radio modem is searching for the network (GSM only).

**Pin\_Error** (BOOL)

TRUE if the SIM card is not installed or installed incorrectly.

**Bit10** (BOOL)

Not used.

**CS104\_Connected** (BOOL)

TRUE if an IEC 60870-5-104 connection has been established.

**Error\_Frame** (BOOL)

TRUE if there is data transfer error.

**Error\_Timeout** (BOOL)

TRUE if there is data transfer error, no answer.

**Error\_Answer** (BOOL)

TRUE if there is data transfer error, wrong answer.

**Error\_Main** (BOOL)

TRUE if there is any other error.

#### 4.4.4 IEC60870\_RTC\_DST

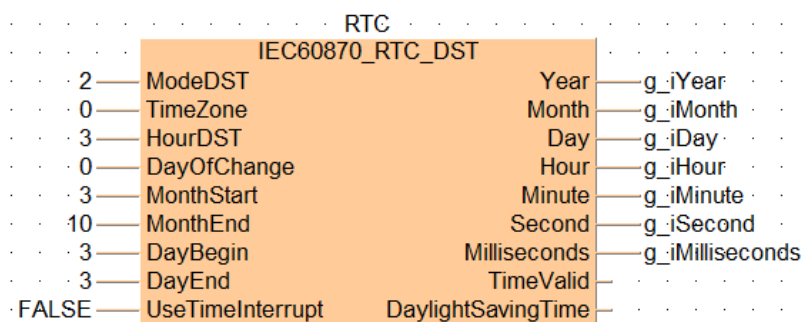
Function block to control the real-time clock of the PLC and the internal UNIX timestamp (with daylight saving time function).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

This function block or **IEC60870\_RTC\_NoDST** should be used in all IEC60870 applications and should be located in the network after the IEC60870 main function block.

If the real-time clock of the PLC is not working well, the battery is empty or the RTC unit is not installed, this function block will use an internal time stamp. When the internal UNIX time is used (32-bit DINT value), the function block can output all time and date information.

This function block allows the use of daylight saving time rules, which will update the PLC time according to time zone and season.



### Parameters

#### Input

##### ModeDST (INT)

Set the daylight saving time (DST) mechanism.

Values:

- 0: No offset and no daylight saving time calculation
- 1: PLC real-time clock set by NTP with UTC, PLC calculates DST offset
- 2: PLC real-time clock set by NTP with DST and automatically by PLC

Range: 0 to 2 (default: 0)

##### TimeZone (INT)

Set the difference time to Greenwich when using UTC time.

Range: 0 to 2 (default: 0)

For Amsterdam, Berlin, Rome: +1

##### HourDST (INT)

Set the time when the DST changes.

Range: 0–23 (default: 3 = 3:00 o'clock)

Example: Berlin set to **3**

Example: Berlin set to **3**

Last Sunday in March: 2:00 --> **3:00**

Last Sunday in October: **3:00** --> 2:00

### **DayOfChange** (INT)

Set the day of the week.

Values:

0: Sunday

1: Monday

2: ...

Range: 0–6 (default: 0 = Sunday)

### **MonthStart** (INT)

Set the month for the changeover from normal time to DST.

Range: 1 to 12 (default: 3 = March)

For Berlin: March=3

### **MonthEnd** (INT)

Set the month for the changeover from DST to normal time.

Range: 1–12 (default: 10 = October)

For Berlin: October=10

### **DayBegin** (INT)

Set the week for the changeover from normal time to DST.

Values:

0: First

1: Second

2: Last but one

3: Last

Range: 0 to 3 (default: 3 = last)

For Berlin: Last week = 3

### **DayEnd** (INT)

Set the month for the changeover from normal time to DST.

Values:

0: First

1: Second

2: Last but one

3: Last

Range: 0 to 3 (default: 3 = last)

For Berlin: Last week = 3

### **UseTimeInterrupt** (BOOL)

Turn to TRUE if an external interrupt-based high-accurate timer is used.

Default: FALSE

### Output

#### **Year** (INT)

Returns the year.

#### **Month** (INT)

Returns the month.

#### **Day** (INT)

Returns the day.

#### **Hour** (INT)

Returns the hour.

#### **Minute** (INT)

Returns the minute.

#### **Second** (INT)

Returns the second.

#### **Millisecond** (INT)

Returns the millisecond.

### **TimeValid** (BOOL)

TRUE if the real-time clock and the internal UNIX timestamp are valid.

### **DaylightSavingTime** (BOOL)

TRUE if the daylight saving time mechanism is used (default: TRUE).

Values:

TRUE: Summer

FALSE: Winter

---

## 4.4.5 IEC60870\_RTC\_NoDST

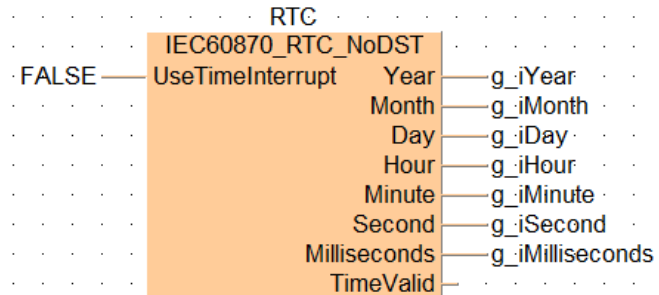
Function block to control the real-time clock of the PLC and the internal UNIX timestamp (without daylight saving time function).

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [🔗](#).



This function block or **IEC60870\_RTC\_DST** should be used in all IEC60870 applications and should be located in the network after the IEC60870 main function block.

If the real-time clock of the PLC is not working well, the battery is empty or the RTC unit is not installed, this function block will use an internal time stamp. When the internal UNIX time is used (32-bit DINT value), the function block can output all time and date information.



## Parameters

### Input

#### **UseTimeInterrupt** (BOOL)

Turn to TRUE if an external interrupt-based high-accurate timer is used.

Default: FALSE

### Output

#### **Year** (INT)

Returns the year.

#### **Month** (INT)

Returns the month.

#### **Day** (INT)

Returns the day.

#### **Hour** (INT)

Returns the hour.

#### **Minute** (INT)

Returns the minute.

#### **Second** (INT)

Returns the second.

#### **Millisecond** (INT)

Returns the millisecond.

**TimeValid (BOOL)**

TRUE if the real-time clock and the internal UNIX timestamp are valid.

**DaylightSavingTime (BOOL)**

TRUE if the daylight saving time mechanism is used (default: TRUE).

Values:

TRUE: Summer

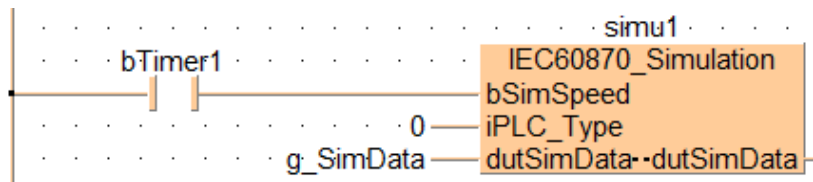
FALSE: Winter

#### 4.4.6 IEC60870\_Simulation

Function block to simulate measurement values.

This function block is available in the M\_IEC60870\_PRO7 V.3.2.2.0 library [\[link\]](#).

This function block generates simulation values for a first test of the IEC60870 communication.



### Parameters

Input

**bSimSpeed (BOOL)**

Turn to TRUE to generate the data.

**iPLC\_Type (INT)**

Set the PLC type.

Values:

0: FP0R, FP-X, FPΣ

2: FP2, FP2SH

7: FP7 (no internal measurement scaling)

Input/output

**dutSimData (DUT)**

Set the DUT specifying the measurement values.


## 5 Panasonic hotline

---

If you have questions that cannot be clarified by the manuals or online help, please contact one of our sales offices.

You can help us by having the following data at hand:

- Your product's serial number and/or version number.
- The version and service pack numbers of MS-Windows installed on your computer.
- The type of hardware you are using.
- The exact wording of any message that appears on your screen.
- What happened and what did you do when the problem occurred?
- How did you attempt to solve the problem?

For inquiries outside of Europe, please visit our [global Web site](#) .

### Panasonic Industry Europe GmbH

- Germany and European countries not listed on this page:  
+49 89 45354-2748, [plc.pieu@eu.panasonic.com](mailto:plc.pieu@eu.panasonic.com) (PLC, FP-I4C, touch panels)  
+49 89 45354-2737, [sensors.pieu@eu.panasonic.com](mailto:sensors.pieu@eu.panasonic.com) (sensors)  
+49 89 45354-2750, [servo.pieu@eu.panasonic.com](mailto:servo.pieu@eu.panasonic.com) (servo drives)
- France:  
+33 160 135757, [pieufr\\_support@eu.panasonic.com](mailto:pieufr_support@eu.panasonic.com)

### Panasonic Industry Austria GmbH

Austria, Bosnia and Herzegovina, Bulgaria, Croatia, Montenegro, Serbia, Slovenia, Switzerland:

+43 2236 26846, [info.at@eu.panasonic.com](mailto:info.at@eu.panasonic.com)

### Panasonic Industry Benelux B.V.

Belgium, Denmark, Luxembourg, Norway, Sweden, The Netherlands:

+31 499 372727, [sales.pibnl@eu.panasonic.com](mailto:sales.pibnl@eu.panasonic.com)

### Panasonic Industry Italia srl

Italy:

+39 045 6752711, [support.piit@eu.panasonic.com](mailto:support.piit@eu.panasonic.com)

### **Panasonic Industry Poland sp. z o.o.**

Baltic states, Czech Republic, Finland, Hungary, Poland, Romania, Slovakia:

+48 42 2309633, [tech.pipl@eu.panasonic.com](mailto:tech.pipl@eu.panasonic.com) (customers in Poland) or  
[tech.cee@eu.panasonic.com](mailto:tech.cee@eu.panasonic.com) (other countries)

### **Panasonic Industry Iberia S.A.**

Portugal, Spain:

+34 91 3293875, [soporte.tecnico@eu.panasonic.com](mailto:soporte.tecnico@eu.panasonic.com)

### **Panasonic Industry UK Ltd.**

United Kingdom of Great Britain and Ireland:

+44 1908 231555, [info.piuk@eu.panasonic.com](mailto:info.piuk@eu.panasonic.com)