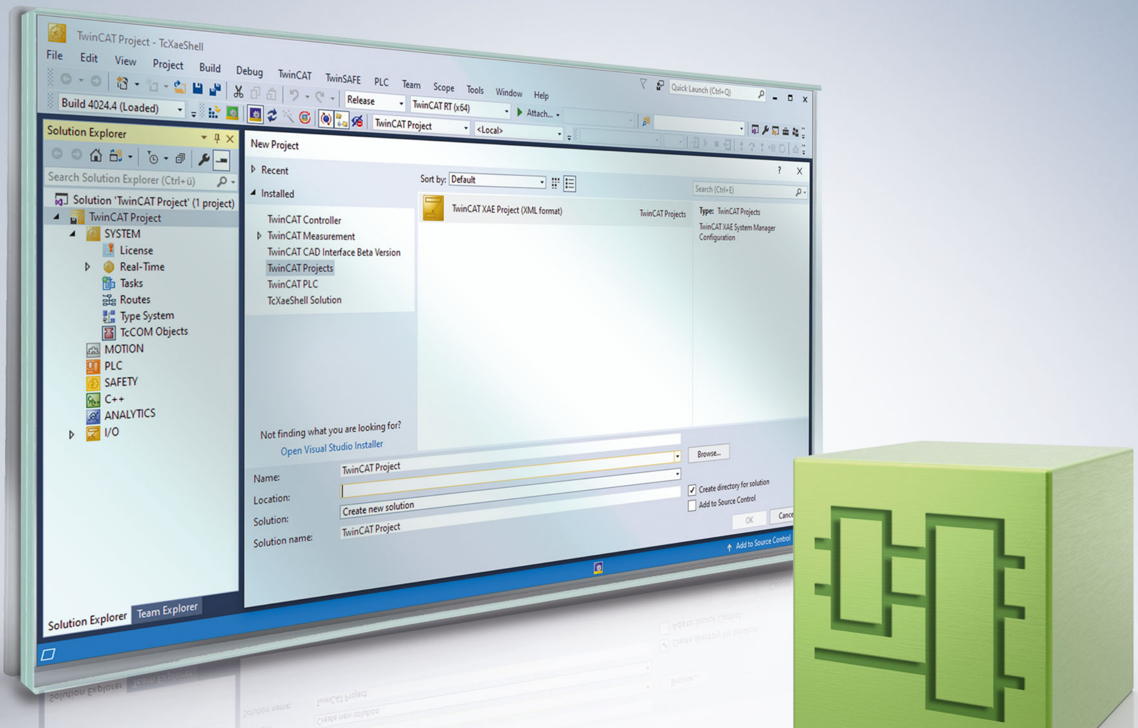
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Lib: Tc2_MBus



2022-08-17 | Version: 1.6

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the requirements for safety, including all the relevant laws, regulations, guidelines and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without prior announcement.
No claims for the modification of products that have already been supplied may be made on the basis of the data, diagrams and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation and drive engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Introduction

The Tc2_MBus library is a comprehensive TwinCAT PLC library for reading M-Bus devices.

The application of this PLC library significantly simplifies the engineering in these areas of building technical equipment.

The function blocks are object-oriented and characterized by a self-contained, more or less complex function.

The input parameters form the interface to the user. The parameters can be used to adapt the function block to its specific task within the associated system.

Thanks to strongly object-oriented encapsulation of complex system functions within the function blocks, comprehensive system programs can be set up with a few function blocks. The blocks are linked to each other via a small number of PLC variables.

The status of all objects is indicated through a large number of different output variables at the function blocks. The simplifies the connection of HMI and visualization systems.

These features offer the following benefits for system programmers during system setup and for system operators during operation:

- Faster creation of system programs.
- Faster system parameterization and commissioning.
- Guarantee of a very large range of system functions at all times.
- Improved readability of programs (prerequisite for long-term maintainability and expandability of the systems)
- Improved reusability of templates for systems or system components
- Easier familiarization of personnel.
- Easier extension of existing systems.
- Programs are easier to document.

The user of this library requires basic knowledge of the following:

- TwinCAT XAE
- PC and network knowledge
- Structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- Technology of M-Bus devices
- Relevant safety regulations for building technical equipment

This software library is intended for building automation system partners of Beckhoff Automation GmbH & Co. KG. The system partners operate in the field of building automation and are concerned with the installation, commissioning, expansion, maintenance and service of measurement, control and regulating systems for the technical equipment of buildings.

The Tc2_MBus library is usable on all hardware platforms that support TwinCAT 3.1 or higher.

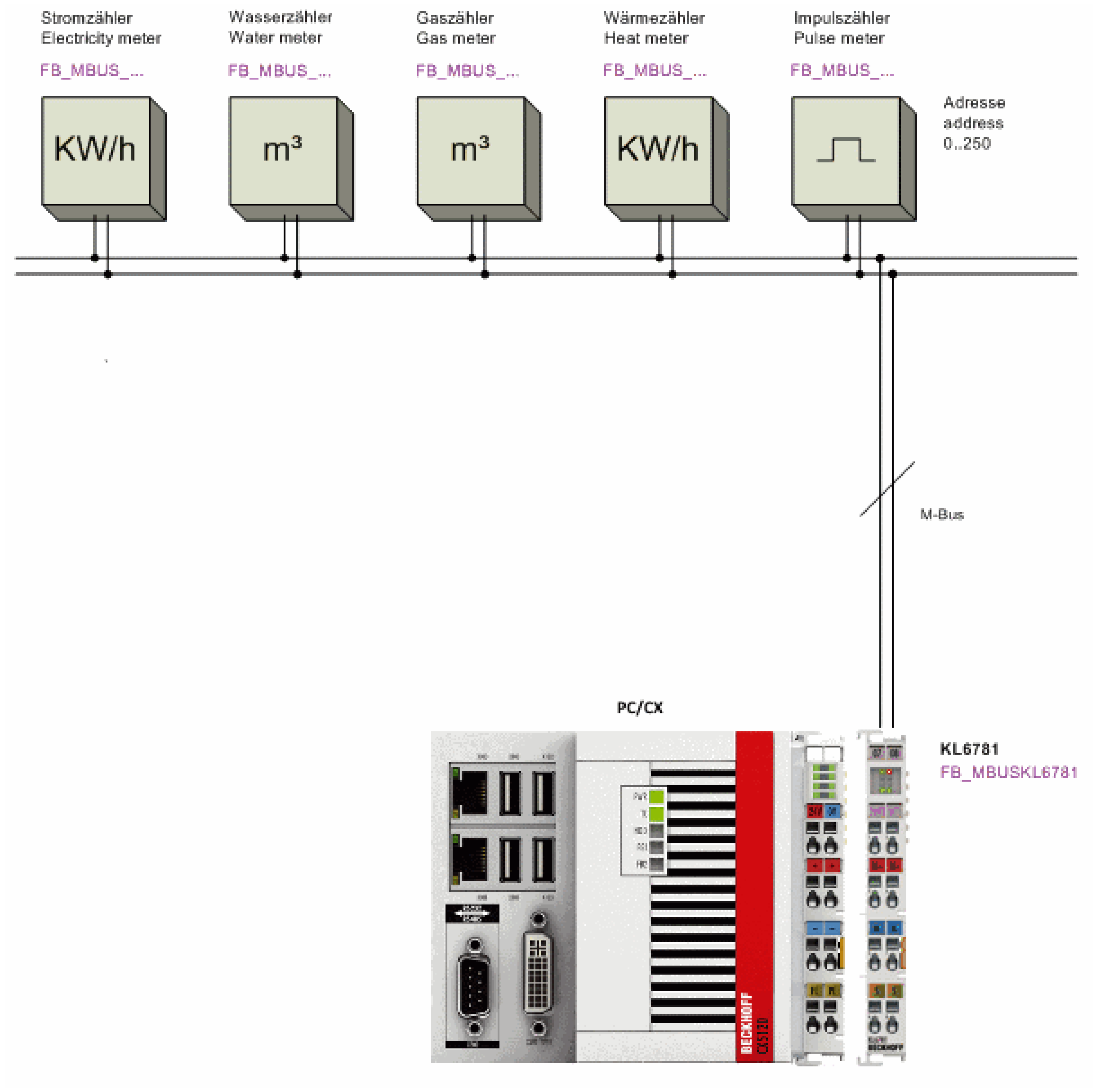
Hardware documentation in the Beckhoff information system:

https://infosys.beckhoff.com/content/1033/kl6781/index.html

Preferred format: LReal.

M-Bus devices may supply very large values (the DWord value range may be exceeded). They are therefore output in string format. Conversions to Real format may lead to inaccuracies/invalid values. Conversions to LReal format are therefore preferable.

3 M-Bus



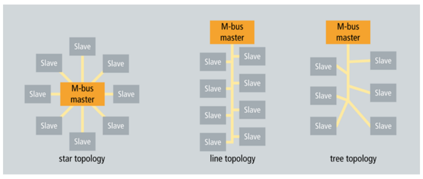
M-Bus = metering bus

The M-Bus is a fieldbus for the recording consumption data (e.g. energy meters). Further details about M-Bus can be found under www.m-bus.com. The M-Bus is European standard and is described in the EN1434 standard. The data are sent serially from a slave (measuring device) to a master (level converter with PC). Master and slave are connected via a two-wire cable that is protected against polarity reversal. With primary addressing up to 250 slaves can be connected in star, strand or tree topologies. Ring structures are not permitted. Devices from different manufacturers can be operated on the same bus.

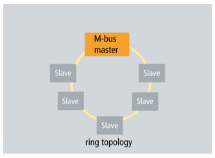
The master controls the communication on the bus by requesting data from the slaves. The slaves can respond with a fixed or variable data structure. The M-Bus library only evaluates data with variable data structure (low byte first). The slaves do not communicate with each other. The data have to be requested sequentially from the slaves.

3.1 Topology

Star, line and tree topology



Ring topology



Ring topology not supported

Ring topology is possible for M-Bus, but not recommended and therefore also not supported by Beckhoff.

3.2 Bulletin

3.2.1 Functionality of the function block

Three methods for reading M-Bus meters are offered:

1. The variable **tMinSendTime**> t#0s of the meter block is used to read the meter automatically once the time has elapsed. The variable is internally preset to t#2s.
2. A positive edge of the variable **bStart** of the meter block triggers one reading of the meter.
3. A positive edge of the variables **bStart** of the block FB_MBUS_KL6781() [▶ 19] triggers one reading of all meters.

If several meter blocks receive a start command at the same time, they are started in the order in which they are called in the PLC.

The variable **bReady** becomes TRUE for one cycle once the block has received the data.

bError becomes TRUE if an error has occurred. This error is described with eError [▶ 198].

To read the meter after a start/ restart of the PLC, set the variable **bReadInit** to TRUE; otherwise to FALSE. Internally this variable is preset to TRUE.

eBaudrate: This variable is internally preset to 2400 baud. To read the meter with this baud rate (2400 baud), this variable does not have to be set explicitly. If the baud rate is changed, the KL6781 is adjusted automatically. This makes it possible to read meters with different baud rates in an M-Bus network. The baud rate of the meters is not changed. They must be able to operate with the baud rate specified here. Some meters operate with automatic baud rate detection. For further information please refer to the user guide of the meter.

bSND_NKE: Internally this variable is preset to TRUE. SND_NKE is a special telegram to the slave. This telegram triggers an initialization of the receiver. This telegram is important for meters, which send several telegrams. These meters respond to a SND_NKE with the first telegram. If TRUE, the SND_NK telegram is sent before the actual query. If FALSE, the SND_NKE telegram is not sent.

bDisabled =TRUE can be used to interrupt processing of the block. If a meter query is in progress, it is completed.

3.2.2 Long set

Data is sent to the M-bus device with a long set. The long set is composed of a maximum of 255 bytes and is transferred to the counter with the FB_MBUS_General_Send() [▸ 86] block.

Structure of the protocol:

Byte	Long set	Description	Assignment in the "FB_MBUS_General_Send" block
1	Start character	68hex	Is added in the block
2	L field	Length of user data plus 3	Is added in the block
3	L field	Length of user data plus 3	Is added in the block
4	Start character	68hex	Is added in the block
5	**C field**	Function field	Is transferred to the **"byC_Field"** input variable
6	**A field**	Primary address of the M-Bus device	Is transferred to the **"usiAddress"** input variable
7	**CI Field**	Identifier field	Is transferred to the **"byCI_Field"** input variable
8..x	**User data (0..240)**	User data	Are transferred to the **'arrData'** input variable
x+1	Checksum	Checksum	Is added in the block
x+2	Stop character	16hex	Is added in the block

Only the bytes marked in bold letters need to be transferred to the block.

The user data in the 'arrData' array must contain '16hex' as the last character. It is important to ensure that the subsequent bytes are empty.

Sample: Changing the primary address at address 14, old address is 0

```
(*Transfer of user data*)

fbSend.arrData[0]:=16#01; (*DIF / Data format 8 -bit integer*)
fbSend.arrData[1]:=16#7A; (*VIF / Change address*)
fbSend.arrData[2]:=14; (*New address = 14*)
fbSend.arrData[3]:=16#16; (*Do not transfer stop character/checksum; they will be calculated in the block*)

fbSend.byC_Field:=16#53; (*C field*)
fbSend.byCI_Field:=16#51; (*CI field*)
fbSend.usiAddress:=0; (*Old address*)

fbSend(iComId:=1, (*Block call*)
bStart:=bStart,
bInit:=TRUE);
```

Sending is started with the 'bStart' variable.

3.2.3 Primary address

The counters are addressed via the primary address. This can be set at the device, via manufacturer software, or with the function blocks FB_MBUS_ChangeAdr() [▸ 76] and FB_MBUS_General_send() [▸ 86].

All meters on a level converter/serial interface must have a unique address (0..250).

Address 0-250: Addresses of the devices

Address 251: not used at present

Address 252: not used at present

Address 253: Use of secondary addressing

Address 254: Send to all M-bus devices with response (E5 hexadecimal). If several devices are connected, all will answer. This leads to data collisions. Therefore, this address should only be used if only one device is connected.

Address 255: Send to all M-Bus devices without response.

3.2.4 Secondary address

Like the primary address, the secondary address is used to identify the terminal device. Like the primary address, the secondary address is used to identify the terminal device. The identification number alone can be used to form 100 million different values. In addition, it is not necessary to allocate primary addresses.

A secondary address has the following structure, according to the M-Bus standard:
Ident no.: 4 bytes / 8-digit BCD device ID data
Manufacturer code: 2 bytes / manufacturer code
Version: 1 byte / generation number of the manufacturer
Medium: 1 byte / medium

To use secondary addressing, set the primary address to 253.

The secondary address is transferred to the function block via the structure "stSecAdr" (ST_MBUS_SecAdr [▸ 206]).

The manufacturer code, version and medium are internally preset to 16#FF, so that these values do not have to be specified explicitly.

Sample calls:

```
stSecAdr1.udiIdNumber    := 16#12345678;
stSecAdr1.uiManufacturer := 16#FFFF;
stSecAdr1.usiMedium      := 16#FF;
stSecAdr1.usiVersion     := 16#FF;
fbmeter(
  usiAddress            := 253,
  stSecAdr.udiIdNumber := stSecAdr1,
  stCom                 := stComKL6781_1);
```

or

```
fbmeter.stSecAdr.udiIdNumber := 16#12345678;
fbmeter(
  usiAddress := 253,
  stCom      := stComKL6781_1);
```

4 Programming

The manufacturer-specific blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General() [▶ 77], FB_MBUS_General_Ext() [▶ 80] or FB_MBUS_General_Param() [▶ 84] from the folder "General [▶ 75]" should be used. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
general	Communication with KL6781		FB_MBUSKL6781 [▶ 19]
General [▶ 75]	Electricity meter	all electricity meters	FB_MBUS_General_Electricity [▶ 78]
	Heat meter	all heat meters	FB_MBUS_General_Heat [▶ 82]
	Water meter	all water meters	FB_MBUS_General_Water [▶ 87]
	Raw data of the first telegram	all	FB_MBUS_RawData [▶ 89]
	max. 40 values from the first telegram	all	FB_MBUS_General [▶ 77]
	all telegrams for all values	all	FB_MBUS_General_Ext [▶ 80]
	values parameterisable	all	FB_MBUS_General_Param [▶ 84]
	universal send blocks	all	FB_MBUS_General_Send [▶ 86]
	scan block	all	FB_MBUS_Scan [▶ 90]
	Change primary address	all	FB_MBUS_ChangeAdr [▶ 76]
ABB [▶ 20]	Electricity meter	DELTAplus DZ+	FB_MBUS_ABB_DZ [▶ 21]
Actaris [▶ 23]	Heat meter	CF-Echo II	FB_MBUS_ACW_CF [▶ 23]
	Arithmetic unit	CF-51	FB_MBUS_ACW_CF [▶ 23]
	Arithmetic unit	CF-55	FB_MBUS_ACW_CF [▶ 23]
	Water meter	MB +M	FB_MBUS_ACW_PlusM [▶ 25]
Aquametro [▶ 27]	Water meter	SAPHIR	FB_MBUS_AMT_SAPHIR [▶ 36]
	Heat meter	CALEC MB	FB_MBUS_AMT_CALEC [▶ 32]
	Heat meter	CALEC ST, version C4	FB_MBUS_AMT_CALEC_STC4 [▶ 34]
	Heat meter	AMTRON	FB_MBUS_AMT_AMTRON [▶ 30]
	Pulse collector	AMBUS	FB_MBUS_AMT_AMBUS [▶ 28]
	Heat meter	AMTRON SONIC D	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
Berg [▶ 38]	Electricity meter	DZ+	FB_MBUS_BEC_DZ [▶ 40]
	Electricity meter	DCMi	FB_MBUS_BEC_DCMi [▶ 38]
Brunata [▶ 42]	Heat meter	HGQ / HGS	FB_MBUS_BHG_HGx [▶ 42]
	Heat meter	Optuna H (775)	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
Carlo Gavazzi [▶ 44]	Energy meter	EM24	FB_MBUS_GAV_EM24 [▶ 45]
Cynox [▶ 47]	Pulse counter	MCount2C	FB_MBUS_CYN_MCount2C [▶ 47]
Elster [▶ 49]	Gas meter	Encoder Z6	FB_MBUS_ELS_EncoderZ6 [▶ 49]
elvaco [▶ 51]	Temperature and humidity sensors	CMa10 & CMa20	FB_MBUS_ELV_CMa10_20 [▶ 51]
EMH [▶ 53]	Electricity meter	DIZ	FB_MBUS_EMH_DIZ [▶ 54]
	Electricity meter	EIZ-E	FB_MBUS_EMH_EIZE [▶ 56]
	Electricity meter	EIZ-G	FB_MBUS_EMH_EIZG [▶ 58]
	Electricity meter	MIZ	FB_MBUS_EMH_MIZ [▶ 60]
EMU [▶ 62]	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7 [▶ 62]

Manufacturer	Type	Device	Block
	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7_Option8 [▶ 65]
	Electricity meter	Allrounder 3/5	FB_MBUS_EMU_3_5_Allrounder [▶ 68]
	Electricity meter	DHZ 5/63	FB_MBUS_EMU_DHZ_5_63 [▶ 70]
Engelmann [▶ 72]	Heat meter	Sensostar 2C	FB_MBUS_EFF_SensoStar2C [▶ 73]
Gossen Metrawatt [▶ 91]	Electricity meter	U128x	FB_MBUS_GMC_Electricity [▶ 92]
	Electricity meter	U138x	FB_MBUS_GMC_Electricity [▶ 92]
GWF [▶ 93]	Water meter		FB_MBUS_GWF_Coder [▶ 94]
	Gas meter	S1	FB_MBUS_GWF_Coder [▶ 94]
	Gas meter	Z1	FB_MBUS_GWF_Coder [▶ 94]
Hydrometer [▶ 95]	2 pulse inputs	HYDRO-PORT Pulse	FB_MBUS_HYD_PortPulse [▶ 100]
	2 analog inputs 1 temperature sensor	HYDRO-PORT Analog	FB_MBUS_HYD_PortAnalog [▶ 98]
	Water meter	Flypper	FB_MBUS_HYD_Flypper [▶ 96]
	Heat meter	Sharky 773	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
	Heat meter	Sharky 775	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
ista [▶ 107]	Water meter	domaqua® m	FB_MBUS_IST_Istameter [▶ 107]
	Water meter	istameter® m	FB_MBUS_IST_Istameter [▶ 107]
	Water meter	istameter III	FB_MBUS_IST_IstameterIII [▶ 109]
	Pulse counter	pulsonic II	FB_MBUS_IST_PulsonicII [▶ 111]
	Heat meter	sensonic II	FB_MBUS_IST_SensonicII [▶ 113]
Itron [▶ 115]	Energy meter	Integral-V-UltraLite	FB_MBUS_ITR_IntegralVUltraLite [▶ 115]
Janitza [▶ 117]	Electricity meter	UMG96S	FB_MBUS_JAN_UMG96S [▶ 118]
Kamstrup [▶ 120]	Electricity meter	Kamstrup 162	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 351	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 382	FB_MBUS_KAM_KamstrupE [▶ 121]
	Heat/cold meter	Maxical III	FB_MBUS_KAM_Maxical_III [▶ 123]
	Heat/cold meter	Multical 401	FB_MBUS_KAM_Multical [▶ 125]
	Heat/cold meter	Multical 402	FB_MBUS_KAM_Multical402 [▶ 127]
	Water meter	Multical 41	FB_MBUS_KAM_Multical41 [▶ 129]
	Heat/cold meter	Multical 601	FB_MBUS_KAM_Multical601 [▶ 131]
KUNDO [▶ 133]	Heat/cold meter	Compact WMZ G20	FB_MBUS_KST_G20 [▶ 134]
	Heat/cold meter	Compact WMZ G21	FB_MBUS_KST_G20 [▶ 134]
	External M-Bus module	him1s	FB_MBUS_KST_him1 [▶ 136]
	External M-Bus module	him1plus	FB_MBUS_KST_him1 [▶ 136]
	Pulse input	him1plus	FB_MBUS_KST_him1Puls [▶ 138]
Landis & Gyr [▶ 139]	Heat/cold meter	ULTRAHEAT 2WR5	FB_MBUS_LUG_Heat [▶ 140]
	Heat/cold meter	ULTRAHEAT 2WR6	FB_MBUS_LUG_Heat [▶ 140]

Manufacturer	Type	Device	Block
	Heat/cold meter	ULTRAHEAT UH50	FB_MBUS_LUG_Heat [▶ 140]
Metrima [▶ 142]	Heat meter	F22 (standard values)	FB_MBUS_SVM_F22 [▶ 142]
	Heat meter	F22 (with additional output values)	FB_MBUS_SVM_F22_Ext [▶ 145]
NZR [▶ 147]	Pulse memory module	IC-M2	FB_MBUS_NZR_ICM2 [▶ 148]
	Pulse memory module	IC-M2C	FB_MBUS_NZR_ICM2 [▶ 148]
	Water meter	Modularis 2	FB_MBUS_NZR_Modularis2 [▶ 150]
OPTEC [▶ 151]	Electricity meter	ECS Type 2	FB_MBUS_OPT_ECSType2 [▶ 152]
Relay [▶ 154]	1-4 analog inputs	AnDi 1-4	FB_MBUS_REL_AnDi [▶ 155]
	4 digital inputs	PadIn 4	FB_MBUS_REL_PadIn4 [▶ 157]
	1-way pulse adapter	PadPuls M1	FB_MBUS_REL_PadPulsM1 [▶ 159]
	1-way pulse adapter	PadPuls M1C	FB_MBUS_REL_PadPulsM1 [▶ 159]
	2-way pulse adapter	PadPuls M2	FB_MBUS_REL_PadPulsM2 [▶ 161]
	2-way pulse adapter	PadPuls M2C	FB_MBUS_REL_PadPulsM2 [▶ 161]
	4-way pulse adapter	PadPuls M4	FB_MBUS_REL_PadPulsM4 [▶ 163]
	4-way pulse adapter	PadPuls M4L	FB_MBUS_REL_PadPulsM4 [▶ 163]
Saia-Burgess [▶ 165]	Electricity meter	ALD1	FB_MBUS_SBC_ALD1 [▶ 166]
	Electricity meter	ALE3	FB_MBUS_SBC_ALE3 [▶ 168]
	Electricity meter	AWD3	FB_MBUS_SBC_ALE3 [▶ 168]
Schlumberger [▶ 170]	Heat meter	Integral-Mk MaXX	FB_MBUS_SLB_MK_MaXX [▶ 173]
	Heat meter	CF Echo I	FB_MBUS_SLB_CFEchoI [▶ 171]
Schneider Electric [▶ 175]	Electricity meter	iEM3135	FB_MBUS_SEC_iEM3135 [▶ 176]
Sensus [▶ 178]	Heat/cold meter	PolluStat E	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluTherm	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluCom E	FB_MBUS_SEN_Pollu [▶ 179]
	Water meter		FB_MBUS_SEN_Water [▶ 181]
Sontex [▶ 183]	Heat/cold meter	Supercal 531	FB_MBUS_SON_Supercal531 [▶ 183]
TIP [▶ 185]	Electricity meter	SINUS 85 M	FB_MBUS_TIP_SINUS85M [▶ 186]
Zenner [▶ 189]	Arithmetic unit	multidataWR3	FB_MBUS_ZRM_multidataWR3 [▶ 190]
	Heat meter	zelsiusZR	FB_MBUS_ZRM_zelsiusZR [▶ 192]

4.1 POUs

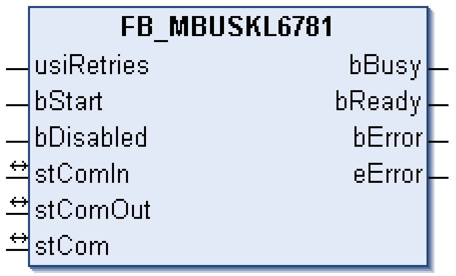
The manufacturer-specific blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General() [▶ 77], FB_MBUS_General_Ext() [▶ 80] or FB_MBUS_General_Param() [▶ 84] from the folder "General [▶ 75]" should be used. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Function block
General	Communication with KL6781		FB_MBUSKL6781 [▸ 19]
General [▸ 75]	Electricity meter	all electricity meters	FB_MBUS_General_Electricity [▸ 78]
	Heat meter	all heat meters	FB_MBUS_General_Heat [▸ 82]
	Water meter	all water meters	FB_MBUS_General_Water [▸ 87]
	Raw data of the first telegram	all	FB_MBUS_RawData [▸ 89]
	max. 40 values from the first telegram	all	FB_MBUS_General [▸ 77]
	all telegrams for all values	all	FB_MBUS_General_Ext [▸ 80]
	values parameterisable	all	FB_MBUS_General_Param [▸ 84]
	universal send blocks	all	FB_MBUS_General_Send [▸ 86]
	scan block	all	FB_MBUS_Scan [▸ 90]
	Change primary address	all	FB_MBUS_ChangeAdr [▸ 76]
ABB [▸ 20]	Electricity meter	DELTAplus DZ+	FB_MBUS_ABB_DZ [▸ 21]
Actaris [▸ 23]	Heat meter	CF-Echo II	FB_MBUS_ACW_CF [▸ 23]
	Arithmetic unit	CF-51	FB_MBUS_ACW_CF [▸ 23]
	Arithmetic unit	CF-55	FB_MBUS_ACW_CF [▸ 23]
	Water meter	MB +M	FB_MBUS_ACW_PlusM [▸ 25]
Aquametro [▸ 27]	Water meter	SAPHIR	FB_MBUS_AMT_SAPHIR [▸ 36]
	Heat meter	CALEC MB	FB_MBUS_AMT_CALEC [▸ 32]
	Heat meter	CALEC ST, version C4	FB_MBUS_AMT_CALEC_STC4 [▸ 34]
	Heat meter	AMTRON	FB_MBUS_AMT_AMTRON [▸ 30]
	Pulse collector	AMBUS	FB_MBUS_AMT_AMBUS [▸ 28]
	Heat meter	AMTRON SONIC D	FB_MBUS_HYD_Sharky [▸ 102], FB_MBUS_HYD_Sharky_00 [▸ 104]
Berg [▸ 38]	Electricity meter	DZ+	FB_MBUS_BEC_DZ [▸ 40]
	Electricity meter	DCMi	FB_MBUS_BEC_DCMi [▸ 38]
Brunata [▸ 42]	Heat meter	HGQ / HGS	FB_MBUS_BHG_HGx [▸ 42]
	Heat meter	Optuna H (775)	FB_MBUS_HYD_Sharky [▸ 102], FB_MBUS_HYD_Sharky_00 [▸ 104]
Carlo Gavazzi [▸ 44]	Energy meter	EM24	FB_MBUS_GAV_EM24 [▸ 45]
Cynox [▸ 47]	Pulse counter	MCount2C	FB_MBUS_CYN_MCount2C [▸ 47]
Elster [▸ 49]	Gas meter	Encoder Z6	FB_MBUS_ELS_EncoderZ6 [▸ 49]
elvaco [▸ 51]	Temperature and humidity sensors	CMa10 & CMa20	FB_MBUS_ELV_CMa10_20 [▸ 51]
EMH [▸ 53]	Electricity meter	DIZ	FB_MBUS_EMH_DIZ [▸ 54]
	Electricity meter	EIZ-E	FB_MBUS_EMH_EIZE [▸ 56]
	Electricity meter	EIZ-G	FB_MBUS_EMH_EIZG [▸ 58]
	Electricity meter	MIZ	FB_MBUS_EMH_MIZ [▸ 60]
EMU [▸ 62]	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7 [▸ 62]
	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7_Option8 [▸ 65]

Manufacturer	Type	Device	Function block
	Electricity meter	Allrounder 3/5	FB_MBUS_EMU_3_5_Allrounder [▶ 68]
	Electricity meter	DHZ 5/63	FB_MBUS_EMU_DHZ_5_63 [▶ 70]
Engelmann [▶ 72]	Heat meter	Sensostar 2C	FB_MBUS_EFF_SensoStar2C [▶ 73]
Gossen Metrawatt [▶ 91]	Electricity meter	U128x	FB_MBUS_GMC_Electricity [▶ 92]
	Electricity meter	U138x	FB_MBUS_GMC_Electricity [▶ 92]
GWF [▶ 93]	Water meter		FB_MBUS_GWF_Coder [▶ 94]
	Gas meter	S1	FB_MBUS_GWF_Coder [▶ 94]
	Gas meter	Z1	FB_MBUS_GWF_Coder [▶ 94]
Hydrometer [▶ 95]	2 pulse inputs	HYDRO-PORT Pulse	FB_MBUS_HYD_PortPulse [▶ 100]
	2 analog inputs 1 temperature sensor	HYDRO-PORT Analog	FB_MBUS_HYD_PortAnalog [▶ 98]
	Water meter	Flypper	FB_MBUS_HYD_Flypper [▶ 96]
	Heat meter	Sharky 773	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
	Heat meter	Sharky 775	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]
ista [▶ 107]	Water meter	domaqua® m	FB_MBUS_IST_Istameter [▶ 107]
	Water meter	istameter® m	FB_MBUS_IST_Istameter [▶ 107]
	Water meter	istameter III	FB_MBUS_IST_IstameterIII [▶ 109]
	Pulse counter	pulsonic II	FB_MBUS_IST_PulsonicII [▶ 111]
	Heat meter	sensonic II	FB_MBUS_IST_SensonicII [▶ 113]
Itron [▶ 115]	Energy meter	Integral-V UltraLite	FB_MBUS_ITR_IntegralVUltraLite [▶ 115]
Janitza [▶ 117]	Electricity meter	UMG96S	FB_MBUS_JAN_UMG96S [▶ 118]
Kamstrup [▶ 120]	Electricity meter	Kamstrup 162	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 351	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 382	FB_MBUS_KAM_KamstrupE [▶ 121]
	Heat/cold meter	Maxical III	FB_MBUS_KAM_Maxical_III [▶ 123]
	Heat/cold meter	Multical 401	FB_MBUS_KAM_Multical [▶ 125]
	Heat/cold meter	Multical 402	FB_MBUS_KAM_Multical402 [▶ 127]
	Water meter	Multical 41	FB_MBUS_KAM_Multical41 [▶ 129]
	Heat/cold meter	Multical 601	FB_MBUS_KAM_Multical601 [▶ 131]
KUNDO [▶ 133]	Heat/cold meter	Compact WMZ G20	FB_MBUS_KST_G20 [▶ 134]
	Heat/cold meter	Compact WMZ G21	FB_MBUS_KST_G20 [▶ 134]
	External M-Bus module	him1s	FB_MBUS_KST_him1 [▶ 136]
	External M-Bus module	him1plus	FB_MBUS_KST_him1 [▶ 136]
	Pulse input	him1plus	FB_MBUS_KST_him1Puls [▶ 138]
Landis & Gyr [▶ 139]	Heat/cold meter	ULTRAHEAT 2WR5	FB_MBUS_LUG_Heat [▶ 140]
	Heat/cold meter	ULTRAHEAT 2WR6	FB_MBUS_LUG_Heat [▶ 140]
	Heat/cold meter	ULTRAHEAT UH50	FB_MBUS_LUG_Heat [▶ 140]
Metrima [▶ 142]	Heat meter	F22 (default values)	FB_MBUS_SVM_F22 [▶ 142]

Manufacturer	Type	Device	Function block
		F22 (with additional output values)	FB_MBUS_SVM_F22_Ext [▶ 145]
NZR [▶ 147]	Pulse memory module	IC-M2	FB_MBUS_NZR_ICM2 [▶ 148]
	Pulse memory module	IC-M2C	FB_MBUS_NZR_ICM2 [▶ 148]
	Water meter	Modularis 2	FB_MBUS_NZR_Modularis2 [▶ 150]
OPTEC [▶ 151]	Electricity meter	ECS Type 2	FB_MBUS_OPT_ECSType2 [▶ 152]
Relay [▶ 154]	1-4 analog inputs	AnDi 1-4	FB_MBUS_REL_AnDi [▶ 155]
	4 digital inputs	PadIn 4	FB_MBUS_REL_PadIn4 [▶ 157]
	1-way pulse adapter	PadPuls M1	FB_MBUS_REL_PadPulsM1 [▶ 159]
	1-way pulse adapter	PadPuls M1C	FB_MBUS_REL_PadPulsM1 [▶ 159]
	2-way pulse adapter	PadPuls M2	FB_MBUS_REL_PadPulsM2 [▶ 161]
	2-way pulse adapter	PadPuls M2C	FB_MBUS_REL_PadPulsM2 [▶ 161]
	4-way pulse adapter	PadPuls M4	FB_MBUS_REL_PadPulsM4 [▶ 163]
	4-way pulse adapter	PadPuls M4L	FB_MBUS_REL_PadPulsM4 [▶ 163]
Saia-Burgess [▶ 165]	Electricity meter	ALD1	FB_MBUS_SBC_ALD1 [▶ 166]
	Electricity meter	ALE3	FB_MBUS_SBC_ALE3 [▶ 168]
	Electricity meter	AWD3	FB_MBUS_SBC_ALE3 [▶ 168]
Schlumberger [▶ 170]	Heat meter	Integral-Mk MaXX	FB_MBUS_SLB_MK_MaXX [▶ 173]
	Heat meter	CF Echo I	FB_MBUS_SLB_CFEchoI [▶ 171]
Schneider Electric [▶ 175]	Electricity meter	iEM3135	FB_MBUS_SEC_iEM3135 [▶ 176]
Sensus [▶ 178]	Heat/cold meter	PolluStat E	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluTherm	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluCom E	FB_MBUS_SEN_Pollu [▶ 179]
	Water meter		FB_MBUS_SEN_Water [▶ 181]
Sontex [▶ 183]	Heat/cold meter	Supercal 531	FB_MBUS_SON_Supercal531 [▶ 183]
TIP [▶ 185]	Electricity meter	SINUS 85 M	FB_MBUS_TIP_SINUS85M [▶ 186]
Zenner [▶ 189]	Arithmetic unit	multidataWR3	FB_MBUS_ZRM_multidataWR3 [▶ 190]
	Heat meter	zelsiusZR	FB_MBUS_ZRM_zelsiusZR [▶ 192]

4.1.1 FB_MBUSKL6781



This function block is used to read M-Bus devices via the Bus Terminal KL6781.

The block can only be used in conjunction with at least one **counter block**.

An instance of this block required for each KL6781 terminal.

At 2400 baud the maximum **task time** for this block is 10 ms. If higher task times are required, this block must be processed in a separate fast task.

VAR_INPUT

```
usiRetries : USINT;
bStart     : BOOL;
bDisabled  : BOOL := FALSE;
```

usiRetries: Number of repetitions in the event of errors

bStart: A positive edge at this input triggers one reading of all meters.

bDisabled: TRUE = deselection of the block

VAR_OUTPUT

```
bBusy  : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

VAR_IN_OUT

```
stComIn  : ST_KL6781inData22B;
stComOut : ST_KL6781outData22B;
stCom    : ST_MBUS_Communication;
```

stComIn: Process image of the inputs (see ST_KL6781inData22B [▶ 203]).

stComOut: Process image of the outputs (see ST_KL6781outData22B [▶ 203]).

stCom: This structure is used to link the block with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

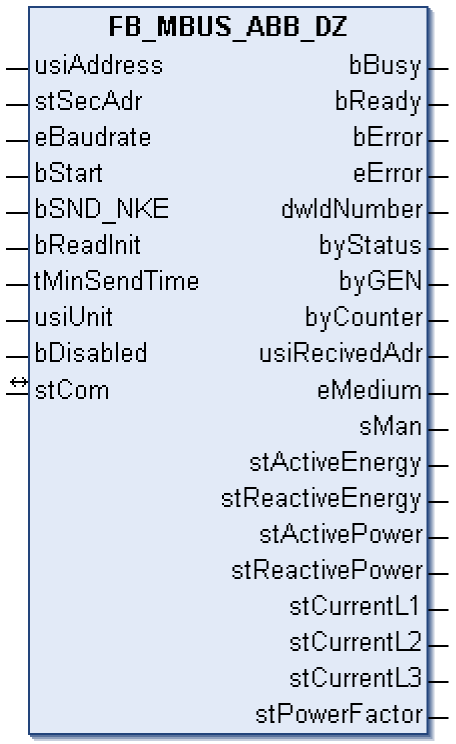
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.2 ABB overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
ABB	Electricity meter	DELTAplus DZ+	FB_MBUS_ABB_DZ [▶ 21]

4.1.2.1 FB_MBUS_ABB_DZ



This block is used to read electricity meters from ABB:

-DELTAplus DZ+

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stActiveEnergy    : ST_MBus_Info;
stReactiveEnergy  : ST_MBus_Info;
stActivePower     : ST_MBus_Info;
stReactivePower   : ST_MBus_Info;
stCurrentL1       : ST_MBus_Info;
stCurrentL2       : ST_MBus_Info;
stCurrentL3       : ST_MBus_Info;
stPowerFactor     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stActiveEnergy: Meter value, total active energy (see ST_MBus_Info [▸ 205]).

stReactiveEnergy: Meter value, total reactive energy (see ST_MBus_Info [▸ 205]).

stActivePower: Current consumption value, total effective power (see ST_MBus_Info [▸ 205]).

stReactivePower: Current consumption value, total reactive power (see ST_MBus_Info [▸ 205]).

stCurrentL1: Current L1 (see ST_MBus_Info [▸ 205]).

stCurrentL2: Current L2 (see ST_MBus_Info [▸ 205]).

stCurrentL3: Current L3 (see ST_MBus_Info [▸ 205]).

stPowerFactor: Total power factor (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

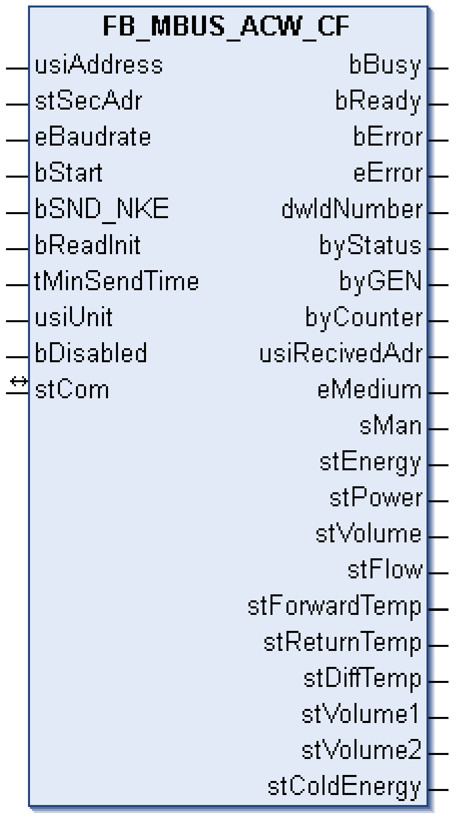
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.3 Actaris overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Actaris	Heat meter	CF-Echo II	FB_MBUS_ACW_CF [▶ 23]
	Arithmetic unit	CF-51	FB_MBUS_ACW_CF [▶ 23]
	Arithmetic unit	CF-55	FB_MBUS_ACW_CF [▶ 23]
	Water meter	MB +M	FB_MBUS_ACW_PlusM [▶ 25]

4.1.3.1 FB_MBUS_ACW_CF



This block is used to read heat meters from Actaris:

-CF-Echo II

-CF-51

-CF-55

Up to two additional water meters can be connected to this device (optional).

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 1200, 2400, 9600 Baud.

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
stVolume1        : ST_MBus_Info;
stVolume2        : ST_MBus_Info;
stColdEnergy     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▶ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▶ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▶ 205]).

stVolume1: Meter reading of additional water meter 1 (option) (see ST_MBus_Info [▶ 205]).

stVolume2: Meter reading of additional water meter 2 (option) (see ST_MBus_Info [▶ 205]).

stColdEnergy: Meter reading, cooling energy consumption (option) (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

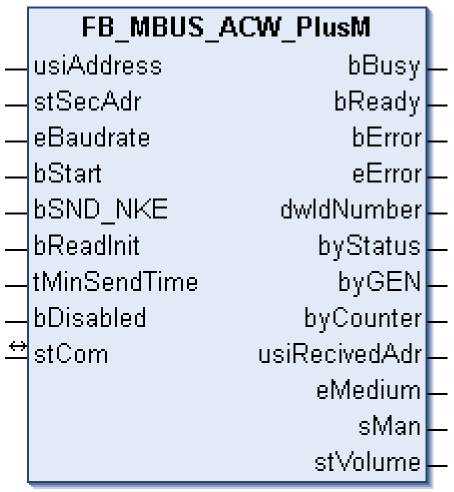
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.3.2 FB_MBUS_ACW_PlusM



This block is used to read water meters from Actaris:

-BM +M

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stVolume         : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

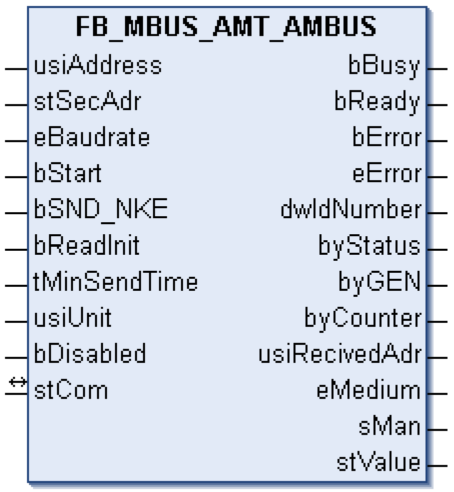
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.4 Aquametro overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Aquametro	Water meter	SAPHIR	FB_MBUS_AMT_SAPHIR [▸ 36]
	Heat meter	CALEC	FB_MBUS_AMT_CALEC [▸ 32]
	Heat meter	CALEC ST, version C4	FB_MBUS_AMT_CALEC_ST C4 [▸ 34]
	Heat meter	AMTRON	FB_MBUS_AMT_AMTRON [▸ 30]
	Pulse collector	AMBUS	FB_MBUS_AMT_AMBUS [▸ 28]
	Heat meter	AMTRON SONIC D	FB_MBUS_HYD_Sharky [▸ 102], FB_MBUS_HYD_Sharky_00 [▸ 104]

4.1.4.1 FB_MBUS_AMT_AMBUS



This block is used to read pulse collectors from Aquametro:

-AMBUS IS

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: (see E_MBUS_baud rate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
```

```
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

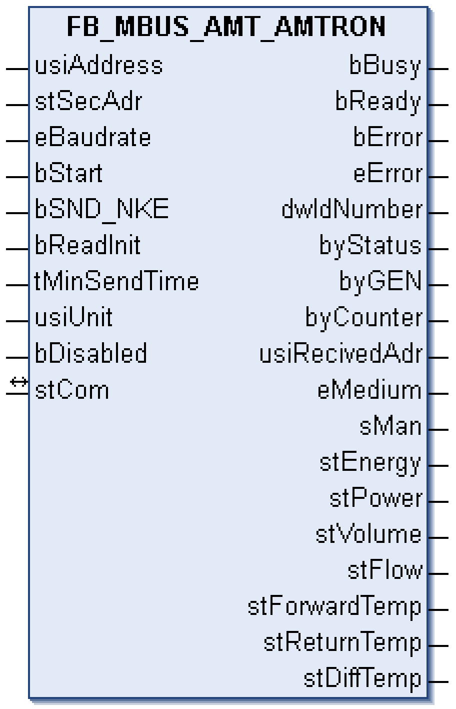
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.4.2 FB_MBUS_AMT_AMTRON



This block is used to read heat meters from Aquametro:

-AMTRON

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stEnergy        : ST_MBus_Info;
stPower         : ST_MBus_Info;
stVolume        : ST_MBus_Info;
stFlow          : ST_MBus_Info;
stForwardTemp   : ST_MBus_Info;
stReturnTemp    : ST_MBus_Info;
stDiffTemp      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

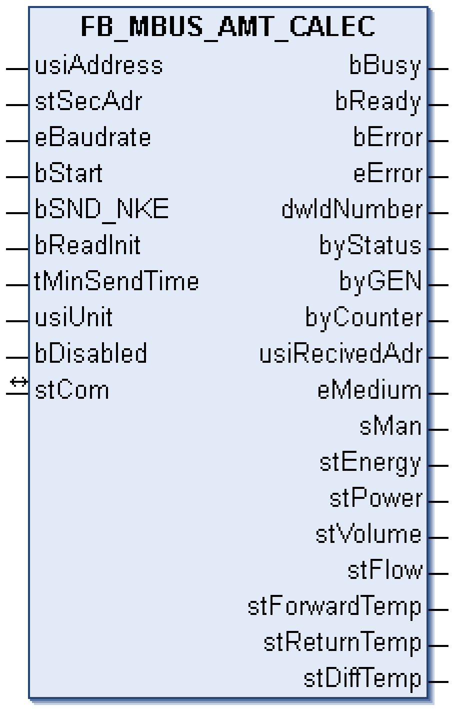
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.4.3 FB_MBUS_AMT_CALEC



This block is used to read heat meters from Aquametro:

-CALEC

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
```

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

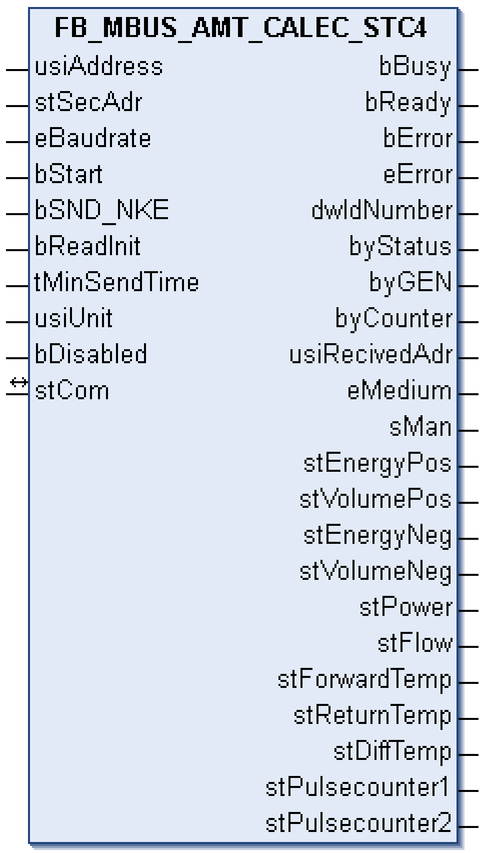
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.4.4 FB_MBUS_AMT_CALEC_STC4



This block is used to read heat meters from Aquametro:

-CALEC ST, version C4

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergyPos        : ST_MBus_Info;
stVolumePos        : ST_MBus_Info;
stEnergyNeg        : ST_MBus_Info;
stVolumeNeg        : ST_MBus_Info;
stPower            : ST_MBus_Info;
stFlow             : ST_MBus_Info;
stForwardTemp      : ST_MBus_Info;
stReturnTemp       : ST_MBus_Info;
stDiffTemp         : ST_MBus_Info;
stPulsecounter1    : ST_MBus_Info;
stPulsecounter2    : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergyPos: Counter value energy consumption (positive) (see ST_MBus_Info [▸ 205]).

stVolumePos: Counter value water consumption (positive) (see ST_MBus_Info [▸ 205]).

stEnergyNeg: Counter value energy consumption (negative) (see ST_MBus_Info [▸ 205]).

stVolumeNeg: Counter value water consumption (negative) (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

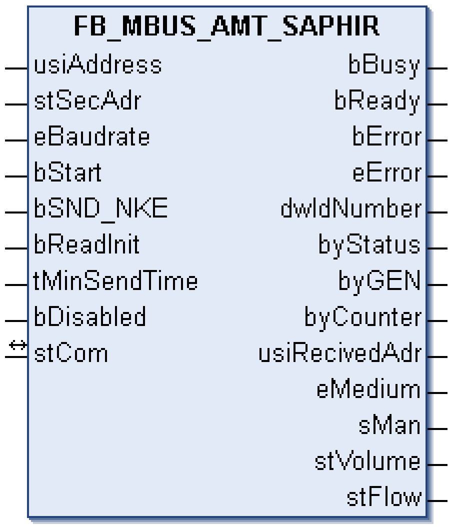
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.4.5 FB_MBUS_AMT_SAPHIR



This block is used to read water meters from Aquametro.

-Saphir

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: (see E_MBUS_baud rate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                 : BOOL;
bReady                : BOOL;
bError                : BOOL;
eError                : E_MBUS_ERROR;
dwIdNumber            : DWORD;
byStatus              : BYTE;
byGEN                 : BYTE;
byCounter             : BYTE;
usiRecivedAdr         : USINT;
eMedium               : E_MBUS_Medium;
sMan                  : STRING(3);
stVolume              : ST_MBus_Info;
stFlow                : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

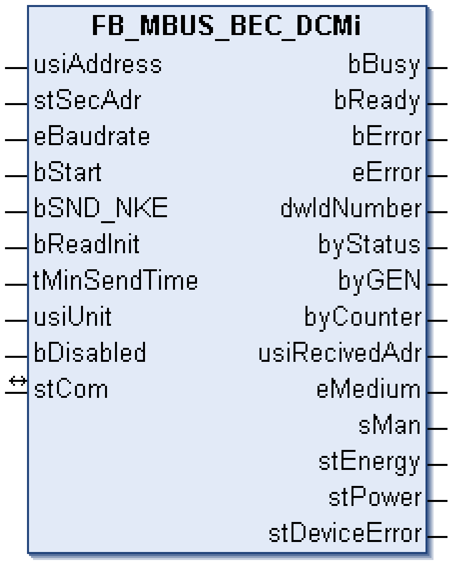
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.5 Berg overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Berg	Electricity meter	DZ+	FB_MBUS_BEC_DZ [▸ 40]
	Electricity meter	DCMi	FB_MBUS_BEC_DCMi [▸ 38]

4.1.5.1 FB_MBUS_BEC_DCMi



This block is used to read electricity meters from Berg:

-DCMi

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stPower            : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

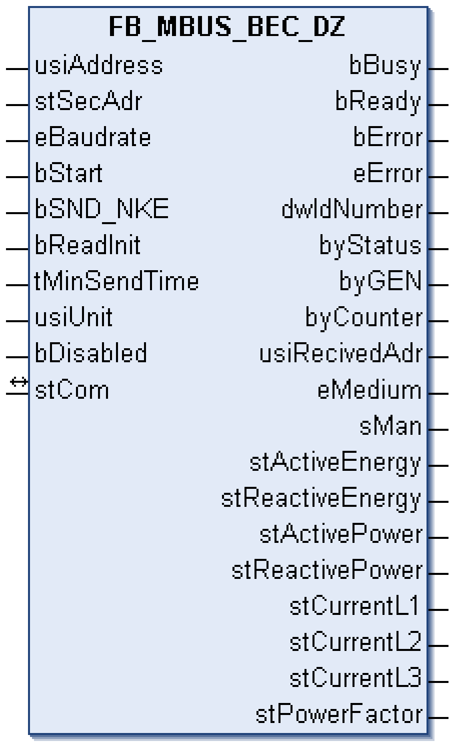
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.5.2 FB_MBUS_BEC_DZ



This block is used to read electricity meters from Berg:

-DZ+

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stActiveEnergy    : ST_MBus_Info;
stReactiveEnergy  : ST_MBus_Info;
stActivePower     : ST_MBus_Info;
stReactivePower   : ST_MBus_Info;
stCurrentL1       : ST_MBus_Info;
stCurrentL2       : ST_MBus_Info;
stCurrentL3       : ST_MBus_Info;
stPowerFactor     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stActiveEnergy: Meter value, total active energy (see ST_MBus_Info [▸ 205]).

stReactiveEnergy: Meter value, total reactive energy (see ST_MBus_Info [▸ 205]).

stActivePower: Current consumption value, total effective power (see ST_MBus_Info [▸ 205]).

stReactivePower: Current consumption value, total reactive power (see ST_MBus_Info [▸ 205]).

stCurrentL1: Current L1 (see ST_MBus_Info [▸ 205]).

stCurrentL2: Current L2 (see ST_MBus_Info [▸ 205]).

stCurrentL3: Current L3 (see ST_MBus_Info [▸ 205]).

stPowerFactor: Total power factor (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

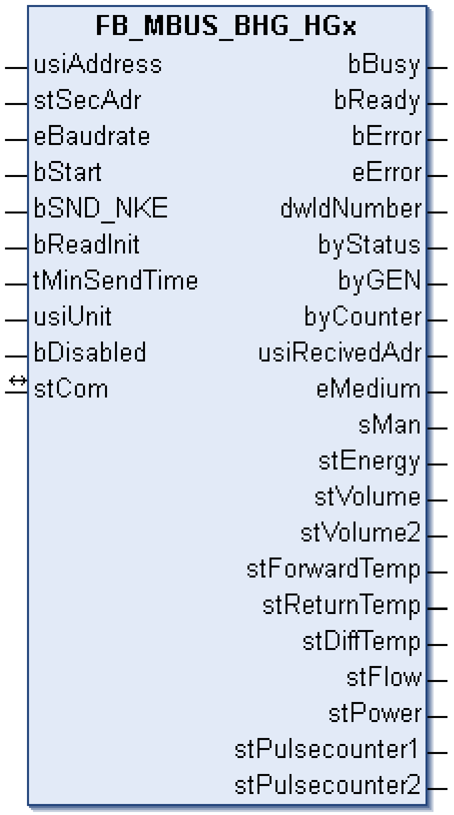
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.6 Brunata overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Brunata	Heat meter	HGQ / HGS	FB_MBUS_BHG_HGx [▶ 42]
	Heat meter	Optuna H (775)	FB_MBUS_HYD_Sharky [▶ 102], FB_MBUS_HYD_Sharky_00 [▶ 104]

4.1.6.1 FB_MBUS_BHG_HGx



This block is used to read heat meters from Brunata:

-HGQ

-HGS

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: (see E_MBUS_baud rate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stVolume2        : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stPower          : ST_MBus_Info;
stPulsecounter1  : ST_MBus_Info;
stPulsecounter2  : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stVolume2: Volume from flow sensor (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

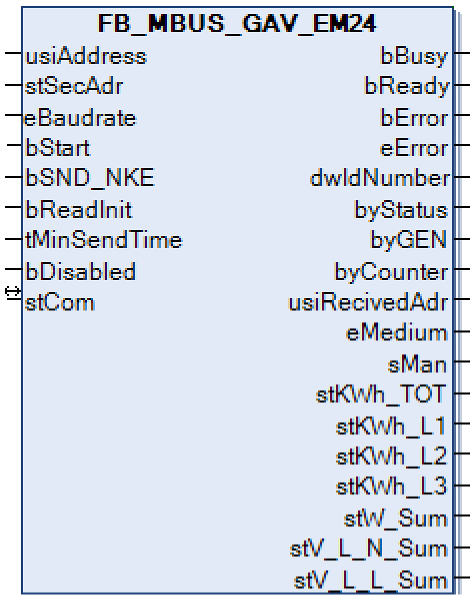
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.7 Carlo Gavazzi overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Carlo Gavazzi	Energy meter	EM24	FB_MBUS_GAV_EM24 [▸ 45]

4.1.7.1 FB_MBUS_GAV_EM24



This block is used to read energy meters from Carl Gavazzi.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
```

```
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stKWh_TOT        : ST_MBus_Info;
stKWh_L1         : ST_MBus_Info;
stKWh_L2         : ST_MBus_Info;
stKWh_L3         : ST_MBus_Info;
stW_Sum          : ST_MBus_Info;
stV_L_N_Sum      : ST_MBus_Info;
stV_L_L_Sum      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stKWh_TOT: See manufacturer information (see ST_MBus_Info [▸ 205]).

stKWh_L1: See manufacturer information (see ST_MBus_Info [▸ 205]).

stKWh_L2: See manufacturer information (see ST_MBus_Info [▸ 205]).

stKWh_L3: See manufacturer information (see ST_MBus_Info [▸ 205]).

stW_Sum: See manufacturer information (see ST_MBus_Info [▸ 205]).

stV_L_N_Sum: See manufacturer information (see ST_MBus_Info [▸ 205]).

stV_L_L_Sum: See manufacturer information (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

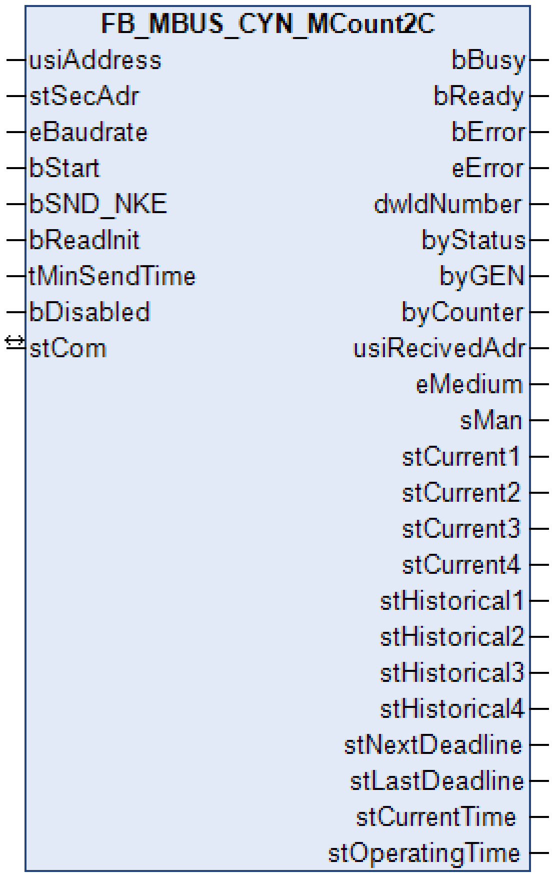
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.8 Cynox

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Function block
Cynox	Pulse counter	MCount2C	FB_MBUS_CYN_MCount2C [▶ 47]

4.1.8.1 FB_MBUS_CYN_MCount2C



This function block is used to read pulse counters from Cynox.

The function block can only be executed together with the function block FB_MBUSKL6781() [▶ 19].

Functionality of the function block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 2400 baud (see E_MBus_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the function block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stCurrent1      : ST_MBus_Info;
stCurrent2      : ST_MBus_Info;
stCurrent3      : ST_MBus_Info;
stCurrent4      : ST_MBus_Info;
stHistorical1   : ST_MBus_Info;
stHistorical2   : ST_MBus_Info;
stHistorical3   : ST_MBus_Info;
stHistorical4   : ST_MBus_Info;
stNextDeadline  : ST_MBus_Info;
stLastDeadline  : ST_MBus_Info;
stCurrentTime   : ST_MBus_Info;
stOperatingTime : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stCurrent1: Current value 1 (see ST_MBus_Info [▸ 205]).

stCurrent2: Current value 2 (see ST_MBus_Info [▸ 205]).

stCurrent3: Current value 3 (see ST_MBus_Info [▸ 205]).

stCurrent4: Current value 4 (see ST_MBus_Info [▸ 205]).

stHistorical1: Historical value 1 (see ST_MBus_Info [▸ 205]).

stHistorical2: Historical value 2 (see ST_MBus_Info [▸ 205]).

stHistorical3: Historical value 3 (see ST_MBus_Info [▸ 205]).

stHistorical4: Historical value 4 (see ST_MBus_Info [▸ 205]).

stNextDeadline: Next reporting date (see ST_MBus_Info [▸ 205]).

stLastDeadline: Last reporting date (see ST_MBus_Info [▸ 205]).

stCurrentTime: Current time (see ST_MBus_Info [▸ 205]).

stOperatingTime: Operating time (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

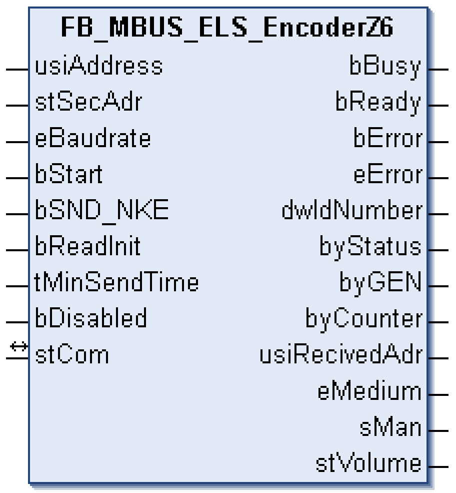
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.32	Tc2_MBus from 3.4.6.0

4.1.9 Elster overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Elster	Gas meter	Encoder Z6	FB_MBUS_ELS_EncoderZ6 [▸ 49]

4.1.9.1 FB_MBUS_ELS_EncoderZ6



This block is used to read meters from Elster:

- Encoder Z6

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stVolume         : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

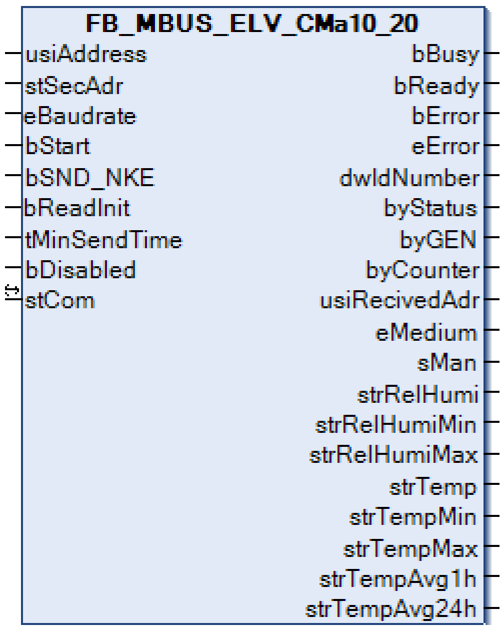
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.10 elvaco overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
elvaco	Temperature and humidity sensors	CMa10 / CMa20	FB_MBUS_ELV_CMa10_20 [▸ 51]

4.1.10.1 FB_MBUS_ELV_CMa10_20



This block is used to read temperature and humidity sensors from elvaco.

Can be used with the sensors CMa10 and CMa20.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
strRelHumi         : ST_MBus_Info;
strRelHumiMin      : ST_MBus_Info;
strRelHumiMax      : ST_MBus_Info;
strTemp            : ST_MBus_Info;
strTempMin         : ST_MBus_Info;
strTempMax         : ST_MBus_Info;
strTempAvg1h       : ST_MBus_Info;
strTempAvg24h      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

strRelHumi: Current relative humidity (see ST_MBus_Info [▸ 205]).

strRelHumiMin: Lowest relative humidity since the last min/max reset command (see ST_MBus_Info [▸ 205]).

strRelHumiMax: Highest relative humidity since the last min/max reset command (see ST_MBus_Info [▸ 205]).

strTemp: Current temperature (see ST_MBus_Info [▸ 205]).

strTempMin: Lowest temperature since the last min/max reset command (see ST_MBus_Info [▸ 205]).

strTempMax: Highest temperature since the last min/max reset command (see ST_MBus_Info [▸ 205]).

strTempAvg1h: 1-hour mean value for temperature (see ST_MBus_Info [▸ 205]).

strTempAvg24h: 24-hour mean value for temperature (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

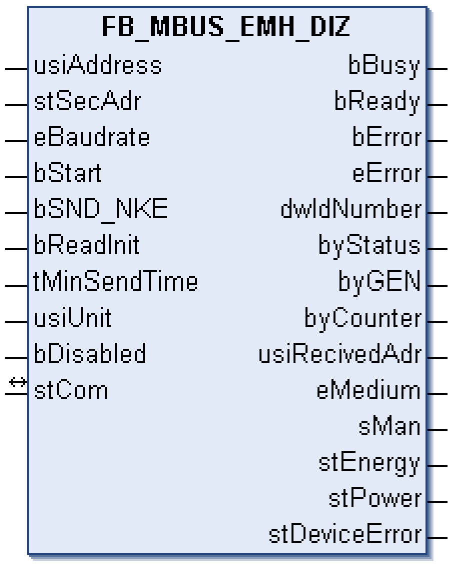
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.11 EMH overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
EMH	Electricity meter	DIZ	FB_MBUS_EMH_DIZ [▸ 54]
	Electricity meter	EIZ-E	FB_MBUS_EMH_EIZE [▸ 56]
	Electricity meter	EIZ-G	FB_MBUS_EMH_EIZG [▸ 58]
	Electricity meter	MIZ	FB_MBUS_EMH_MIZ [▸ 60]

4.1.11.1 FB_MBUS_EMH_DIZ



This block is used to read electricity meters from EMH:

-DIZ

Unidirectional tariff meter only

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
```

```
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stDeviceError    : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

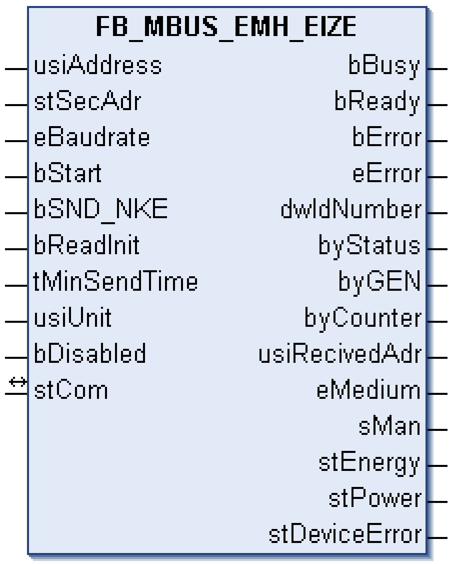
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.11.2 FB_MBUS_EMH_EIZE



This block is used to read electricity meters from EMH:

-EIZ-E

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
```

```
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stDeviceError     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

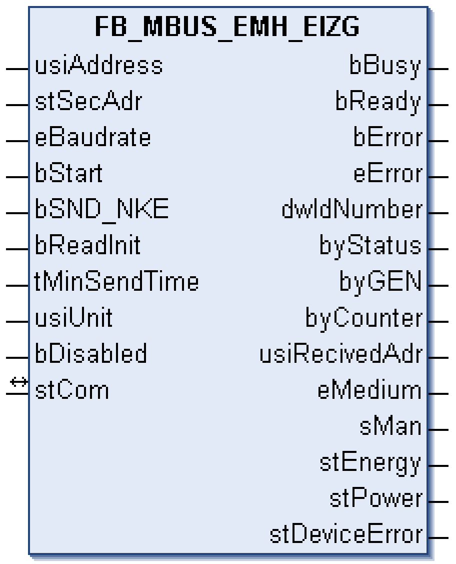
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.11.3 FB_MBUS_EMH_EIZG



This block is used to read electricity meters from EMH:

-EIZ-G

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
```

```
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stDeviceError     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

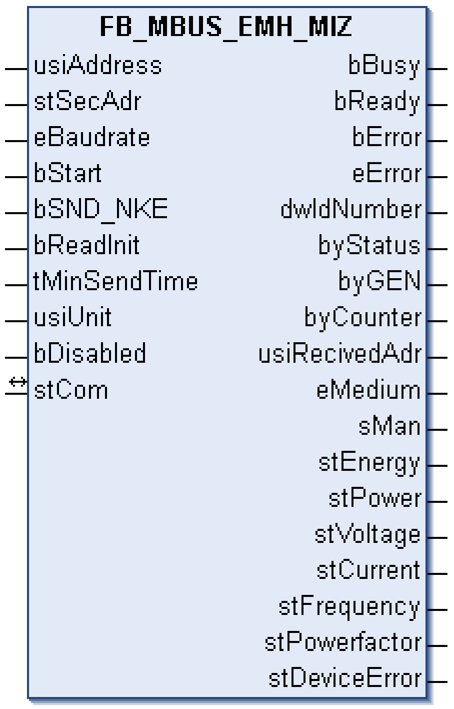
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.11.4 FB_MBUS_EMH_MIZ



This block is used to read electricity meters from EMH:

-MIZ

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stEnergy        : ST_MBus_Info;
stPower         : ST_MBus_Info;
stVoltage       : ST_MBus_Info;
stCurrent       : ST_MBus_Info;
stFrequency     : ST_MBus_Info;
stPowerfactor   : ST_MBus_Info;
stDeviceError   : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stVoltage: Current voltage (see ST_MBus_Info [▶ 205]).

stCurrent: Current current (see ST_MBus_Info [▶ 205]).

stFrequency: Current frequency (see ST_MBus_Info [▶ 205]).

stPowerfactor: Power factor (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

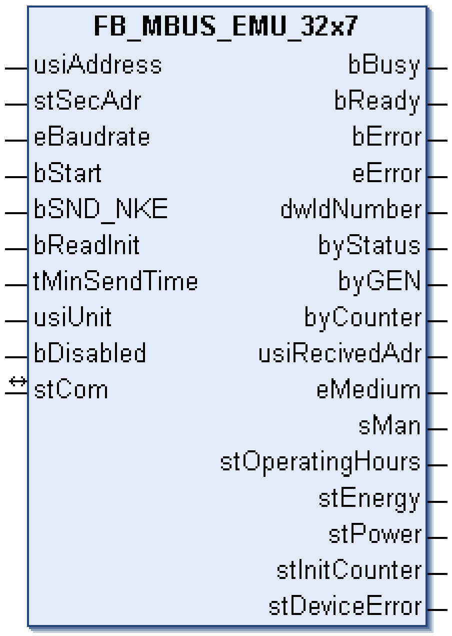
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.12 EMU overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
EMU	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7 [▸ 62]
	Electricity meter	EMU32x7	FB_MBUS_EMU_32x7_Option8 [▸ 65]
	Electricity meter	Allrounder 3/5	FB_MBUS_EMU_3_5_Allrounder [▸ 68]
	Electricity meter	DHZ 5/63	FB_MBUS_EMU_DHZ_5_63 [▸ 70]

4.1.12.1 FB_MBUS_EMU_32x7



This block is used to read electricity meters from EMU:

-EMU32.x7

Only the standard meter data will be read. The meter transmits this data in the standard EMU parameterization, or if the parameter set is set in the device to 00000 hexadecimal. Please refer to the meter documentation for further information regarding this.

In the normal version, the current consumption of the M-Bus interface is equivalent to 3 standard loads. If an M-Bus master interface is used that is designed, for example, for up to 120 standard loads, a maximum of 40 EMU M-Bus meters can be connected. The meter can optionally be supplied with 230 V. The current consumption of the M-Bus interface is then equivalent to one standard load.

The transmission of data from the EMU meter to the M-Bus protocol computer only works if the EMU meter is connected to at least two phases of the mains voltage network.

The EMU meter transmits current data to the device’s M-Bus interface every 40 seconds, so that the readout data is approx. 40 - 45 seconds old.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stOperatingHours  : ST_MBus_Info;
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stInitCounter     : ST_MBus_Info;
stDeviceError     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stOperatingHours: Operating hours of the EMU meter (see ST_MBus_Info [▸ 205]).

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stInitCounter: Number of power failures at the EMU meter (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

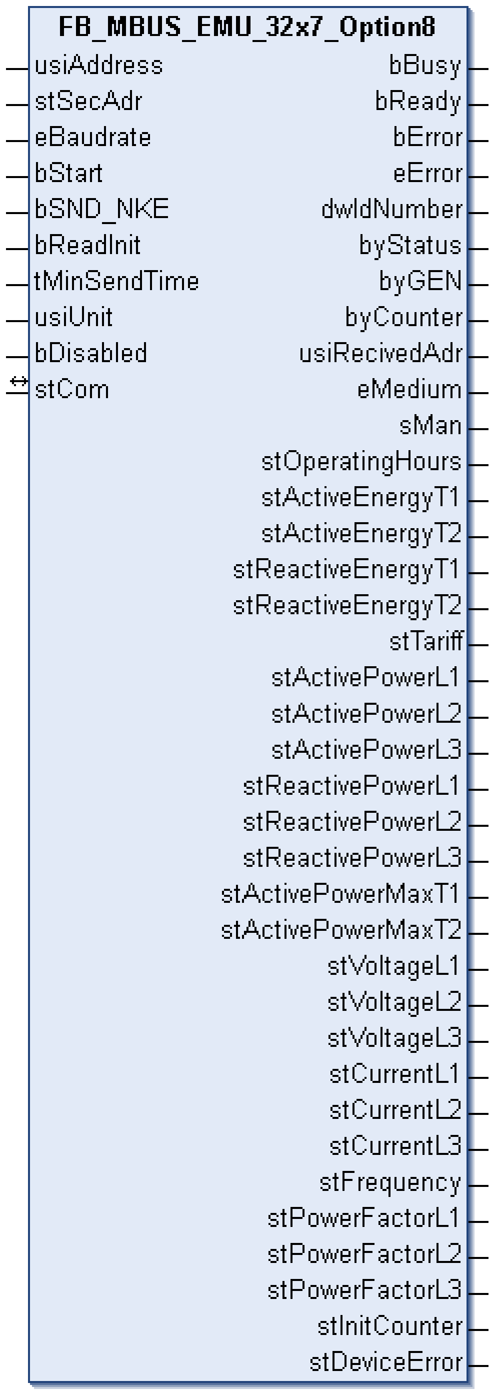
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.12.2 FB_MBUS_EMU_32x7_Option8



This block is used to read electricity meters from EMU:

-EMU32.x7

The parameter set must be set in the device to 70000 hexadecimal (variant 8) in order to read out this data. Please refer to the meter documentation for further information regarding this.

In the normal version, the current consumption of the M-Bus interface is equivalent to 3 standard loads. If an M-Bus master interface is used that is designed, for example, for up to 120 standard loads, a maximum of 40 EMU M-Bus meters can be connected. The meter can optionally be supplied with 230 V. The current consumption of the M-Bus interface is then equivalent to one standard load.

The transmission of data from the EMU meter to the M-Bus protocol computer only works if the EMU meter is connected to at least two phases of the mains voltage network.

The EMU meter transmits current data to the device’s M-Bus interface every 40 seconds, so that the readout data is approx. 40 - 45 seconds old.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                : BOOL;
bReady               : BOOL;
bError               : BOOL;
eError               : E_MBUS_ERROR;
dwIdNumber           : DWORD;
byStatus             : BYTE;
byGEN                : BYTE;
byCounter            : BYTE;
usiRecivedAdr        : USINT;
eMedium              : E_MBUS_Medium;
sMan                 : STRING(3);
stOperatingHours     : ST_MBus_Info;
stActiveEnergyT1     : ST_MBus_Info;
stActiveEnergyT2     : ST_MBus_Info;
stReactiveEnergyT1   : ST_MBus_Info;
stReactiveEnergyT2   : ST_MBus_Info;
stTariff             : ST_MBus_Info;
stActivePowerL1      : ST_MBus_Info;
stActivePowerL2      : ST_MBus_Info;
stActivePowerL3      : ST_MBus_Info;
stReactivePowerL1    : ST_MBus_Info;
stReactivePowerL2    : ST_MBus_Info;
stReactivePowerL3    : ST_MBus_Info;
stActivePowerMaxT1   : ST_MBus_Info;
stActivePowerMaxT2   : ST_MBus_Info;
stVoltageL1          : ST_MBus_Info;
stVoltageL2          : ST_MBus_Info;
stVoltageL3          : ST_MBus_Info;
stCurrentL1          : ST_MBus_Info;
stCurrentL2          : ST_MBus_Info;
stCurrentL3          : ST_MBus_Info;
stFrequency          : ST_MBus_Info;
stPowerFactorL1      : ST_MBus_Info;
stPowerFactorL2      : ST_MBus_Info;
```

```
stPowerFactorL3     : ST_MBus_Info;
stInitCounter       : ST_MBus_Info;
stDeviceError       : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stOperatingHours: Operating hours of the EMU meter (see ST_MBus_Info [▸ 205]).

stActiveEnergyT1: Meter reading, active energy tariff 1 (see ST_MBus_Info [▸ 205]).

stActiveEnergyT2: Meter reading, active energy tariff 2 (see ST_MBus_Info [▸ 205]).

stReactiveEnergyT1: Meter reading, reactive energy tariff 1 (see ST_MBus_Info [▸ 205]).

stReactiveEnergyT2: Meter reading, reactive energy tariff 2 (see ST_MBus_Info [▸ 205]).

stTariff: Current tariff (see ST_MBus_Info [▸ 205]).

stActivePowerL1: Instantaneous consumption, effective power L1 (see ST_MBus_Info [▸ 205]).

stActivePowerL2: Instantaneous consumption, effective power L2 (see ST_MBus_Info [▸ 205]).

stActivePowerL3: Instantaneous consumption, effective power L3 (see ST_MBus_Info [▸ 205]).

stReactivePowerL1: Instantaneous consumption, reactive power L1 (see ST_MBus_Info [▸ 205]).

stReactivePowerL2: Instantaneous consumption, reactive power L2 (see ST_MBus_Info [▸ 205]).

stReactivePowerL3: Instantaneous consumption, reactive power L3 (see ST_MBus_Info [▸ 205]).

stActivePowerMaxT1: Maximum effective power tariff 1 (see ST_MBus_Info [▸ 205]).

stActivePowerMaxT2: Maximum effective power tariff 2 (see ST_MBus_Info [▸ 205]).

stVoltageL1: Current voltage L1 (see ST_MBus_Info [▸ 205]).

stVoltageL2: Current voltage L2 (see ST_MBus_Info [▸ 205]).

stVoltageL3: Current voltage L3 (see ST_MBus_Info [▸ 205]).

stCurrentL1: Current current L1 (see ST_MBus_Info [▸ 205]).

stCurrentL2: Current current L2 (see ST_MBus_Info [▸ 205]).

stCurrentL3: Current current L3 (see ST_MBus_Info [▸ 205]).

stFrequency: Current mains frequency (see ST_MBus_Info [▸ 205]).

stPowerFactorL1: Current form factor phase L1 (cos Phi) (see ST_MBus_Info [▶ 205]).

stPowerFactorL2: Current form factor phase L2 (cos Phi) (see ST_MBus_Info [▶ 205]).

stPowerFactorL3: Current form factor phase L3 (cos Phi) (see ST_MBus_Info [▶ 205]).

stInitCounter: Number of power failures at the EMU meter (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

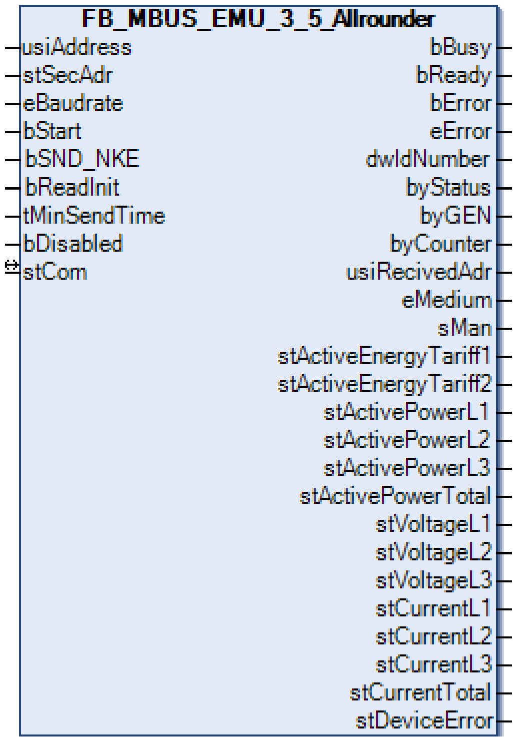
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.12.3 FB_MBUS_EMU_3_5_Allrounder



This block is used to read electricity meters from EMU.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
```

```
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300..9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                  : BOOL;
bReady                 : BOOL;
bError                 : BOOL;
eError                 : E_MBUS_ERROR;
dwIdNumber             : DWORD;
byStatus               : BYTE;
byGEN                  : BYTE;
byCounter              : BYTE;
usiRecivedAdr          : USINT;
eMedium                : E_MBUS_Medium;
sMan                   : STRING(3);
stActiveEnergyTariff1  : ST_MBus_Info;
stActiveEnergyTariff2  : ST_MBus_Info;
stActivePowerL1        : ST_MBus_Info;
stActivePowerL2        : ST_MBus_Info;
stActivePowerL3        : ST_MBus_Info;
stActivePowerTotal     : ST_MBus_Info;
stVoltageL1            : ST_MBus_Info;
stVoltageL2            : ST_MBus_Info;
stVoltageL3            : ST_MBus_Info;
stCurrentL1            : ST_MBus_Info;
stCurrentL2            : ST_MBus_Info;
stCurrentL3            : ST_MBus_Info;
stCurrentTotal         : ST_MBus_Info;
stDeviceError          : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stActiveEnergyTariff1: Active energy tariff 1 (see ST_MBus_Info [▶ 205]).

stActiveEnergyTariff2: Active energy tariff 2 (see ST_MBus_Info [▶ 205]).

stActivePowerL1: Effective power L1 (see ST_MBus_Info [▶ 205]).

stActivePowerL2: Effective power L2 (see ST_MBus_Info [▶ 205]).

stActivePowerL3: Effective power L3 (see ST_MBus_Info [▶ 205]).

stActivePowerTotal: Total effective power (see ST_MBus_Info [▶ 205]).

stVoltageL1: Voltage L1 (see ST_MBus_Info [▶ 205]).

stVoltageL2: Voltage L2 (see ST_MBus_Info [▶ 205]).

stVoltageL3: Voltage L3 (see ST_MBus_Info [▶ 205]).

stCurrentL1: Current intensity L1 (see ST_MBus_Info [▶ 205]).

stCurrentL2: Current intensity L2 (see ST_MBus_Info [▶ 205]).

stCurrentL3: Current intensity L3 (see ST_MBus_Info [▶ 205]).

stCurrentTotal: Total current intensity (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

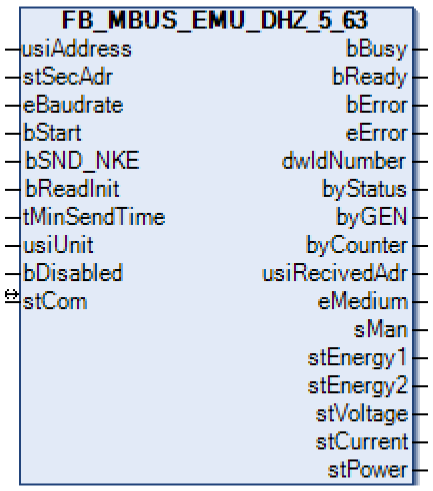
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.12.4 FB_MBUS_EMU_DHZ_5_63



This block is used to read electricity meters from EMU.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300..9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy1         : ST_MBus_Info;
stEnergy2         : ST_MBus_Info;
stVoltage         : ST_MBus_Info;
stCurrent         : ST_MBus_Info;
stPower           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy1: Effective energy 1 (see ST_MBus_Info [▸ 205]).

stEnergy2: Effective energy 2 (resettable) (see ST_MBus_Info [▸ 205]).

stVoltage: Mains voltage (see ST_MBus_Info [▸ 205]).

stCurrent: Instantaneous current (see ST_MBus_Info [▸ 205]).

stPower: Instantaneous active power (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

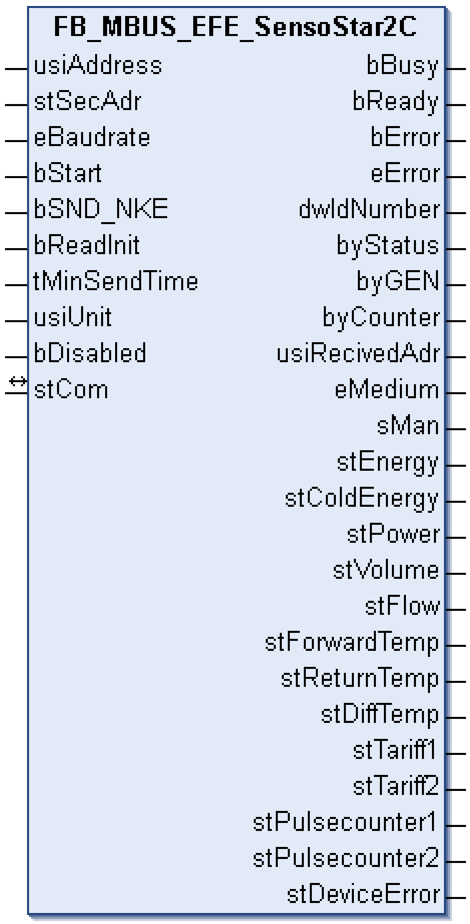
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.13 Engelmann overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Engelmann	Heat meter	Sensostar 2C	FB_MBUS_EFF_SensoStar2C [▸ 73]

4.1.13.1 FB_MBUS_EFE_SensoStar2C



This block is used to read heat meters from Engelmann:

-SENSOSTAR 2C

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: (see E_MBUS_baud rate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stColdEnergy       : ST_MBus_Info;
stPower            : ST_MBus_Info;
stVolume           : ST_MBus_Info;
stFlow             : ST_MBus_Info;
stForwardTemp      : ST_MBus_Info;
stReturnTemp       : ST_MBus_Info;
stDiffTemp         : ST_MBus_Info;
stTariff1          : ST_MBus_Info;
stTariff2          : ST_MBus_Info;
stPulsecounter1    : ST_MBus_Info;
stPulsecounter2    : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stColdEnergy: Energy consumption meter reading (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stTariff1: Tariff register 1 (see ST_MBus_Info [▸ 205]).

stTariff2: Tariff register 2 (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

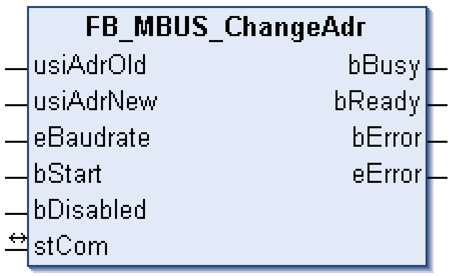
Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14 General device blocks

Manufacturer	Type	Device	Block
General	Electricity meter	all electricity meters	FB_MBUS_General_Electricity [▸ 78]
	Heat meter	all heat meters	FB_MBUS_General_Heat [▸ 82]
	Water meter	all water meters	FB_MBUS_General_Water [▸ 87]
	Raw data of the first telegram	all	FB_MBUS_RawData [▸ 89]
	max. 40 values from the first telegram	all	FB_MBUS_General [▸ 77]
	All telegrams, all values	all	FB_MBUS_General_Ext [▸ 80]
	values parameterizable	all	FB_MBUS_General_Param [▸ 84]
	universal send blocks	all	FB_MBUS_General_Send [▸ 86]
	scan block	all	FB_MBUS_Scan [▸ 90]
	Change address	all	FB_MBUS_ChangeAdr [▸ 76]

4.1.14.1 FB_MBUS_ChangeAdr



This block can be used to change the primary address.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

VAR_INPUT

```
usiAdrOld : USINT;
usiAdrNew : USINT;
eBaudrate : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart    : BOOL;
bDisabled : BOOL:=FALSE;
```

usiAdrOld: Old primary address.

usiAdrNew: New primary address.

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge at this input triggers a change of the primary address of the counter.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy  : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

VAR_IN_OUT

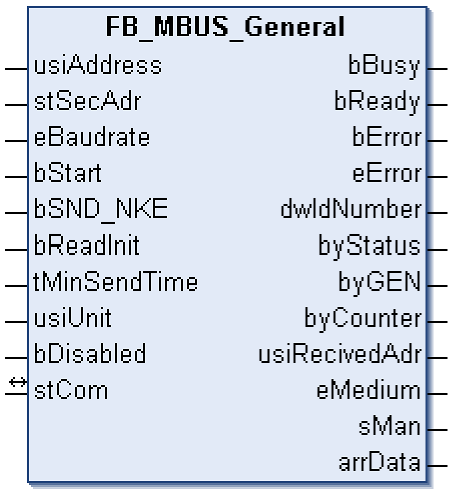
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.2 FB_MBUS_General



This block is used for reading any M-Bus devices. The variable arrData [▸ 204] supplies a maximum of cMBUS_MaxData [▸ 208] values for the first telegram. String values and manufacturer-specific information are not shown correctly.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
```

```
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
arrData           : ARRAY [1..cMBUS_MaxData] OF ST_MBus_Data;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

arrData: Up to cMBUS_MaxData [▶ 208] values of the first telegram. The meaning of the values is explained in the M-Bus protocol for the device.

VAR_IN_OUT

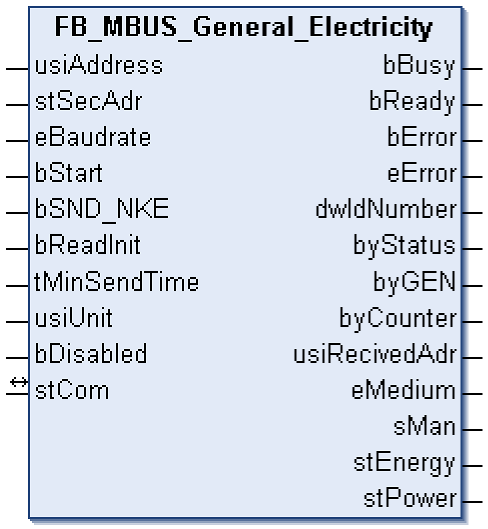
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.3 FB_MBUS_General_Electricity



This block is used to read electricity meters.

Not all electricity meters automatically send power data. In this case the corresponding structure remains empty.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy               : BOOL;
bReady              : BOOL;
bError              : BOOL;
eError              : E_MBUS_ERROR;
dwIdNumber          : DWORD;
byStatus            : BYTE;
byGEN               : BYTE;
byCounter           : BYTE;
usiRecivedAdr       : USINT;
eMedium             : E_MBUS_Medium;
sMan                : STRING(3);
stEnergy            : ST_MBus_Info;
stPower             : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

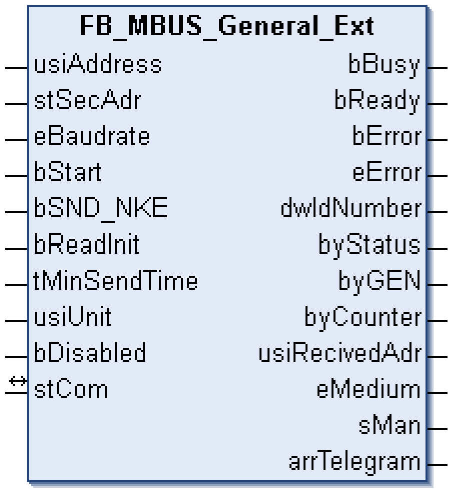
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.4 FB_MBUS_General_Ext



Some M-Bus devices send values distributed over several telegrams. This block can be used to read all telegrams from any M-Bus devices.

The variable *arrTelegram[1..cMBUS_MaxTelegrams].arrData[1..cMBUS_MaxData]* supplies a maximum of cMBUS_MaxData [▶ 208] data from a maximum of cMBUS_MaxTelegrams [▶ 208] telegrams. String values and manufacturer-specific information are not shown correctly.

The number of telegrams to be read can be changed with the constant cMBUS_MaxTelegrams [▶ 208].

The number of data per telegram to be read can be changed with the constant cMBUS_MaxData [▶ 208].

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
```

```
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
arrTelegram       : ARRAY [1..cMBUS_MaxTelegrams] OF ST_MBus_Data2;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

arrTelegram: Up to cMBUS_MaxTelegrams [▸ 208] telegrams (see ST_MBus_data [▸ 205]). The meaning of the values is explained in the M-Bus protocol for the device.

VAR_IN_OUT

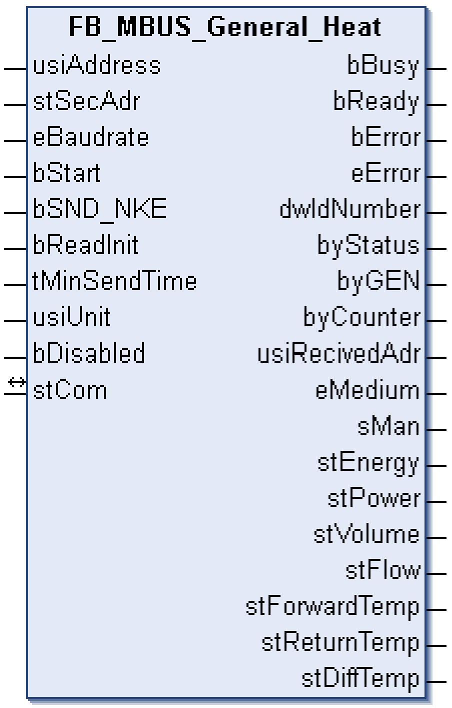
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.5 FB_MBUS_General_Heat



This block is used to read heat meters.

Many heat meters do not send all values. In this case the corresponding structures remain empty.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

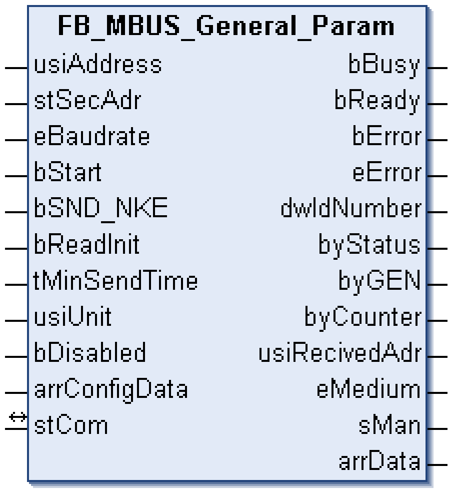
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.6 FB_MBUS_General_Param



This block is used for reading any M-Bus devices. The variable arrData [▸ 204] supplies cMBUS_MaxDataParam [▸ 208] values.

These values can be parameterized in the input array *arrConfigData*. String values and manufacturer-specific information are not shown correctly.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
arrConfigData : ARRAY [1..cMBUS_MaxDataParam] OF WORD;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

arrConfigData: Up to cMBUS_MaxDataParam input parameters for specifying which values are to be displayed in the output array *arrData* (see Global_Variables_MBUS [▸ 208]).

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
arrData            : ARRAY [1..cMBUS_MaxDataParam] OF ST_MBus_Data;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

arrData: Up to cMBUS_MaxDataParam values (see Global_Variables_MBUS [▸ 208]). The values can be configured via the input variable *arrConfigData*. The meaning of the values is explained in the M-Bus protocol for the device.

VAR_IN_OUT

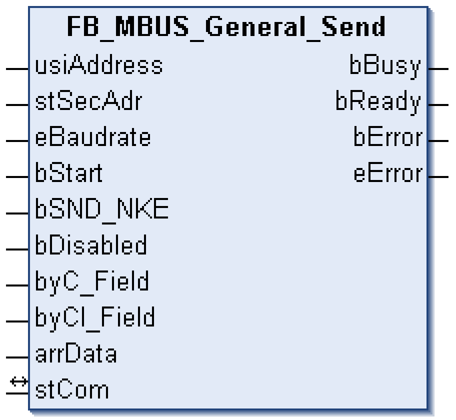
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.7 FB_MBUS_General_Send



This block serves to send data to any M-Bus devices. (for example, the primary address of the meter can be changed with this block)

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the function block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bDisabled    : BOOL := FALSE;
byC_Field    : USINT := 16#53;
byCI_Field   : USINT := 16#51;
arrData      : ARRAY [0..240] OF BYTE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bDisabled: TRUE = deselection of the block.

byC_Field: C-field / function field.

byCI_Field: CI-field / ID field.

arrData: The data to be sent must be written to this variable (see long block [▶ 11]).

VAR_OUTPUT

```
bBusy  : BOOL;
bReady : BOOL;
bError : BOOL;
eError : E_MBUS_ERROR;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

VAR_IN_OUT

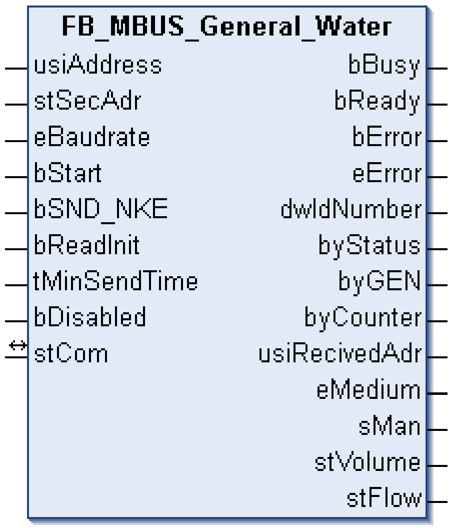
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.8 FB_MBUS_General_Water



This block is used to read water meters.

Not all water meters automatically send the flow rate. In this case the corresponding structure remains empty.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                : BOOL;
bReady               : BOOL;
bError               : BOOL;
eError               : E_MBUS_ERROR;
dwIdNumber           : DWORD;
byStatus             : BYTE;
byGEN                : BYTE;
byCounter            : BYTE;
usiRecivedAdr        : USINT;
eMedium              : E_MBUS_Medium;
sMan                 : STRING(3);
stVolume             : ST_MBus_Info;
stFlow               : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stComIn   : ST_KL6781inData22B;
stComOut  : ST_KL6781outData22B;
stCom     : ST_MBUS_Communication;
```

stComIn: Process image of the inputs (see ST_KL6781inData22B [▶ 203]).

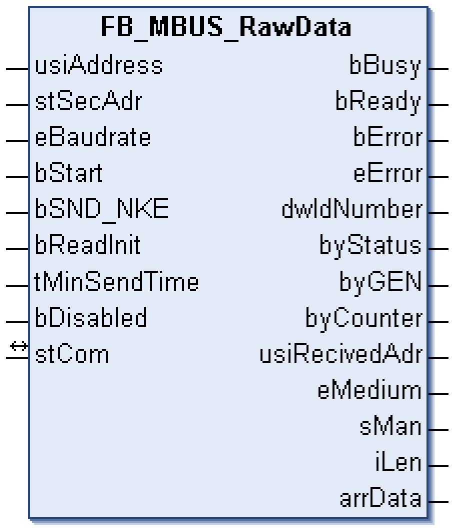
stComOut: Process image of the outputs (see ST_KL6781outData22B [▶ 203]).

stCom: This structure is used to link the block with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.9 FB_MBUS_RawData



This block is used for reading any M-Bus devices. The variable *arrData* supplies the raw data of the M-Bus device. Only the first telegram is evaluated.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
```

```
sMan              : STRING(3);
iLen              : INT;
arrData           : ARRAY [0..259] OF BYTE;
```

iLen: Number of transferred bytes.

arrData: Raw data of the first telegram.

VAR_IN_OUT

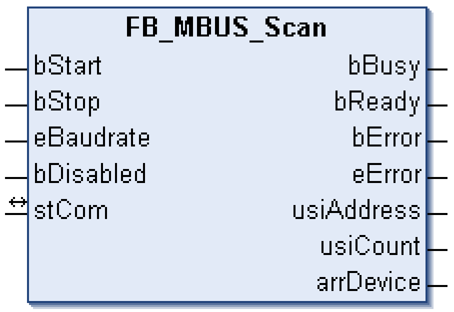
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.14.10 FB_MBUS_Scan



This block can be used to scan the M-Bus bus. All primary addresses (0..250) are queried successively. The array *arrDevice* is used to show certain device information.

Only the primary address is used for scanning.

The primary address [▶ 11] of all devices must be set.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

VAR_INPUT

```
bStart        : BOOL;
bStop         : BOOL;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bDisabled     : BOOL := FALSE;
```

bStart: The search is initiated with a positive edge at this input.

bStop: The search is stopped with a positive edge at this input.

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 Baud.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
usiAddress        : USINT;
usiCount          : USINT;
arrDevice         : ARRAY [0..250] OF ST_MBus_Scan;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

usiCount: Number of detected valid devices.

arrDevice: Information about the detected devices (see ST_MBus_scan [▸ 206]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

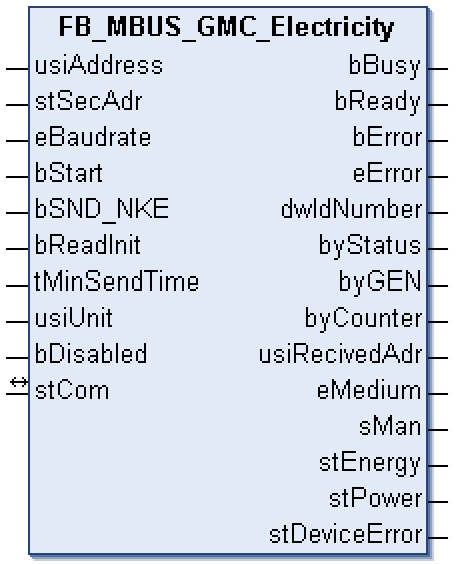
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.15 Gossen Metrawatt overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Gossen Metrawatt	Electricity meter	U128x	FB_MBUS_GMC_Electricity [▸ 92]
	Electricity meter	U138x	FB_MBUS_GMC_Electricity [▸ 92]

4.1.15.1 FB_MBUS_GMC_Electricity



This block is used to read electricity meters from Gossen Metrawatt:

-U128x

-U138x

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
```

```
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stDeviceError    : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

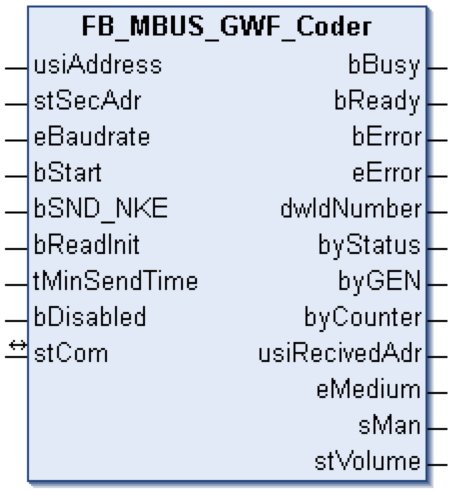
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.16 GWF overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
GWF	Water meter		FB_MBUS_GWF_Coder [▸ 94]
	Gas meter	S1	FB_MBUS_GWF_Coder [▸ 94]
	Gas meter	Z1	FB_MBUS_GWF_Coder [▸ 94]

4.1.16.1 FB_MBUS_GWF_Coder



This block is used to read meters from GWF:

-Water meter

-Gas meter S1

-Gas meter Z1

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stVolume          : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

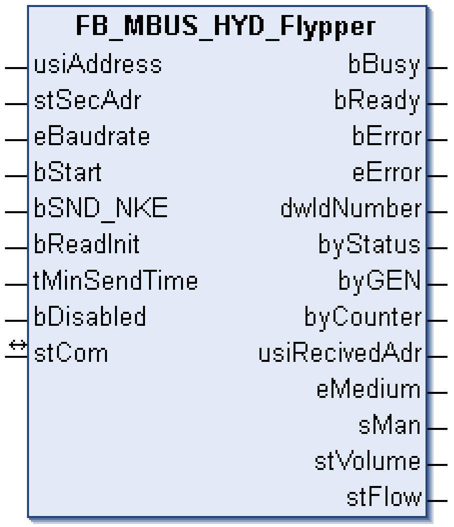
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.17 Hydrometer overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Hydrometer	2 pulse inputs	HYDRO-PORT Pulse	FB_MBUS_HYD_PortPulse [▸ 100]
	2 analog inputs + 1 temperature sensor	HYDRO-PORT Analog	FB_MBUS_HYD_PortAnalog [▸ 98]
	Water meter	Flypper	FB_MBUS_HYD_Flypper [▸ 96]
	Heat meter	Sharky 773	FB_MBUS_HYD_Sharky [▸ 102], FB_MBUS_HYD_Sharky_00 [▸ 104]
	Heat meter	Sharky 775	FB_MBUS_HYD_Sharky [▸ 102], FB_MBUS_HYD_Sharky_00 [▸ 104]

4.1.17.1 FB_MBUS_HYD_Flypper



This block is used to read water meters from Hydrometer:

-Flypper

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stVolume           : ST_MBus_Info;
stFlow             : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

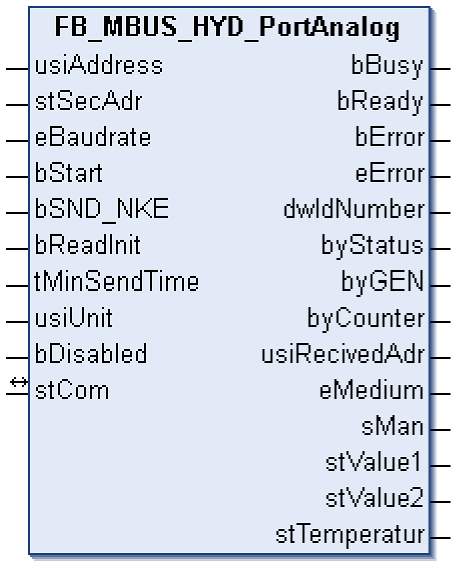
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.17.2 FB_MBUS_HYD_PortAnalog



This block is used for reading energy meters with analog output from Hydrometer:

-HYDRO-PORT analog (2x0/4-20 mA / 1xPT temperature sensor)

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy          : BOOL;
bReady         : BOOL;
bError         : BOOL;
eError         : E_MBUS_ERROR;
dwIdNumber     : DWORD;
```

```
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue1          : ST_MBus_Info;
stValue2          : ST_MBus_Info;
stTemperatur      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stValue1: Meter reading 1 (see ST_MBus_Info [▸ 205]).

stValue2: Meter reading 2 (see ST_MBus_Info [▸ 205]).

stTemperatur: temperature (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

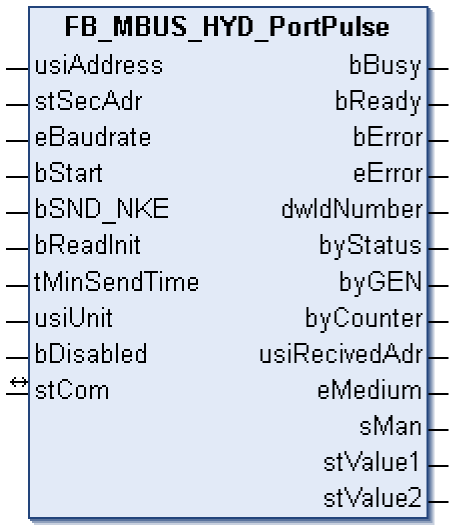
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.17.3 FB_MBUS_HYD_PortPulse



This block is used for reading energy meters with pulse output from Hydrometer:

-HYDRO-PORT Pulse

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
```

```
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue1          : ST_MBus_Info;
stValue2          : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stValue1: Meter reading 1 (see ST_MBus_Info [▶ 205]).

stValue2: Meter reading 2 (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

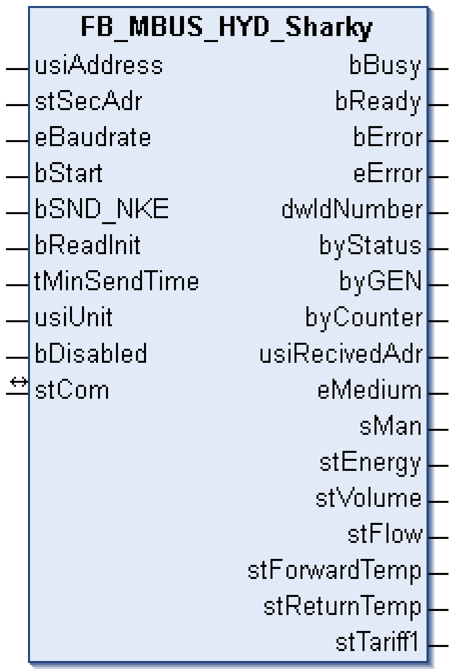
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.17.4 FB_MBUS_HYD_Sharky



This block is used to read energy meters from:

Hydrometer:

-Sharky 773

-Sharky 775

-ENERGY INT 6

Brunata:

-Brunata Optuna H (775)

Aquametro:

-AMNTRONIC SONIC D

Only the most common values (see "VAR_OUTPUT") of the telegrams:

00 (Application Reset-Subcode 00 / All)
10 (Application Reset-Subcode 10 / User data)
20 (Application Reset-Subcode 20 / Simple billing)
30 (Application Reset-Subcode 30 / Enhanced billing)
40 (Application Reset-Subcode 40 / Multi tariff billing)
or 50 (Application Reset-Subcode 50 / Instant values)

are read.

The device is not switched to these telegrams; it must be set to one of these telegrams.

The block FB_MBUS_HYD_Sharky_00() [▸ 104] can be used if further data are required, or the block FB_MBUS_General_Send() [▸ 86] can be used to select the required telegram, and the block FB_MBUS_General() [▸ 77] can be used to read all data of the respective telegram.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stVolume          : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stForwardTemp     : ST_MBus_Info;
stReturnTemp      : ST_MBus_Info;
stTariff1         : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▶ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▶ 205]).

stTariff1: Energy consumption tariff 1 (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

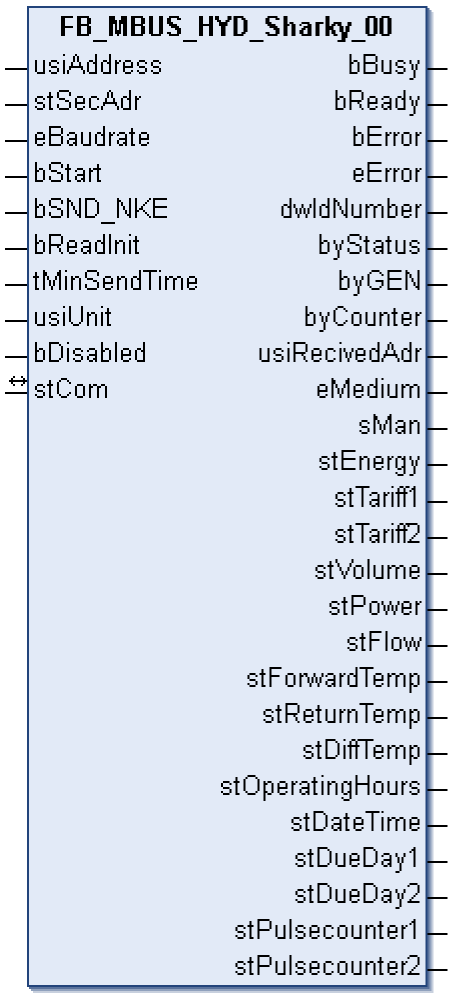
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.17.5 FB_MBUS_HYD_Sharky_00



This block is used to read energy meters from:

Hydrometer:

-Sharky 773

-Sharky 775

-ENERGY INT 6

Brunata:

-Brunata Optuna H (775)

Aquametro:

-AMNTRONIC SONIC D

All values of telegram 00 (application reset subcode 00 / All) are read. The device automatically switches to the corresponding mode.

stPulsecounter1 and stPulsecounter2 are only output if the pulse module is connected.

If further telegrams are required, the block FB_MBUS_General_Send() [▶ 86] can be used to select the required telegram, and the block FB_MBUS_General() [▶ 77] can be used to read all data of the respective telegram.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
```

```
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stTariff1         : ST_MBus_Info;
stTariff2         : ST_MBus_Info;
stVolume          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stForwardTemp     : ST_MBus_Info;
stReturnTemp      : ST_MBus_Info;
stDiffTemp        : ST_MBus_Info;
stOperatingHours : ST_MBus_Info;
stDateTime        : ST_MBus_Info;
stDueDay1         : ST_MBUS_DueDayHYD1;
stDueDay2         : ST_MBUS_DueDayHYD1;
stPulsecounter1   : ST_MBus_Info;
stPulsecounter2   : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stTariff1: Meter reading tariff 1 (see ST_MBus_Info [▸ 205]).

stTariff2: Meter reading tariff 2 (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stOperatingHours: Current operating hours (see ST_MBus_Info [▸ 205]).

stDateTime: Current date, time (see ST_MBus_Info [▸ 205]).

stDueDay1: Values cutoff date 1 (see ST_MBUS_DueDayHYD1 [▸ 207]).

stDueDay2: Values cutoff date 2 (see ST_MBUS_DueDayHYD1 [▸ 207]).

stPulsecounter1: Meter reading pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Meter reading pulse counter 2 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

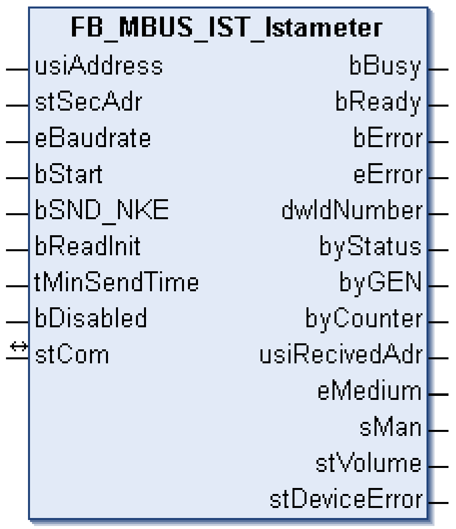
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.18 ista overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
ista	Water meter	domaqua® m	FB_MBUS_IST_Istameter [▸ 107]
	Water meter	istameter® m	FB_MBUS_IST_Istameter [▸ 107]
	Water meter	istameter III	FB_MBUS_IST_IstameterIII [▸ 109]
	Pulse counter	pulsonic II	FB_MBUS_IST_PulsonicII [▸ 111]
	Heat meter	sensonic II	FB_MBUS_IST_SensonicII [▸ 113]

4.1.18.1 FB_MBUS_IST_Istameter



This block is used to read water meters from Ista:

-istameter® m

-domaqua® m

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

The devices are supplied from a battery. The number of read operations is therefore limited. An internal meter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stVolume         : ST_MBus_Info;
stDeviceError    : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

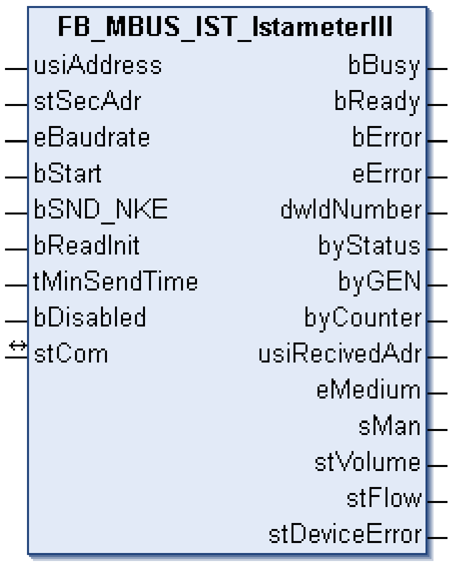
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.18.2 FB_MBUS_IST_IstameterIII



This block is used to read water meters from Ista:

-istameter III

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

The devices are supplied from a battery. The number of read operations is therefore limited. An internal meter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stVolume          : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stDeviceError     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

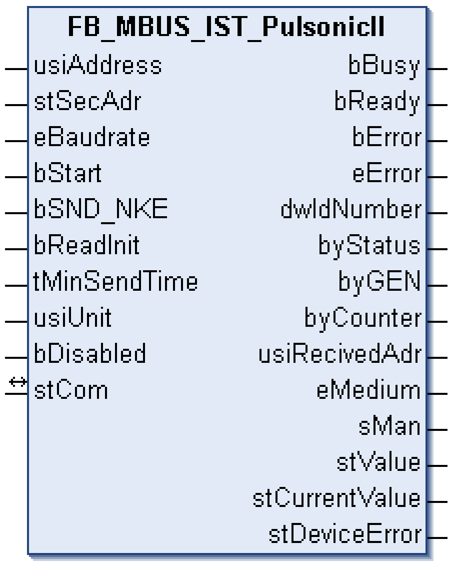
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.18.3 FB_MBUS_IST_PulsonicII



This block is used to read energy meters with pulse output from Ista:

-Pulsonic II

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Maximum number of readings

The devices are supplied from a battery. The number of read operations is therefore limited. An internal counter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stValue            : ST_MBus_Info;
stCurrentValue     : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stValue: Current consumption value (see ST_MBus_Info [▶ 205]).

stCurrentValue: Current flow rate / power (see ST_MBus_Info [▶ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

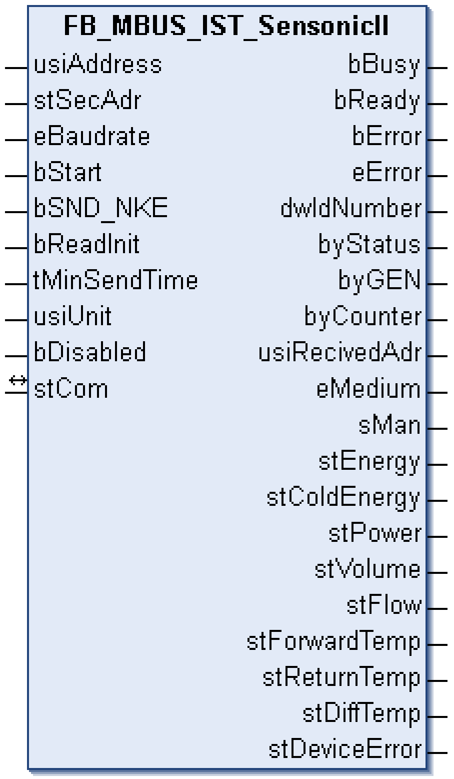
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.18.4 FB_MBUS_IST_SensonicII



This block is used to read heat meters from Ista:

-Sensonic II

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Maximum number of readings

The devices are supplied from a battery. The number of read operations is therefore limited. An internal counter prevents communication exceeding 96 times per day on average. The user must make sure that excessive queries are prevented.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stColdEnergy       : ST_MBus_Info;
stPower            : ST_MBus_Info;
stVolume           : ST_MBus_Info;
stFlow             : ST_MBus_Info;
stForwardTemp      : ST_MBus_Info;
stReturnTemp       : ST_MBus_Info;
stDiffTemp         : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stColdEnergy: Meter reading, cooling energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

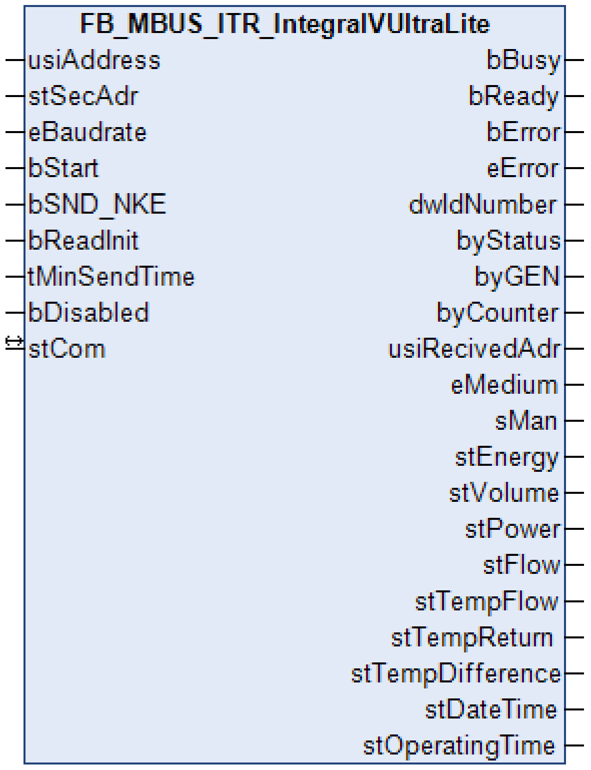
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.19 Itron

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Function block
Itron	Energy meter	Integral-V UltraLite	FB_MBUS_ITR_IntegralVUltraLite [▸ 115]

4.1.19.1 FB_MBUS_ITR_IntegralVUltraLite



This function block is used to read energy meters from Itron.

The function block can only be executed together with the function block FB_MBUSKL6781() [▸ 19].

Functionality of the function block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the function block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stVolume          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stTempFlow        : ST_MBus_Info;
stTempReturn      : ST_MBus_Info;
stTempDifference  : ST_MBus_Info;
stDateTime        : ST_MBus_Info;
stOperatingTime   : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Current energy (see ST_MBus_Info [▶ 205]).

stVolume: Current volume (see ST_MBus_Info [▶ 205]).

stPower: Current output (see ST_MBus_Info [▶ 205]).

stFlow: Current flow rate (see ST_MBus_Info [▶ 205]).

stTempFlow: Current flow temperature (see ST_MBus_Info [▶ 205]).

stTempReturn: Current return temperature (see ST_MBus_Info [▶ 205]).

stTempDifference: Current temperature difference (see ST_MBus_Info [▶ 205]).

stDateTime: Date and time (see ST_MBus_Info [▶ 205]).

stOperatingTime: Operating time (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

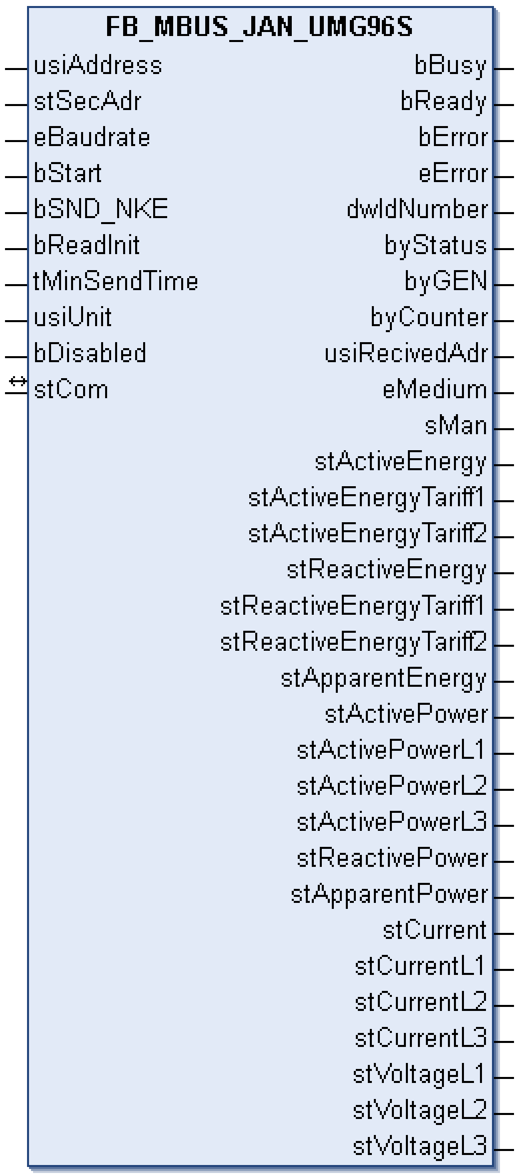
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.32	Tc2_MBus from 3.4.6.0

4.1.20 Janitza overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Janitza	Electricity meter	UMG96S	FB_MBUS_JAN_UMG96S [▶ 118]

4.1.20.1 FB_MBUS_JAN_UMG96S



This block is used to read electricity meters from Janitza:

-UMG96S

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                    : BOOL;
bReady                   : BOOL;
bError                   : BOOL;
eError                   : E_MBUS_ERROR;
dwIdNumber               : DWORD;
byStatus                 : BYTE;
byGEN                    : BYTE;
byCounter                : BYTE;
usiRecivedAdr            : USINT;
eMedium                  : E_MBUS_Medium;
sMan                     : STRING(3);
stActiveEnergy           : ST_MBus_Info;
stActiveEnergyTariff1    : ST_MBus_Info;
stActiveEnergyTariff2    : ST_MBus_Info;
stReactiveEnergy         : ST_MBus_Info;
stReactiveEnergyTariff1  : ST_MBus_Info;
stReactiveEnergyTariff2  : ST_MBus_Info;
stApparentEnergy         : ST_MBus_Info;
stActivePower            : ST_MBus_Info;
stActivePowerL1          : ST_MBus_Info;
stActivePowerL2          : ST_MBus_Info;
stActivePowerL3          : ST_MBus_Info;
stReactivePower          : ST_MBus_Info;
stApparentPower          : ST_MBus_Info;
stCurrent                : ST_MBus_Info;
stCurrentL1              : ST_MBus_Info;
stCurrentL2              : ST_MBus_Info;
stCurrentL3              : ST_MBus_Info;
stVoltageL1              : ST_MBus_Info;
stVoltageL2              : ST_MBus_Info;
stVoltageL3              : ST_MBus_Info;
```

stActiveEnergy: Active energy without backstop (telegram 2, data point 14) (see ST_MBus_Info [▸ 205]).

stActiveEnergyTariff1: Active energy, relative (telegram 2, data point 15) (see ST_MBus_Info [▸ 205]).

stActiveEnergyTariff2: Active energy, supplied (telegram 2, data point 16) (see ST_MBus_Info [▸ 205]).

stReactiveEnergy: Reactive energy, inductive (telegram 2, data point 17) (see ST_MBus_Info [▸ 205]).

stReactiveEnergyTariff1: Reactive energy, capacitive (telegram 2, data point 18) (see ST_MBus_Info [▸ 205]).

stReactiveEnergyTariff2: Reactive energy, without backstop (telegram 2, data point 19) (see ST_MBus_Info [▸ 205]).

stApparentEnergy: Apparent energy (telegram 2, data point 20) (see ST_MBus_Info [▸ 205]).

stActivePower: Instantaneous consumption, power, total (telegram 2, data point 29) (see ST_MBus_Info [▸ 205]).

stActivePowerL1: Instantaneous consumption, power, phase L1 (telegram 2, data point 38) (see ST_MBus_Info [▸ 205]).

stActivePowerL2: Instantaneous consumption, power, phase L2 (telegram 2, data point 39) (see ST_MBus_Info [▶ 205]).

stActivePowerL3: Instantaneous consumption, power, phase L3 (telegram 2, data point 40) (see ST_MBus_Info [▶ 205]).

stReactivePower: Reactive power (telegram 2, data point 30) (see ST_MBus_Info [▶ 205]).

stApparentPower: Apparent power (telegram 2, data point 31) (see ST_MBus_Info [▶ 205]).

stCurrent: Current (telegram 2, data point 28) (see ST_MBus_Info [▶ 205]).

stCurrentL1: Current L1 (telegram 2, data point 35) (see ST_MBus_Info [▶ 205]).

stCurrentL2: Current L2 (telegram 2, data point 36) (see ST_MBus_Info [▶ 205]).

stCurrentL3: Current L3 (telegram 2, data point 37) (see ST_MBus_Info [▶ 205]).

stVoltageL1: Voltage L1 (telegram 2, data point 32) (see ST_MBus_Info [▶ 205]).

stVoltageL2: Voltage L2 (telegram 2, data point 33) (see ST_MBus_Info [▶ 205]).

stVoltageL3: Voltage L3 (telegram 2, data point 34) (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

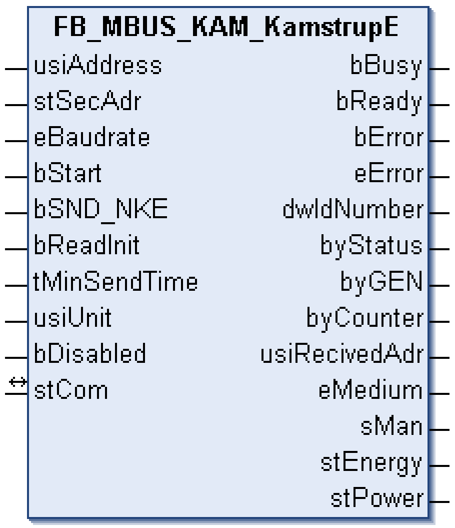
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21 Kamstrup overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Kamstrup	Electricity meter	Kamstrup 162	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 351	FB_MBUS_KAM_KamstrupE [▶ 121]
	Electricity meter	Kamstrup 382	FB_MBUS_KAM_KamstrupE [▶ 121]
	Heat/cold meter	Maxical III	FB_MBUS_KAM_Maxical_III [▶ 123]
	Heat/cold meter	Multical 401	FB_MBUS_KAM_Multical [▶ 125]
	Heat/cold meter	Multical 402	FB_MBUS_KAM_Multical402 [▶ 127]
	Water meter	Multical 41	FB_MBUS_KAM_Multical41 [▶ 129]
	Heat/cold meter	Multical 601	FB_MBUS_KAM_Multical601 [▶ 131]

4.1.21.1 FB_MBUS_KAM_KamstrupE



This block is used to read electricity meters from Kamstrup:

-Kamstrup 162

-Kamstrup 351

-Kamstrup 382

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

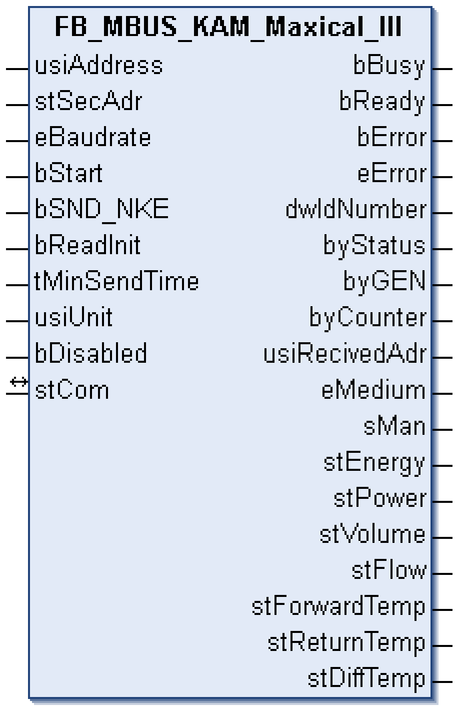
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21.2 FB_MBUS_KAM_Maxical_III



This block is used for reading heat/cold meters from Kamstrup:

-Maxical III

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stEnergy        : ST_MBus_Info;
stPower         : ST_MBus_Info;
stVolume        : ST_MBus_Info;
stFlow          : ST_MBus_Info;
stForwardTemp   : ST_MBus_Info;
stReturnTemp    : ST_MBus_Info;
stDiffTemp      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

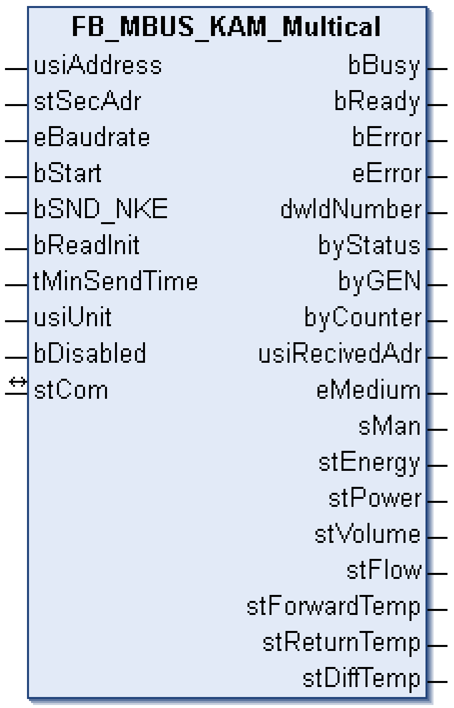
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21.3 FB_MBUS_KAM_Multical



This block is used for reading heat/cold meters from Kamstrup:

-Multical 401

-Multical 601

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stVolume          : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stForwardTemp     : ST_MBus_Info;
stReturnTemp      : ST_MBus_Info;
stDiffTemp        : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▶ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▶ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

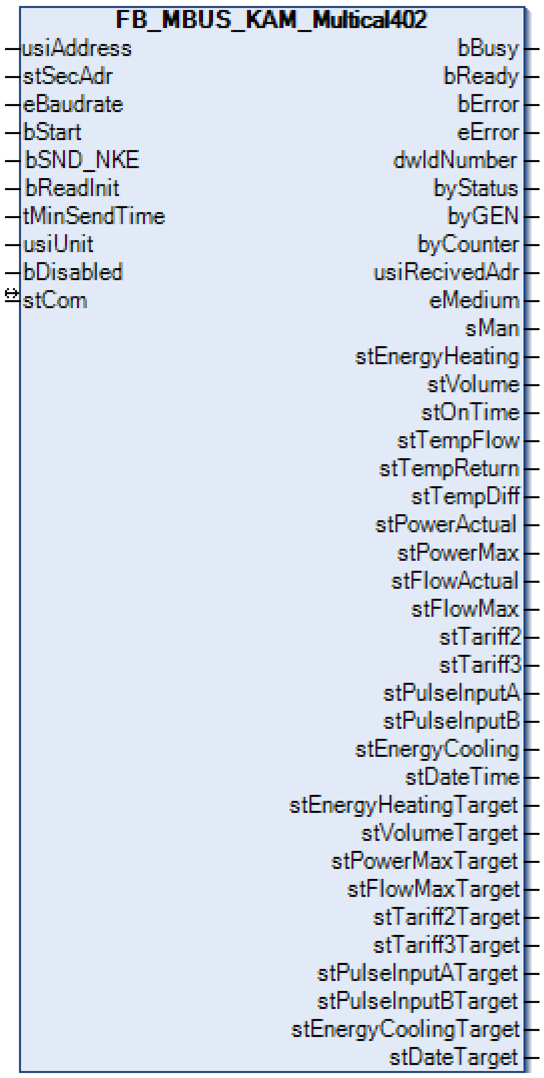
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21.4 FB_MBUS_KAM_Multical402



This block is used to read energy meters from Kamstrup.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                  : BOOL;
bReady                 : BOOL;
bError                 : BOOL;
eError                 : E_MBUS_ERROR;
dwIdNumber             : DWORD;
byStatus               : BYTE;
byGEN                  : BYTE;
byCounter              : BYTE;
usiRecivedAdr          : USINT;
eMedium                : E_MBUS_Medium;
sMan                   : STRING(3);
stEnergyHeating        : ST_MBus_Info;
stVolume               : ST_MBus_Info;
stOnTime               : ST_MBus_Info;
stTempFlow             : ST_MBus_Info;
stTempReturn           : ST_MBus_Info;
stTempDiff             : ST_MBus_Info;
stPowerActual          : ST_MBus_Info;
stPowerMax             : ST_MBus_Info;
stFlowActual           : ST_MBus_Info;
stFlowMax              : ST_MBus_Info;
stTariff2              : ST_MBus_Info;
stTariff3              : ST_MBus_Info;
stPulseInputA          : ST_MBus_Info;
stPulseInputB          : ST_MBus_Info;
stEnergyCooling        : ST_MBus_Info;
stDateTime             : ST_MBus_Info;
stEnergyHeatingTarget  : ST_MBus_Info;
stVolumeTarget         : ST_MBus_Info;
stPowerMaxTarget       : ST_MBus_Info;
stFlowMaxTarget        : ST_MBus_Info;
stTariff2Target        : ST_MBus_Info;
stTariff3Target        : ST_MBus_Info;
stPulseInputATarget    : ST_MBus_Info;
stPulseInputBTarget    : ST_MBus_Info;
stEnergyCoolingTarget  : ST_MBus_Info;
stDateTarget           : ST_MBus_Info;
```

stEnergyHeating: Heat energy (see ST_MBus_Info [▸ 205]).

stVolume: Water consumption from district heating (see ST_MBus_Info [▸ 205]).

stOnTime: Operating hours (see ST_MBus_Info [▸ 205]).

stTempFlow: Current flow temperature (see ST_MBus_Info [▸ 205]).

stTempReturn: Current return temperature (see ST_MBus_Info [▸ 205]).

stTempDiff: Current temperature difference (see ST_MBus_Info [▸ 205]).

stPowerActual: Current effective power (see ST_MBus_Info [▸ 205]).

stPowerMax: Effective power (max.) (see ST_MBus_Info [▸ 205]).

stFlowActual: Current flow rate (see ST_MBus_Info [▸ 205]).

stFlowMax: Flow rate (max.) (see ST_MBus_Info [▸ 205]).

stTariff2: Tariff 2 (see ST_MBus_Info [▸ 205]).

stTariff3: Tariff 3 (see ST_MBus_Info [▸ 205]).

stPulseInputA: Pulse input A (see ST_MBus_Info [▸ 205]).

stPulseInputB: Pulse input B (see ST_MBus_Info [▸ 205]).

stEnergyCooling: Cooling energy (see ST_MBus_Info [▸ 205]).

stDateTime: Date and time (see ST_MBus_Info [▸ 205]).

stEnergyHeatingTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stVolumeTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stPowerMaxTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stFlowMaxTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stTariff2Target: See manufacturer information (see ST_MBus_Info [▸ 205]).

stTariff3Target: See manufacturer information (see ST_MBus_Info [▸ 205]).

stPulseInputATarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stPulseInputBTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stEnergyCoolingTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

stDateTarget: See manufacturer information (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

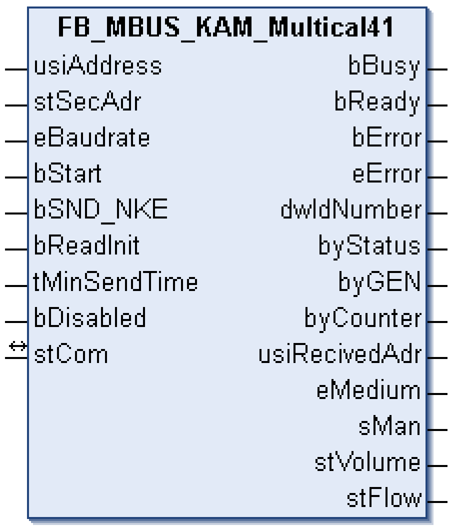
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21.5 FB_MBUS_KAM_Multical41



This block is used to read water meters from Kamstrup:

-Multical 41

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy          : BOOL;
bReady         : BOOL;
bError         : BOOL;
eError         : E_MBUS_ERROR;
dwIdNumber     : DWORD;
byStatus       : BYTE;
byGEN          : BYTE;
byCounter      : BYTE;
usiRecivedAdr  : USINT;
eMedium        : E_MBUS_Medium;
sMan           : STRING(3);
stVolume       : ST_MBus_Info;
stFlow         : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

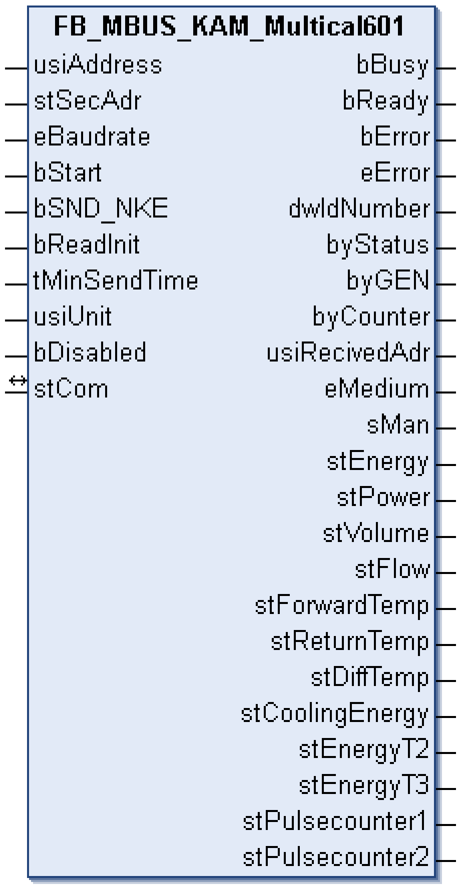
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.21.6 FB_MBUS_KAM_Multical601



This block is used for reading heat/cold meters from Kamstrup:

-Multical 601

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
```

```
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
stCoolingEnergy  : ST_MBus_Info;
stEnergyT2       : ST_MBus_Info;
stEnergyT3       : ST_MBus_Info;
stPulsecounter1  : ST_MBus_Info;
stPulsecounter2  : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stCoolingEnergy: Meter reading, cooling energy consumption (see ST_MBus_Info [▸ 205]).

stEnergyT2: Meter reading, energy consumption, tariff 2 (see ST_MBus_Info [▸ 205]).

stEnergyT3: Meter reading, energy consumption, tariff 3 (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

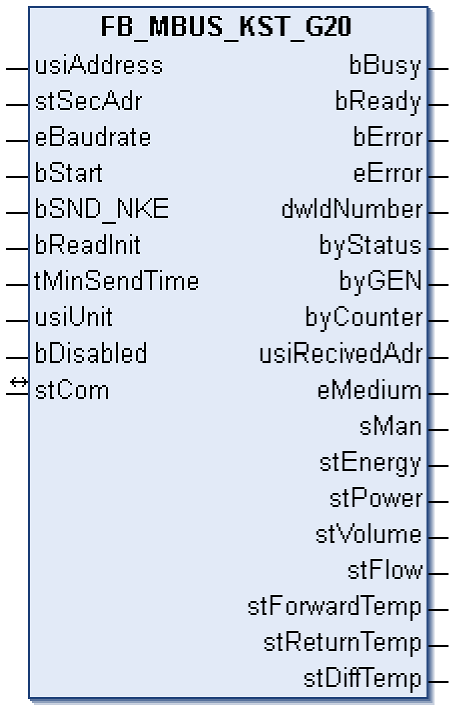
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.22 Kundo overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
KUNDO	Heat/cold meter	Compact WMZ G20	FB_MBUS_KST_G20 [▸ 134]
	Heat/cold meter	Compact WMZ G21	FB_MBUS_KST_G20 [▸ 134]
	External M-Bus module	him1s	FB_MBUS_KST_him1 [▸ 136]
	External M-Bus module	him1plus	FB_MBUS_KST_him1 [▸ 136]
	Pulse input	him1plus	FB_MBUS_KST_him1Puls [▸ 138]

4.1.22.1 FB_MBUS_KST_G20



This block is used for reading heat/cold meters from KUNDO System Technik:

-Kompakt WZM G20 (with internal M-Bus module)

-Kompakt WZM G21 (with internal M-Bus module)

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stEnergy        : ST_MBus_Info;
stPower         : ST_MBus_Info;
stVolume        : ST_MBus_Info;
stFlow          : ST_MBus_Info;
stForwardTemp   : ST_MBus_Info;
stReturnTemp    : ST_MBus_Info;
stDiffTemp      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

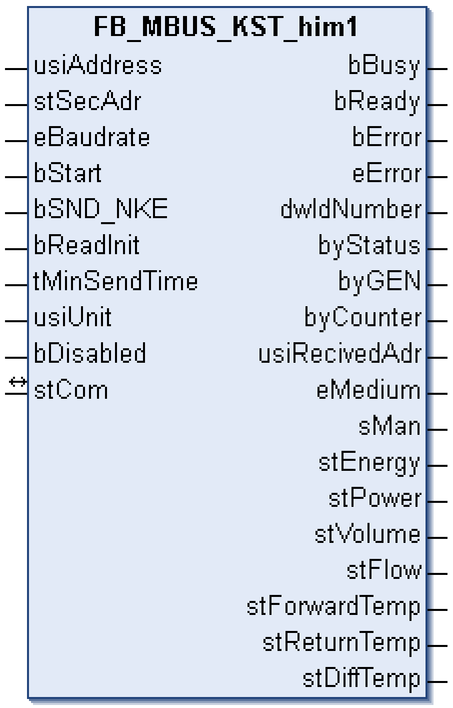
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.22.2 FB_MBUS_KST_him1



This block is used for reading M-Bus modules from KUNDO System Technik:

-him1s

-him1plus

These modules can be used for reading consumption data from a KUNDO arithmetic unit.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stPower          : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▶ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▶ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▶ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▶ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

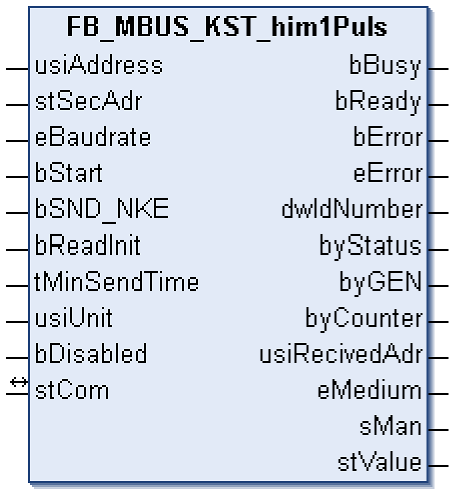
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.22.3 FB_MBUS_KST_him1Puls



This block is used for reading M-Bus modules from KUNDO System Technik:

-him1plus (pulse input)

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy           : BOOL;
bReady          : BOOL;
bError          : BOOL;
eError          : E_MBUS_ERROR;
dwIdNumber      : DWORD;
byStatus        : BYTE;
byGEN           : BYTE;
byCounter       : BYTE;
usiRecivedAdr   : USINT;
eMedium         : E_MBUS_Medium;
sMan            : STRING(3);
stValue         : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

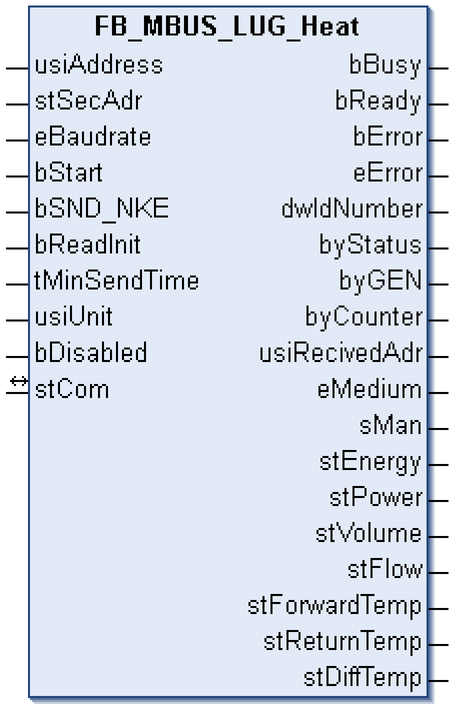
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.23 Landis & Gyr overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Landis & Gyr	Heat/cold meter	ULTRAHEAT 2WR5	FB_MBUS_LUG_Heat [▸ 140]
	Heat/cold meter	ULTRAHEAT 2WR6	FB_MBUS_LUG_Heat [▸ 140]
	Heat/cold meter	ULTRAHEAT UH50	FB_MBUS_LUG_Heat [▸ 140]

4.1.23.1 FB_MBUS_LUG_Heat



This block is used for reading heat/cold meters from Landis & Gyr:

-2WR5

-2WR6

-UH50

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 1200, 2400, 4800 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                 : BOOL;
bReady                : BOOL;
bError                : BOOL;
eError                : E_MBUS_ERROR;
dwIdNumber            : DWORD;
byStatus              : BYTE;
byGEN                 : BYTE;
byCounter             : BYTE;
usiRecivedAdr         : USINT;
eMedium               : E_MBUS_Medium;
sMan                  : STRING(3);
stEnergy              : ST_MBus_Info;
stPower               : ST_MBus_Info;
stVolume              : ST_MBus_Info;
stFlow                : ST_MBus_Info;
stForwardTemp         : ST_MBus_Info;
stReturnTemp          : ST_MBus_Info;
stDiffTemp            : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

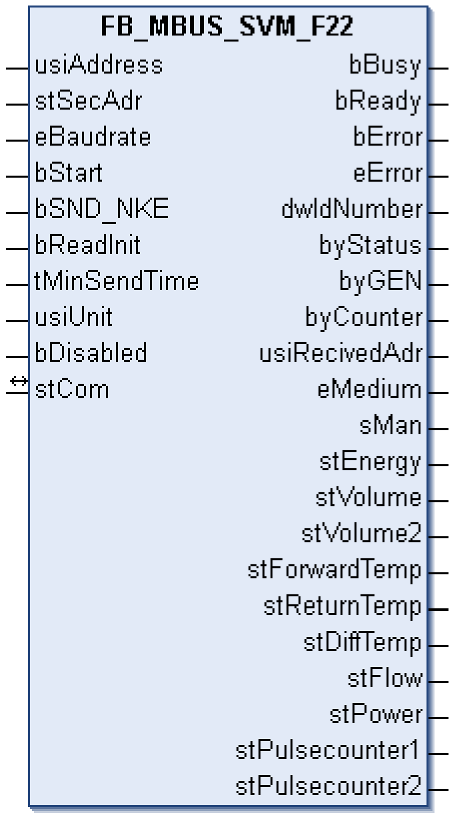
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.24 Metrima overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Metrima	Heat meter	F22 (standard values)	FB_MBUS_SVM_F22 [▶ 142]
	Heat meter	F22 (with additional output values)	FB_MBUS_SVM_F22_Ext [▶ 145]

4.1.24.1 FB_MBUS_SVM_F22



This block is used to read heat meters from Metrima:

-F22

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stVolume2        : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stPower          : ST_MBus_Info;
stPulsecounter1  : ST_MBus_Info;
stPulsecounter2  : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stVolume2: Accumulated volume. Energy calculation (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

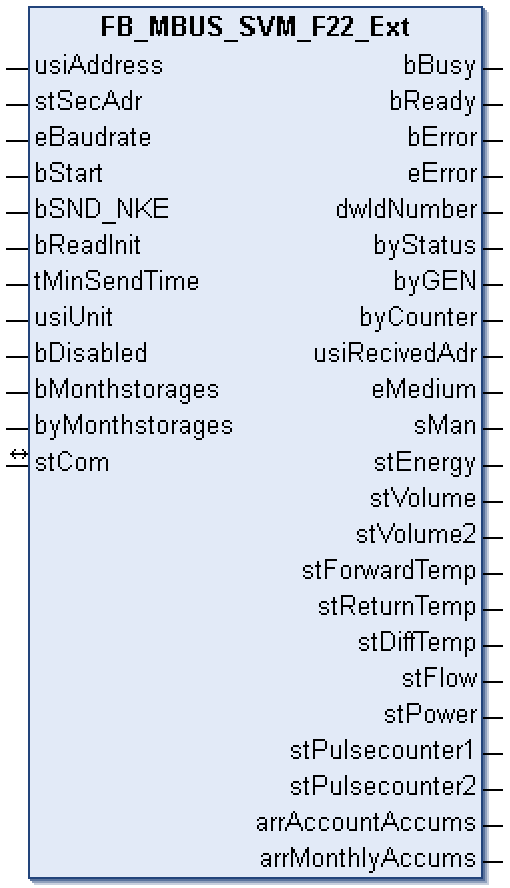
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.24.2 FB_MBUS_SVM_F22_Ext



This block is used to read heat meters from Metrima:

-F22 (as FB_MBUS_SVM_F22() [▶ 142], but with the extended output values *arrAcountAccums* and *arrMonthlyAccums*.)

This block is not suitable for BC/BX.

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress      : USINT;
stSecAdr        : ST_MBUS_SecAdr;
eBaudrate       : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart          : BOOL;
bSND_NKE        : BOOL := TRUE;
bReadInit       : BOOL := TRUE;
tMinSendTime    : TIME := t#2s;
usiUnit         : USINT;
bDisabled       : BOOL := FALSE;
bMonthstorages  : BOOL;
byMonthstorages : BYTE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

bMonthstorages: =False, cutoff dates and monthly values are not read (arrAccountAccums and arrMonthlyAccums) / =TRUE, cutoff dates and monthly values are read (arrAccountAccums and arrMonthlyAccums). The number of monthly values (arrMonthlyAccums) can be changed and depends on the variable byMonthstorages.

byMonthstorages: Number of monthly values (arrMonthlyAccums), maximum 37 values. Only applies if byMonthstorages =TRUE.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergy         : ST_MBus_Info;
stVolume         : ST_MBus_Info;
stVolume2        : ST_MBus_Info;
stForwardTemp    : ST_MBus_Info;
stReturnTemp     : ST_MBus_Info;
stDiffTemp       : ST_MBus_Info;
stFlow           : ST_MBus_Info;
stPower          : ST_MBus_Info;
stPulsecounter1  : ST_MBus_Info;
stPulsecounter2  : ST_MBus_Info;
arrAccountAccums : ARRAY [1..2] OF ST_MBus_F22;
arrMonthlyAccums : ARRAY [1..37] OF ST_MBus_F22;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stVolume2: Accumulated volume. Energy calculation (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Pulse counter 2 (see ST_MBus_Info [▸ 205]).

arrAccountAccums: 2 cutoff date values (energy, volume 1, volume 2, pulse counter 1, pulse counter 2, date). Values are only read if bMonthstorages =TRUE (see ST_MBus_F22 [▸ 207]).

arrMonthlyAccums: Maximum 37 monthly values (energy, volume 1, volume 2, pulse counter 1, pulse counter 2, date). Values are only read if bMonthstorages =TRUE. The number of values depends on the variable byMonthstorages (see ST_MBus_F22 [▸ 207]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

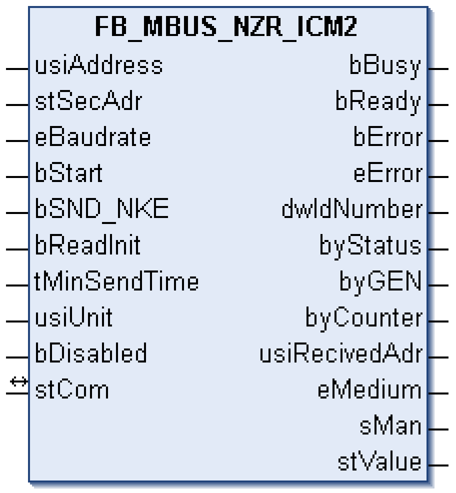
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.25 NZR overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
NZR	2-way pulse adapter	IC-M2	FB_MBUS_NZR_ICM2 [▸ 148]
	2-way pulse adapter	IC-M2C	FB_MBUS_NZR_ICM2 [▸ 148]
	Water meter	Modularis 2	FB_MBUS_NZR_Modularis 2 [▸ 150]

4.1.25.1 FB_MBUS_NZR_ICM2



This block is used to read energy meters with pulse output from NZR:

-IC-M2

-IC-M2C

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Up to 2 pulse generators can be connected to an IC-M2/IC-M2C at the same time. The IC-M2/IC-M2C behaves like 2 independent slaves.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

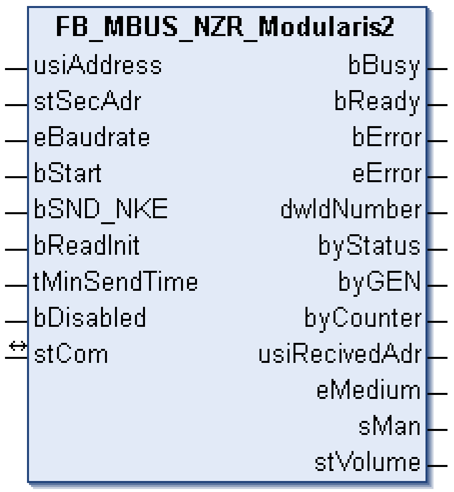
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.25.2 FB_MBUS_NZR_Modularis2



This block is used to read water meters from NZR:

-Modularis 2

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
```

```
eMedium              : E_MBUS_Medium;
sMan                 : STRING(3);
stVolume             : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

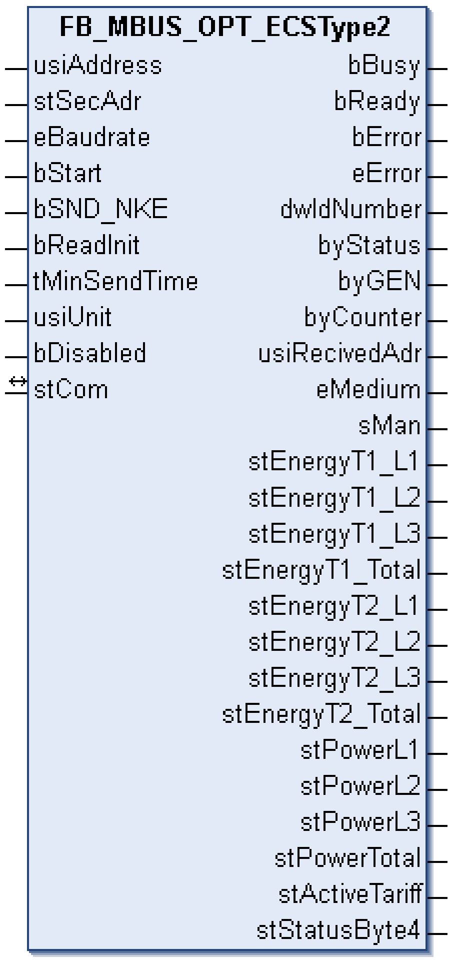
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.26 OPTEC overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
OPTEC	Electricity meter	ECS Type 2	FB_MBUS_OPT_ECSType2 [▶ 152]

4.1.26.1 FB_MBUS_OPT_ECSType2



This block is used to read electricity meters from OPTEC:

-ECS (default readout data type 2)

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergyT1_L1     : ST_MBus_Info;
stEnergyT1_L2     : ST_MBus_Info;
stEnergyT1_L3     : ST_MBus_Info;
stEnergyT1_Total  : ST_MBus_Info;
stEnergyT2_L1     : ST_MBus_Info;
stEnergyT2_L2     : ST_MBus_Info;
stEnergyT2_L3     : ST_MBus_Info;
stEnergyT2_Total  : ST_MBus_Info;
stPowerL1         : ST_MBus_Info;
stPowerL2         : ST_MBus_Info;
stPowerL3         : ST_MBus_Info;
stPowerTotal      : ST_MBus_Info;
stActiveTariff    : ST_MBus_Info;
stStatusByte4     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergyT1_L1: Counter value, active energy consumption, tariff 1, phase L1 (see ST_MBus_Info [▶ 205]).

stEnergyT1_L2: Meter reading, active energy consumption, tariff 1, phase L2 (see ST_MBus_Info [▶ 205]).

stEnergyT1_L3: Meter reading, active energy consumption, tariff 1, phase L3 (see ST_MBus_Info [▶ 205]).

stEnergyT1_Total: Meter reading, active energy consumption, tariff 1, total (see ST_MBus_Info [▶ 205]).

stEnergyT2_L1: Meter reading, active energy consumption, tariff 2, phase L1 (see ST_MBus_Info [▶ 205]).

stEnergyT2_L2: Meter reading, active energy consumption, tariff 2, phase L2 (see ST_MBus_Info [▶ 205]).

stEnergyT2_L3: Meter reading, active energy consumption, tariff 2, phase L3 (see ST_MBus_Info [▶ 205]).

stEnergyT2_Total: Meter reading, active energy consumption, tariff 2, total (see ST_MBus_Info [▸ 205]).

stPowerL1: Instantaneous consumption, power, phase L1 (see ST_MBus_Info [▸ 205]).

stPowerL2: Instantaneous consumption, power, phase L2 (see ST_MBus_Info [▸ 205]).

stPowerL3: Instantaneous consumption, power, phase L3 (see ST_MBus_Info [▸ 205]).

stPowerTotal: Instantaneous consumption, power, total (see ST_MBus_Info [▸ 205]).

stActiveTariff: Current tariff (see ST_MBus_Info [▸ 205]).

stStatusByte4: Range overflow alarms (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

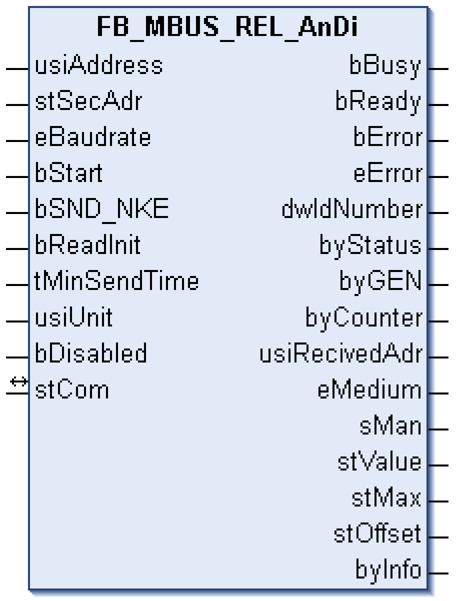
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.27 Relay overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Relay	1-4 analog inputs	AnDi 1-4	FB_MBUS_REL_AnDi [▸ 155]
	4 digital inputs	PadIn 4	FB_MBUS_REL_PadIn4 [▸ 157]
	1-way pulse adapter	PadPuls M1	FB_MBUS_REL_PadPulsM1 [▸ 159]
	1-way pulse adapter	PadPuls M1C	FB_MBUS_REL_PadPulsM1 [▸ 159]
	2-way pulse adapter	PadPuls M2	FB_MBUS_REL_PadPulsM2 [▸ 161]
	2-way pulse adapter	PadPuls M2C	FB_MBUS_REL_PadPulsM2 [▸ 161]
	4-way pulse adapter	PadPuls M4	FB_MBUS_REL_PadPulsM4 [▸ 163]
	4-way pulse adapter	PadPuls M4L	FB_MBUS_REL_PadPulsM4 [▸ 163]

4.1.27.1 FB_MBUS_REL_AnDi



This block is used for reading analog converters from Relay:

-AnDi 1 (1x 0/4-20 mA or 0-10 V)

-AnDi 2 (2x 0/4-20 mA or 0-10 V)

-AnDi 3 (3x 0/4-20 mA or 0-10 V)

-AnDi 4 (4x 0/4-20 mA or 0-10 V)

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Up to 4 sensors can be connected to an AnDi 4 at the same time. The AnDi 4 behaves like 4 independent slaves.

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue           : ST_MBus_Info;
stMax             : ST_MBus_Info;
stOffset          : ST_MBus_Info;
byInfo            : BYTE;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stValue: Counter value.

stMax: Maximum value.

stOffset: Offset.

byInfo: Information byte.

nBit7-4: Information about the A/D modules installed in AnDi4
nBit3: Protection bit (1: protection enabled)
nBit2-1: no. of the current measuring input (0: Port1 ... 3: Port4)
nBit0: I/V measurement (1: current measurement)

VAR_IN_OUT

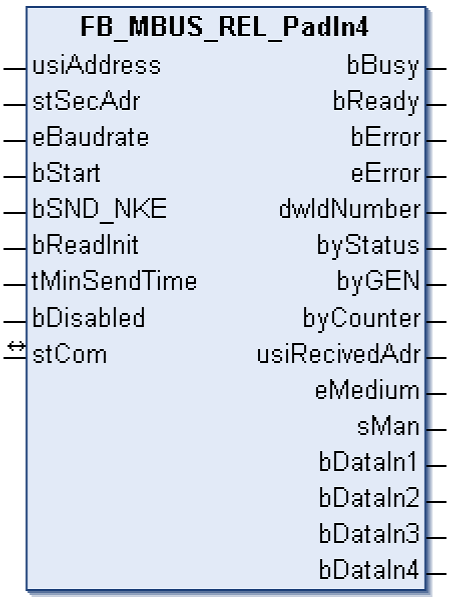
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.27.2 FB_MBUS_REL_PadIn4



This block is used for reading digital inputs from Relay:

-PadIn 4 (4 digital inputs)

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
```

```
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
bDataIn1         : BOOL;
bDataIn2         : BOOL;
bDataIn3         : BOOL;
bDataIn4         : BOOL;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

bDataIn1: Digital input 1

bDataIn2: Digital input 2

bDataIn3: Digital input 3

bDataIn4: Digital input 4

VAR_IN_OUT

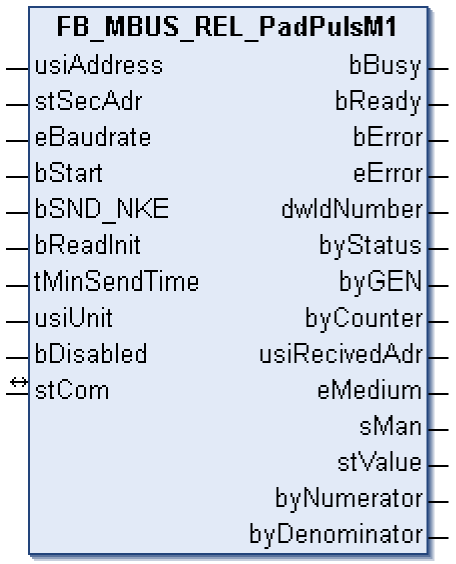
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.27.3 FB_MBUS_REL_PadPulsM1



This block is used for reading energy meters with pulse output from Relay:

-PadPuls M1

-PadPuls M1C

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
```

```
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stValue           : ST_MBus_Info;
byNumerator       : BYTE;
byDenominator     : BYTE;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▶ 205]).

byNumerator: Pulse value numerator (range 1..255).

byDenominator: Pulse value denominator (range 1..255).

VAR_IN_OUT

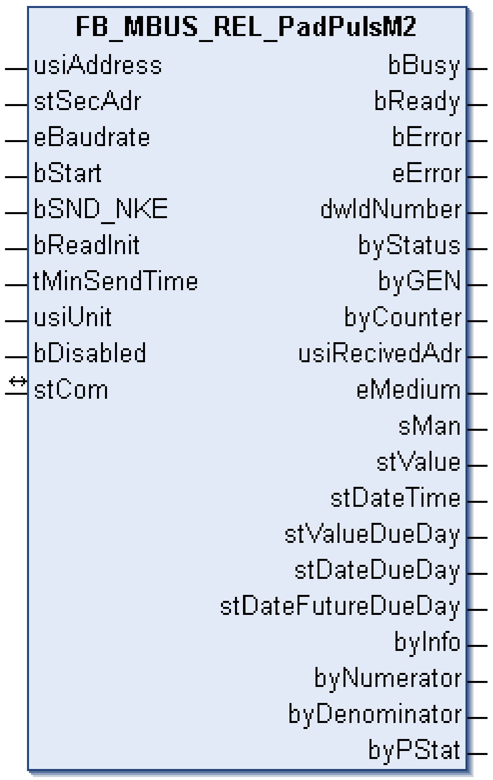
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.27.4 FB_MBUS_REL_PadPulsM2



This block is used for reading energy meters with pulse output from Relay:

-PadPuls M2

-PadPuls M2C

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Up to 2 pulse generators can be connected to a PadPuls 2/PadPuls 2C at the same time. The PadPuls 2/ PadPuls 2C behaves like 2 independent slaves.

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                : BOOL;
bReady               : BOOL;
bError               : BOOL;
eError               : E_MBUS_ERROR;
dwIdNumber           : DWORD;
byStatus             : BYTE;
byGEN                : BYTE;
byCounter            : BYTE;
usiRecivedAdr        : USINT;
eMedium              : E_MBUS_Medium;
sMan                 : STRING(3);
stValue              : ST_MBus_Info;
stDateTime           : ST_MBus_Info;
stValueDueDay        : ST_MBus_Info;
stDateDueDay         : ST_MBus_Info;
stDateFutureDueDay   : ST_MBus_Info;
byInfo               : BYTE;
byNumerator          : BYTE;
byDenominator        : BYTE;
byPStat              : BYTE;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▶ 205]).

stDateTime: Current date (see ST_MBus_Info [▶ 205]).

stValueDueDay: Cutoff date meter reading (see ST_MBus_Info [▶ 205]).

stDateDueDay: Last cutoff date (see ST_MBus_Info [▶ 205]).

stDateFutureDueDay: Future cutoff date (see ST_MBus_Info [▶ 205]).

byInfo: Information byte (tariff and sampling method).

byNumerator: Pulse value numerator (1..99).

byDenominator: Pulse value denominator (1..255, 0 -> 256).

byPStat: Port status (current contact state at the port inputs).

VAR_IN_OUT

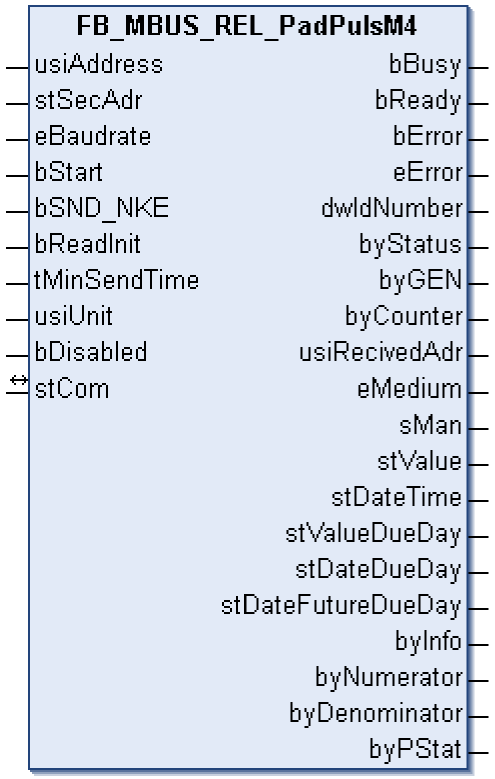
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.27.5 FB_MBUS_REL_PadPulsM4



This block is used for reading energy meters with pulse output from Relay:

-PadPuls M4

-PadPuls M4L

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Up to 4 pulse generators can be connected to a PadPuls 4/PadPuls 4L at the same time. The PadPuls 4/ PadPuls 4L behaves like 4 independent slaves.

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy               : BOOL;
bReady              : BOOL;
bError              : BOOL;
eError              : E_MBUS_ERROR;
dwIdNumber          : DWORD;
byStatus            : BYTE;
byGEN               : BYTE;
byCounter           : BYTE;
usiRecivedAdr       : USINT;
eMedium             : E_MBUS_Medium;
sMan                : STRING(3);
stValue             : ST_MBus_Info;
stDateTime          : ST_MBus_Info;
stValueDueDay       : ST_MBus_Info;
stDateDueDay        : ST_MBus_Info;
stDateFutureDueDay  : ST_MBus_Info;
byInfo              : BYTE;
byNumerator         : BYTE;
byDenominator       : BYTE;
byPStat             : BYTE;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stValue: Meter reading (see ST_MBus_Info [▸ 205]).

stDateTime: Current date (see ST_MBus_Info [▸ 205]).

stValueDueDay: Cutoff date meter reading (see ST_MBus_Info [▸ 205]).

stDateDueDay: Last cutoff date (see ST_MBus_Info [▸ 205]).

stDateFutureDueDay: Future cutoff date (see ST_MBus_Info [▸ 205]).

byInfo: Information byte (tariff and sampling method).

byNumerator: Pulse value numerator (1..99).

byDenominator: Pulse value denominator (1..255, 0 -> 256).

byPStat: Port status (current contact state at the port inputs).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

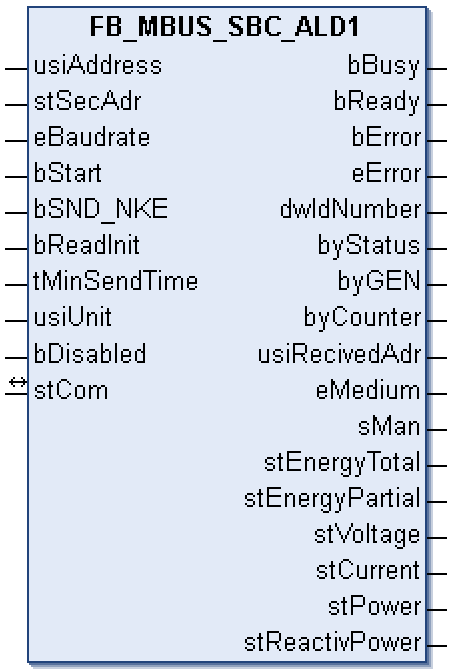
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.28 Saia-Burgess overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Saia-Burgess	Electricity meter	ALD1	FB_MBUS_SBC_ALD1 [▸ 166]
	Electricity meter	ALE3	FB_MBUS_SBC_ALE3 [▸ 168]
	Electricity meter	AWD3	FB_MBUS_SBC_ALE3 [▸ 168]

4.1.28.1 FB_MBUS_SBC_ALD1



This block is used to read electricity meters from Saia-Burgess:

-ALD1

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stEnergyTotal    : ST_MBus_Info;
stEnergyPartial  : ST_MBus_Info;
stVoltage        : ST_MBus_Info;
stCurrent        : ST_MBus_Info;
stPower          : ST_MBus_Info;
stReactivPower   : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergyTotal: Meter reading, energy total (see ST_MBus_Info [▸ 205]).

stEnergyPartial: Meter reading, partial energy consumption. This value is resettable (see ST_MBus_Info [▸ 205]).

stVoltage: Voltage (see ST_MBus_Info [▸ 205]).

stCurrent: Current (see ST_MBus_Info [▸ 205]).

stPower: Power (see ST_MBus_Info [▸ 205]).

stReactivPower: Reactive power (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

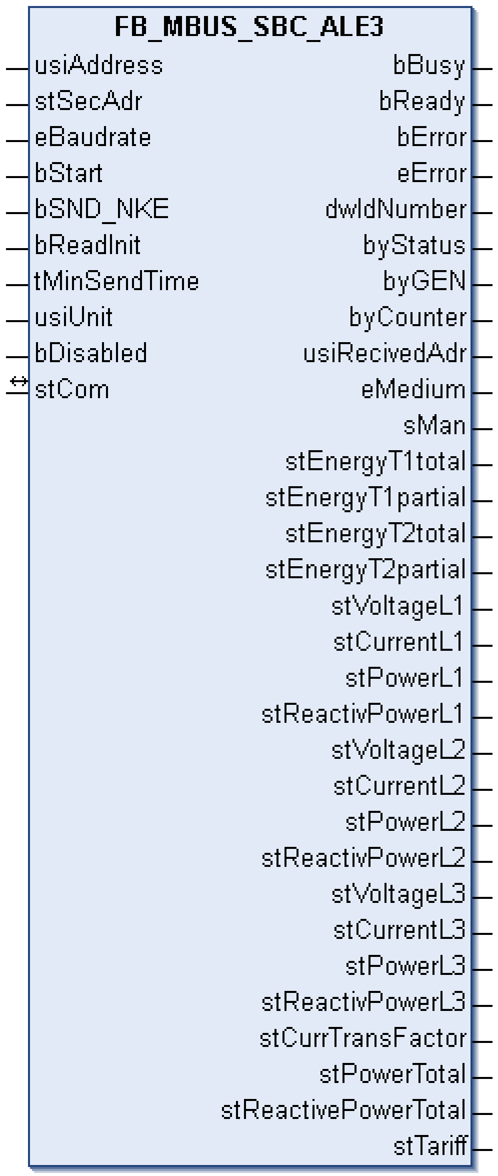
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.28.2 FB_MBUS_SBC_ALE3



This block is used to read electricity meters from Saia-Burgess:

-ALE3

-AWD3

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                    : BOOL;
bReady                   : BOOL;
bError                   : BOOL;
eError                   : E_MBUS_ERROR;
dwIdNumber               : DWORD;
byStatus                 : BYTE;
byGEN                    : BYTE;
byCounter                : BYTE;
usiRecivedAdr            : USINT;
eMedium                  : E_MBUS_Medium;
sMan                     : STRING(3);
stEnergyT1total          : ST_MBus_Info;
stEnergyT1partial        : ST_MBus_Info;
stEnergyT2total          : ST_MBus_Info;
stEnergyT2partial        : ST_MBus_Info;
stVoltageL1              : ST_MBus_Info;
stCurrentL1              : ST_MBus_Info;
stPowerL1                : ST_MBus_Info;
stReactivPowerL1         : ST_MBus_Info;
stVoltageL2              : ST_MBus_Info;
stCurrentL2              : ST_MBus_Info;
stPowerL2                : ST_MBus_Info;
stReactivPowerL2         : ST_MBus_Info;
stVoltageL3              : ST_MBus_Info;
stCurrentL3              : ST_MBus_Info;
stPowerL3                : ST_MBus_Info;
stReactivPowerL3         : ST_MBus_Info;
stCurrTransFactor        : ST_MBus_Info;
stPowerTotal             : ST_MBus_Info;
stReactivePowerTotal : ST_MBus_Info;
stTariff                 : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stEnergyT1total: Meter reading, energy total tariff 1 (see ST_MBus_Info [▸ 205]).

stEnergyT1partial: Meter reading, partial energy consumption tariff 1. This value is resettable (see ST_MBus_Info [▸ 205]).

stEnergyT2total: Meter reading, energy total tariff 2 (see ST_MBus_Info [▸ 205]).

stEnergyT2partial: Meter reading, partial energy consumption tariff 2. This value is resettable (see ST_MBus_Info [▸ 205]).

stVoltageL1: Voltage phase L1 (see ST_MBus_Info [▸ 205]).

stCurrentL1: Current phase L1 (see ST_MBus_Info [▸ 205]).

stPowerL1: Power phase L1 (see ST_MBus_Info [▸ 205]).

stReactivPowerL1: Reactive power phase L1 (see ST_MBus_Info [▸ 205]).

stVoltageL2: Voltage phase L2 (see ST_MBus_Info [▸ 205]).

stCurrentL2: Current phase L2 (see ST_MBus_Info [▸ 205]).

stPowerL2: Power phase L2 (see ST_MBus_Info [▸ 205]).

stReactivPowerL2: Reactive power phase L2 (see ST_MBus_Info [▸ 205]).

stVoltageL3: Voltage phase L3 (see ST_MBus_Info [▸ 205]).

stCurrentL3: Current phase L3 (see ST_MBus_Info [▸ 205]).

stPowerL3: Power phase L3 (see ST_MBus_Info [▸ 205]).

stReactivPowerL3: Reactive power phase L3 (see ST_MBus_Info [▸ 205]).

stCurrTransFactor: Transformer ratio (=0 for ALE3 devices) (see ST_MBus_Info [▸ 205]).

stPowerTotal: Total power (see ST_MBus_Info [▸ 205]).

stReactivePowerTotal: Total reactive power (see ST_MBus_Info [▸ 205]).

stTariff: Current tariff (=0 for AWD3 devices) (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

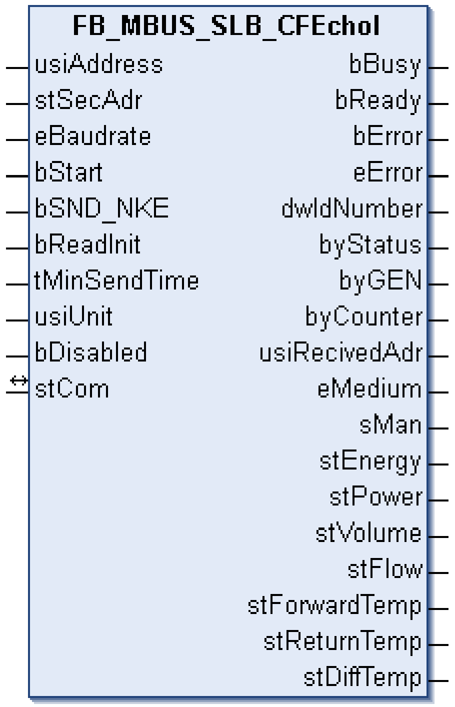
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.29 Schlumberger overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Schlumberger	Heat meter	Integral-Mk MaXX	FB_MBUS_SLB_MK_MaXX [▸ 173]
	Heat meter	CF Echo I	FB_MBUS_SLB_CFEchol [▸ 171]

4.1.29.1 FB_MBUS_SLB_CFEchol



This block is used to read heat meters from Schlumberger:

-CF Echo I

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy             : BOOL;
bReady            : BOOL;
bError            : BOOL;
eError            : E_MBUS_ERROR;
dwIdNumber        : DWORD;
byStatus          : BYTE;
byGEN             : BYTE;
byCounter         : BYTE;
usiRecivedAdr     : USINT;
eMedium           : E_MBUS_Medium;
sMan              : STRING(3);
stEnergy          : ST_MBus_Info;
stPower           : ST_MBus_Info;
stVolume          : ST_MBus_Info;
stFlow            : ST_MBus_Info;
stForwardTemp     : ST_MBus_Info;
stReturnTemp      : ST_MBus_Info;
stDiffTemp        : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

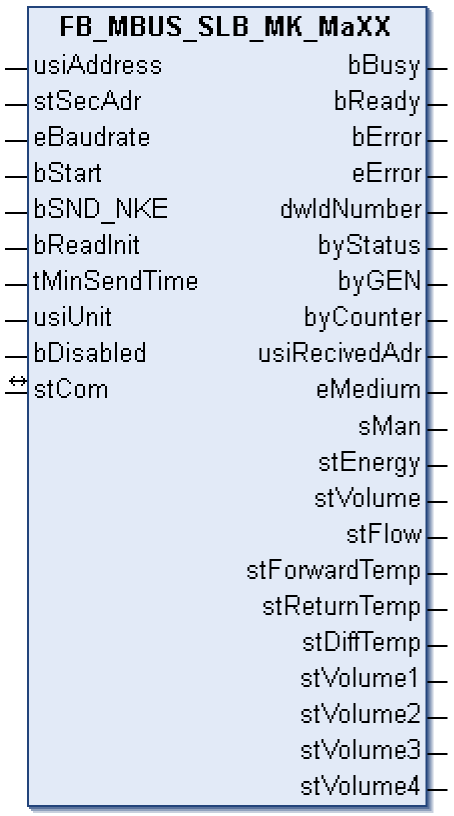
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.29.2 FB_MBUS_SLB_MK_MaXX



This block is used to read heat meters from Schlumberger:

-Integral-MK Maxx / Up to 4 additional water meters can be connected to this device.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stVolume           : ST_MBus_Info;
stFlow             : ST_MBus_Info;
stForwardTemp      : ST_MBus_Info;
stReturnTemp       : ST_MBus_Info;
stDiffTemp         : ST_MBus_Info;
stVolume1          : ST_MBus_Info;
stVolume2          : ST_MBus_Info;
stVolume3          : ST_MBus_Info;
stVolume4          : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

stVolume1: Meter reading of additional water meter 1 (see ST_MBus_Info [▸ 205]).

stVolume2: Meter reading of additional water meter 2 (see ST_MBus_Info [▸ 205]).

stVolume3: Meter reading of additional water meter 3 (see ST_MBus_Info [▸ 205]).

stVolume4: Meter reading of additional water meter 4 (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

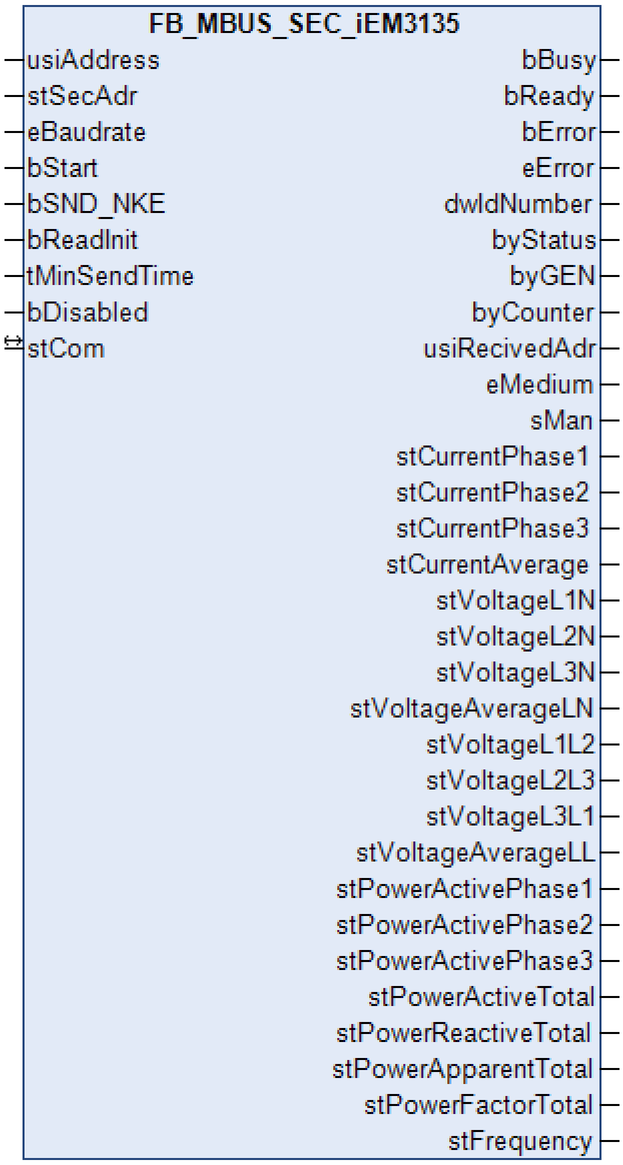
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.30 Schneider Electric

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Function block
Schneider Electric	Electricity meter	iEM3135	FB_MBUS_SEC_iEM3135 [▸ 176]

4.1.30.1 FB_MBUS_SEC_iEM3135



This function block is used to read electricity meters from Schneider Electric.

The function block can only be executed together with the function block FB_MBUSKL6781() [▸ 19].

Functionality of the function block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300..9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the function block.

VAR_OUTPUT

```
bBusy                  : BOOL;
bReady                 : BOOL;
bError                 : BOOL;
eError                 : E_MBUS_ERROR;
dwIdNumber             : DWORD;
byStatus               : BYTE;
byGEN                  : BYTE;
byCounter              : BYTE;
usiRecivedAdr          : USINT;
eMedium                : E_MBUS_Medium;
sMan                   : STRING(3);
stCurrentPhase1        : ST_MBus_Info;
stCurrentPhase2        : ST_MBus_Info;
stCurrentPhase3        : ST_MBus_Info;
stCurrentAverage       : ST_MBus_Info;
stVoltageL1N           : ST_MBus_Info;
stVoltageL2N           : ST_MBus_Info;
stVoltageL3N           : ST_MBus_Info;
stVoltageAverageLN     : ST_MBus_Info;
stVoltageL1L2          : ST_MBus_Info;
stVoltageL2L3          : ST_MBus_Info;
stVoltageL3L1          : ST_MBus_Info;
stVoltageAverageLL     : ST_MBus_Info;
stPowerActivePhase1    : ST_MBus_Info;
stPowerActivePhase2    : ST_MBus_Info;
stPowerActivePhase3    : ST_MBus_Info;
stPowerActiveTotal     : ST_MBus_Info;
stPowerReactiveTotal   : ST_MBus_Info;
stPowerApparentTotal   : ST_MBus_Info;
stPowerFactorTotal     : ST_MBus_Info;
stFrequency            : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stCurrentPhase1: Current intensity phase 1 (see ST_MBus_Info [▶ 205]).

stCurrentPhase2: Current intensity phase 2 (see ST_MBus_Info [▶ 205]).

stCurrentPhase3: Current intensity phase 3 (see ST_MBus_Info [▶ 205]).

stCurrentAverage: Average value of current intensity (see ST_MBus_Info [▶ 205]).

stVoltageL1N: Voltage L1-N (see ST_MBus_Info [▶ 205]).

stVoltageL2N: Voltage L2-N (see ST_MBus_Info [▶ 205]).

stVoltageL3N: Voltage L3-N (see ST_MBus_Info [▶ 205]).

stVoltageAverageLN: Average value of voltage L-N (see ST_MBus_Info [▶ 205]).

stVoltageL1L2: Voltage L1-L2 (see ST_MBus_Info [▶ 205]).

stVoltageL2L3: Voltage L2-L3 (see ST_MBus_Info [▶ 205]).

stVoltageL3L1: Voltage L3-L1 (see ST_MBus_Info [▶ 205]).

stVoltageAverageLL: Average value of voltage L-L (see ST_MBus_Info [▶ 205]).

stPowerActivePhase1: Effective power phase 1 (see ST_MBus_Info [▶ 205]).

stPowerActivePhase2: Effective power phase 2 (see ST_MBus_Info [▶ 205]).

stPowerActivePhase3: Effective power phase 3 (see ST_MBus_Info [▶ 205]).

stPowerActiveTotal: Total effective power (see ST_MBus_Info [▶ 205]).

stPowerReactiveTotal: Total reactive power (see ST_MBus_Info [▶ 205]).

stPowerApparentTotal: Total apparent power (see ST_MBus_Info [▶ 205]).

stPowerFactorTotal: Power factor (see ST_MBus_Info [▶ 205]).

stFrequency: Frequency (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

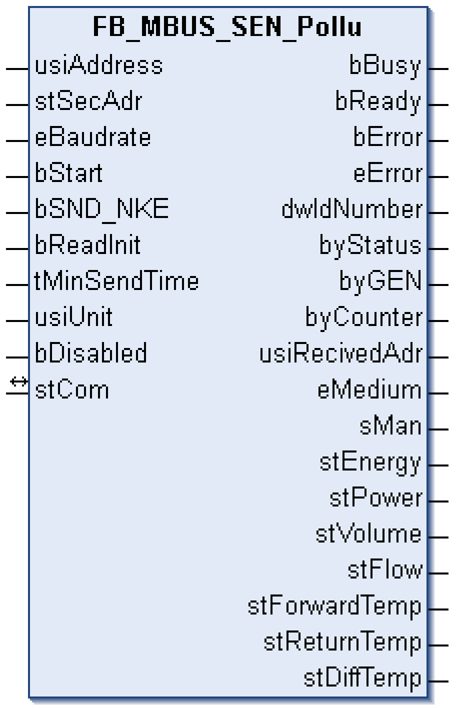
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.32	Tc2_MBus from 3.4.6.0

4.1.31 Sensus overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▶ 77], FB_MBUS_General_Ext [▶ 80] or FB_MBUS_General_Param [▶ 84] from the folder "General [▶ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▶ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Sensus	Heat/cold meter	PolluStat E	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluTherm	FB_MBUS_SEN_Pollu [▶ 179]
	Heat/cold meter	PolluCom E	FB_MBUS_SEN_Pollu [▶ 179]
	Water meter		FB_MBUS_SEN_Water [▶ 181]

4.1.31.1 FB_MBUS_SEN_Pollu



This block is used for reading heat/cold meters from Sensus:

-PolluStat E

-PolluCom E

-PolluTherm

The block can only be executed together with the block FB_MBUSKL6781() [▶ 19].

Functionality of the block [▶ 10]

VAR_INPUT

```
usiAddress    : USINT;
stSecAdr      : ST_MBUS_SecAdr;
eBaudrate     : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit       : USINT;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy               : BOOL;
bReady              : BOOL;
bError              : BOOL;
eError              : E_MBUS_ERROR;
dwIdNumber          : DWORD;
byStatus            : BYTE;
byGEN               : BYTE;
byCounter           : BYTE;
usiRecivedAdr       : USINT;
eMedium             : E_MBUS_Medium;
sMan                : STRING(3);
stEnergy            : ST_MBus_Info;
stPower             : ST_MBus_Info;
stVolume            : ST_MBus_Info;
stFlow              : ST_MBus_Info;
stForwardTemp       : ST_MBus_Info;
stReturnTemp        : ST_MBus_Info;
stDiffTemp          : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDiffTemp: Temperature difference (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

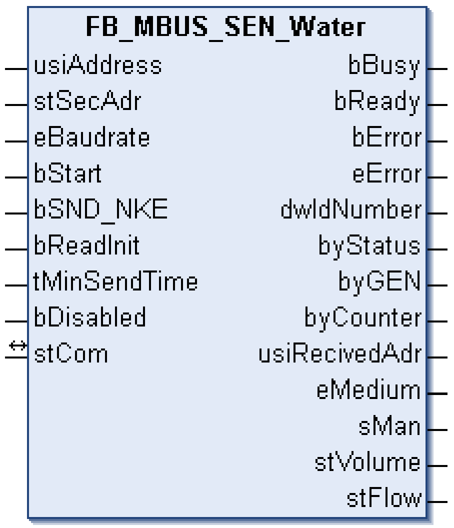
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.31.2 FB_MBUS_SEN_Water



This block is used to read water meters from Sensus.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy            : BOOL;
bReady           : BOOL;
bError           : BOOL;
eError           : E_MBUS_ERROR;
dwIdNumber       : DWORD;
byStatus         : BYTE;
byGEN            : BYTE;
byCounter        : BYTE;
usiRecivedAdr    : USINT;
eMedium          : E_MBUS_Medium;
sMan             : STRING(3);
stVolume         : ST_MBus_Info;
stFlow           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stVolume: Meter reading, water consumption (see ST_MBus_Info [▶ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▶ 203] with the meter blocks (see ST_MBUS_Communication [▶ 203]).

Requirements

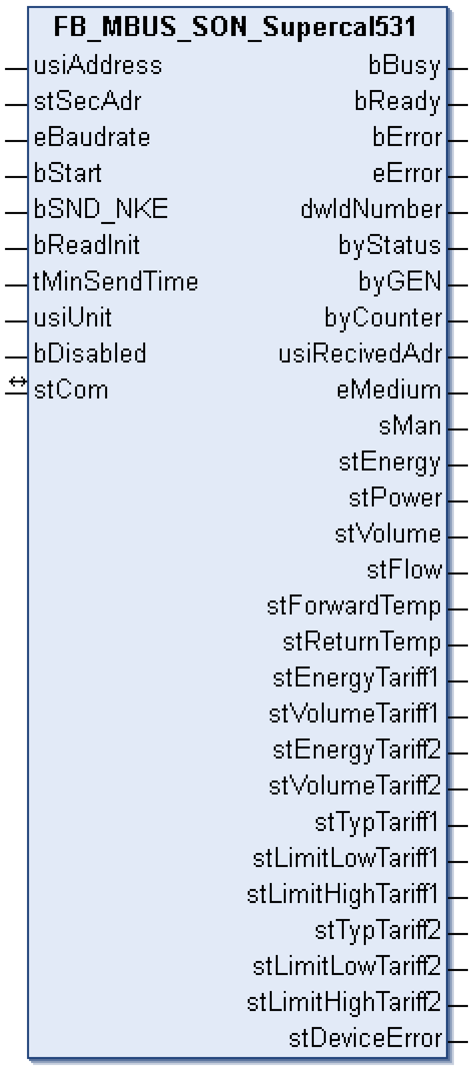
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.32 Sontex overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Sontex	Heat/cold meter	Supercal 531 (standard values)	FB_MBUS_SON_Supercal531 [▸ 183]

4.1.32.1 FB_MBUS_SON_Supercal531



This block is used for reading heat/cold meters from Sontex:

-Supercal 531

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
usiUnit      : USINT;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

usiUnit: Unit of the energy values to be output by the block. 0=W(h) / 1=KW(h) / 2 =MW(h) / 3=GW(h).

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stPower            : ST_MBus_Info;
stVolume           : ST_MBus_Info;
stFlow             : ST_MBus_Info;
stForwardTemp      : ST_MBus_Info;
stReturnTemp       : ST_MBus_Info;
stEnergyTariff1    : ST_MBus_Info;
stVolumeTariff1    : ST_MBus_Info;
stEnergyTariff2    : ST_MBus_Info;
stVolumeTariff2    : ST_MBus_Info;
stTypTariff1       : ST_MBus_Info;
stLimitLowTariff1  : ST_MBus_Info;
stLimitHighTariff1 : ST_MBus_Info;
stTypTariff2       : ST_MBus_Info;
stLimitLowTariff2  : ST_MBus_Info;
stLimitHighTariff2 : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Meter reading, energy consumption (see ST_MBus_Info [▸ 205]).

stPower: Current energy consumption, power (see ST_MBus_Info [▸ 205]).

stVolume: Meter reading, water consumption (see ST_MBus_Info [▸ 205]).

stFlow: Current water consumption (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stEnergyTariff1: Meter reading, energy consumption, tariff 1 (see ST_MBus_Info [▸ 205]).

stVolumeTariff1: Meter reading, water consumption, tariff 1 (see ST_MBus_Info [▸ 205]).

stEnergyTariff2: Meter reading, energy consumption, tariff 2 (see ST_MBus_Info [▸ 205]).

stVolumeTariff2: Meter reading, water consumption, tariff 2 (see ST_MBus_Info [▸ 205]).

stTypTariff1: Type tariff 1 (see ST_MBus_Info [▸ 205]).

stLimitLowTariff1: Lower limit value tariff 1 (see ST_MBus_Info [▸ 205]).

stLimitHighTariff1: Upper limit value tariff 1 (see ST_MBus_Info [▸ 205]).

stTypTariff2: Type tariff 2 (see ST_MBus_Info [▸ 205]).

stLimitLowTariff2: Lower limit value tariff 2 (see ST_MBus_Info [▸ 205]).

stLimitHighTariff2: Upper limit value tariff 2 (see ST_MBus_Info [▸ 205]).

stDeviceError: Error message from device (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

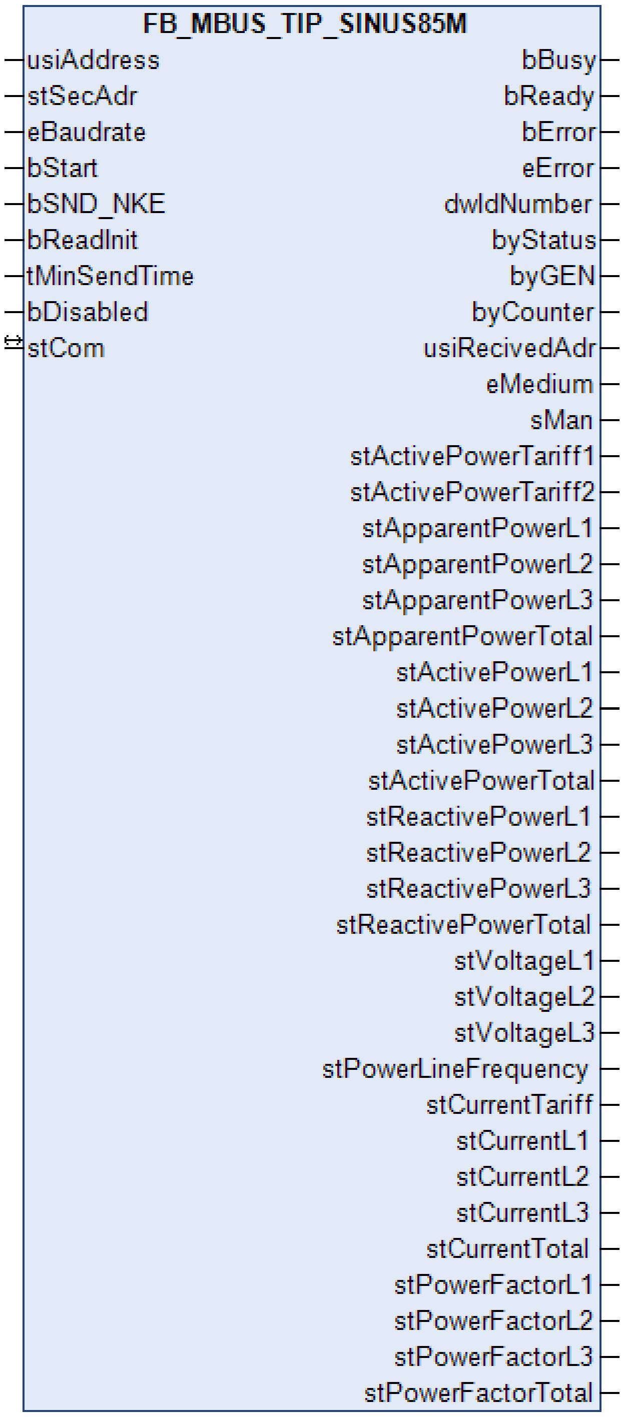
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.33 TIP

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Function block
TIP	Electricity meter	SINUS 85 M	FB_MBUS_TIP_SINUS85M [▶ 186]

4.1.33.1 FB_MBUS_TIP_SINUS85M



This function block is used to read electricity meters from Thüringer Industrie Produkte GmbH.

The function block can only be executed together with the function block FB_MBUSKL6781() [▶ 19].

Functionality of the function block [▶ 10]

VAR_INPUT

```
usiAddress     : USINT;
stSecAdr       : ST_MBUS_SecAdr;
eBaudrate      : E_MBUS_Baudrate := eMBUS_Baud2400;
```

```
bStart        : BOOL;
bSND_NKE      : BOOL := TRUE;
bReadInit     : BOOL := TRUE;
tMinSendTime  : TIME := t#2s;
bDisabled     : BOOL := FALSE;
```

usiAddress: Primary address [▶ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▶ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▶ 206]).

eBaudrate: 300, 600, 1200, 2400, 4800, 9600 baud (see E_MBUS_Baudrate [▶ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the function block.

VAR_OUTPUT

```
bByte
```

```
bBusy                  : BOOL;
bReady                 : BOOL;
bError                 : BOOL;
eError                 : E_MBUS_ERROR;
dwIdNumber             : DWORD;
byStatus               : BYTE;
byGEN                  : BYTE;
byCounter              : BYTE;
usiRecivedAdr          : USINT;
eMedium                : E_MBUS_Medium;
sMan                   : STRING(3);
stActivePowerTariff1 : ST_MBus_Info;
stActivePowerTariff2 : ST_MBus_Info;
stApparentPowerL1      : ST_MBus_Info;
stApparentPowerL2      : ST_MBus_Info;
stApparentPowerL3      : ST_MBus_Info;
stApparentPowerTotal : ST_MBus_Info;
stActivePowerL1        : ST_MBus_Info;
stActivePowerL2        : ST_MBus_Info;
stActivePowerL3        : ST_MBus_Info;
stActivePowerTotal     : ST_MBus_Info;
stReactivePowerL1      : ST_MBus_Info;
stReactivePowerL2      : ST_MBus_Info;
stReactivePowerL3      : ST_MBus_Info;
stReactivePowerTotal : ST_MBus_Info;
stVoltageL1            : ST_MBus_Info;
stVoltageL2            : ST_MBus_Info;
stVoltageL3            : ST_MBus_Info;
stPowerLineFrequency : ST_MBus_Info;
stCurrentTariff        : ST_MBus_Info;
stCurrentL1            : ST_MBus_Info;
stCurrentL2            : ST_MBus_Info;
stCurrentL3            : ST_MBus_Info;
stCurrentTotal         : ST_MBus_Info;
stPowerFactorL1        : ST_MBus_Info;
stPowerFactorL2        : ST_MBus_Info;
stPowerFactorL3        : ST_MBus_Info;
stPowerFactorTotal     : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▶ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

stActivePowerTariff1: Active energy import tariff 1 (see ST_MBus_Info [▶ 205]).

stActivePowerTariff2: Active energy import tariff 2 (see ST_MBus_Info [▶ 205]).

stApparentPowerL1: Current apparent power L1 (see ST_MBus_Info [▶ 205]).

stApparentPowerL2: Current apparent power L2 (see ST_MBus_Info [▶ 205]).

stApparentPowerL3: Current apparent power L3 (see ST_MBus_Info [▶ 205]).

stApparentPowerTotal: Current total apparent power (see ST_MBus_Info [▶ 205]).

stActivePowerL1: Current effective power phase L1 (see ST_MBus_Info [▶ 205]).

stActivePowerL2: Current effective power phase L2 (see ST_MBus_Info [▶ 205]).

stActivePowerL3: Current effective power phase L3 (see ST_MBus_Info [▶ 205]).

stActivePowerTotal: Current total effective power (see ST_MBus_Info [▶ 205]).

stReactivePowerL1: Current reactive power phase L1 (see ST_MBus_Info [▶ 205]).

stReactivePowerL2: Current reactive power phase L2 (see ST_MBus_Info [▶ 205]).

stReactivePowerL3: Current reactive power phase L3 (see ST_MBus_Info [▶ 205]).

stReactivePowerTotal: Current total reactive power (see ST_MBus_Info [▶ 205]).

stVoltageL1: Current voltage phase L1 (see ST_MBus_Info [▶ 205]).

stVoltageL2: Current voltage phase L2 (see ST_MBus_Info [▶ 205]).

stVoltageL3: Current voltage phase L3 (see ST_MBus_Info [▶ 205]).

stPowerLineFrequency: Current mains frequency (see ST_MBus_Info [▶ 205]).

stCurrentTariff: Current tariff (see ST_MBus_Info [▶ 205]).

stCurrentL1: Current phase L1 current (see ST_MBus_Info [▶ 205]).

stCurrentL2: Current phase L2 current (see ST_MBus_Info [▶ 205]).

stCurrentL3: Current phase L3 current (see ST_MBus_Info [▶ 205]).

stCurrentTotal: Current total current (see ST_MBus_Info [▶ 205]).

stPowerFactorL1: Current form factor phase L1 (cos Phi) (see ST_MBus_Info [▶ 205]).

stPowerFactorL2: Current form factor phase L2 (cos Phi) (see ST_MBus_Info [▶ 205]).

stPowerFactorL3: Current form factor phase L3 (cos Phi) (see ST_MBus_Info [▶ 205]).

stPowerFactorTotal: Current total form factor (cos Phi) (see ST_MBus_Info [▶ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

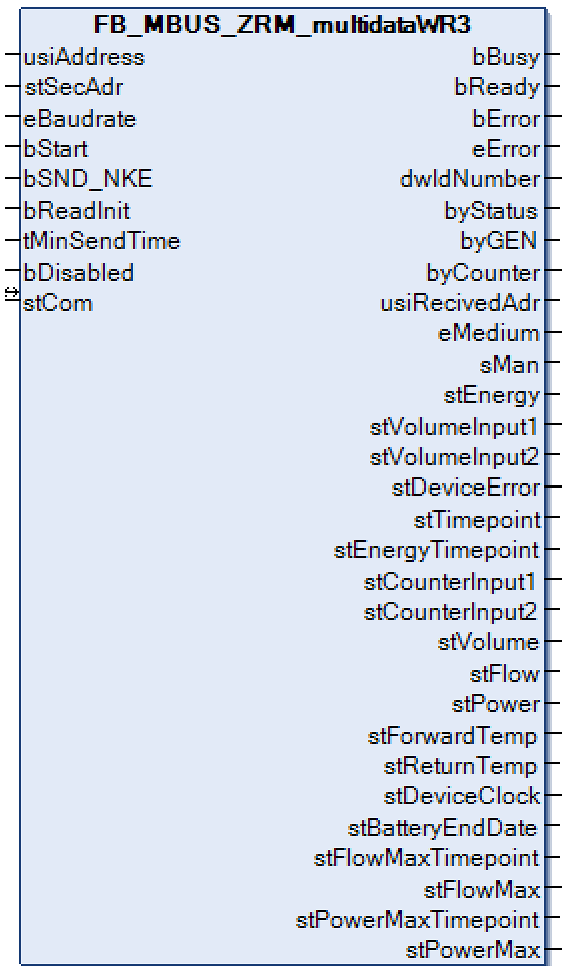
Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.32	Tc2_MBus from 3.4.6.0

4.1.34 Zenner overview

The blocks only output a selection of the most common data. These data are described on the respective pages under "VAR_OUTPUT". If more or all data are required, the blocks FB_MBUS_General [▸ 77], FB_MBUS_General_Ext [▸ 80] or FB_MBUS_General_Param [▸ 84] from the folder "General [▸ 75]" should be used. Note that these blocks do not run BC and BX systems. The block FB_MBUS_General_Send() [▸ 86] can be used to send data to the device (e.g. setting of the primary address).

Manufacturer	Type	Device	Block
Zenner	Arithmetic unit	multidataWR3	FB_MBUS_ZRM_multidataWR3 [▸ 190]
	Heat meter	zelsiusZR	FB_MBUS_ZRM_zelsiusZR [▸ 192]

4.1.34.1 FB_MBUS_ZRM_multidataWR3



This block is used to read arithmetic units from Zenner.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy                : BOOL;
bReady               : BOOL;
bError               : BOOL;
eError               : E_MBUS_ERROR;
dwIdNumber           : DWORD;
byStatus             : BYTE;
byGEN                : BYTE;
byCounter            : BYTE;
usiRecivedAdr        : USINT;
eMedium              : E_MBUS_Medium;
sMan                 : STRING(3);
stEnergy             : ST_MBus_Info;
stVolumeInput1       : ST_MBus_Info;
stVolumeInput2       : ST_MBus_Info;
stDeviceError        : ST_MBus_Info;
stTimepoint          : ST_MBus_Info;
stEnergyTimepoint    : ST_MBus_Info;
stCounterInput1      : ST_MBus_Info;
stCounterInput2      : ST_MBus_Info;
stVolume             : ST_MBus_Info;
stFlow               : ST_MBus_Info;
stPower              : ST_MBus_Info;
stForwardTemp        : ST_MBus_Info;
stReturnTemp         : ST_MBus_Info;
stDeviceClock        : ST_MBus_Info;
stBatteryEndDate     : ST_MBus_Info;
stFlowMaxTimepoint   : ST_MBus_Info;
stFlowMax            : ST_MBus_Info;
stPowerMaxTimepoint  : ST_MBus_Info;
stPowerMax           : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Current heat energy (see ST_MBus_Info [▸ 205]).

stVolumeInput1: Volume input 1 (see ST_MBus_Info [▸ 205]).

stVolumeInput2: Volume input 2 (see ST_MBus_Info [▸ 205]).

stDeviceError: Error status MBus output (see ST_MBus_Info [▸ 205]).

stTimepoint: Cutoff date (date and time of the next cutoff date) (see ST_MBus_Info [▸ 205]).

stEnergyTimepoint: Heat energy on the cutoff date (see ST_MBus_Info [▸ 205]).

stCounterInput1: Count value input 1 on the cutoff date (see ST_MBus_Info [▸ 205]).

stCounterInput2: Count value input 2 on the cutoff date (see ST_MBus_Info [▸ 205]).

stVolume: Volume (see ST_MBus_Info [▸ 205]).

stFlow: Instantaneous flow rate (see ST_MBus_Info [▸ 205]).

stPower: Power (see ST_MBus_Info [▸ 205]).

stForwardTemp: Flow temperature (see ST_MBus_Info [▸ 205]).

stReturnTemp: Return temperature (see ST_MBus_Info [▸ 205]).

stDeviceClock: Current meter time (see ST_MBus_Info [▸ 205]).

stBatteryEndDate: Expected battery shelf life (see ST_MBus_Info [▸ 205]).

stFlowMaxTimepoint: Timing of maximum flow rate reading (absolute) (see ST_MBus_Info [▸ 205]).

stFlowMax: Maximum flow rate (absolute) (see ST_MBus_Info [▸ 205]).

stPowerMaxTimepoint: Timing of maximum power reading (absolute) (see ST_MBus_Info [▸ 205]).

stPowerMax: Maximum power (absolute) (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

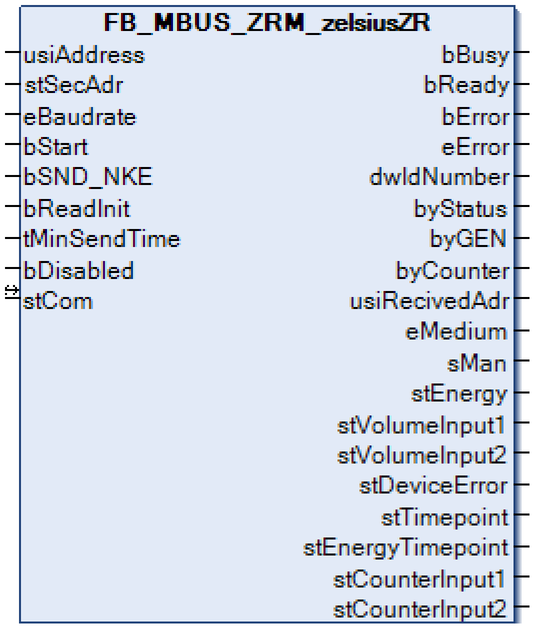
```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.34.2 FB_MBUS_ZRM_zelsiusZR



This block is used to read heat meters from Zenner.

The block can only be executed together with the block FB_MBUSKL6781() [▸ 19].

Functionality of the block [▸ 10]

VAR_INPUT

```
usiAddress   : USINT;
stSecAdr     : ST_MBUS_SecAdr;
eBaudrate    : E_MBUS_Baudrate := eMBUS_Baud2400;
bStart       : BOOL;
bSND_NKE     : BOOL := TRUE;
bReadInit    : BOOL := TRUE;
tMinSendTime : TIME := t#2s;
bDisabled    : BOOL := FALSE;
```

usiAddress: Primary address [▸ 11] of the meter to be read with this block.

stSecAdr: Secondary address [▸ 12] of the meter to be read with this block (see ST_MBUS_SecAdr [▸ 206]).

eBaudrate: 300, 2400, 9600 baud (see E_MBUS_Baudrate [▸ 198]).

bStart: A positive edge of this input triggers one reading of the meter.

bSND_NKE: TRUE initializes the meter for each read operation and sets the meter to the first telegram (SND_NKE).

bReadInit: If the PLC is restarted, the meter is read once.

tMinSendTime: Standard t#2s. The meter is read again, once the time set here has elapsed. If t#0s the meter is not read and can be read manually with *bStart*.

bDisabled: TRUE = deselection of the block.

VAR_OUTPUT

```
bBusy              : BOOL;
bReady             : BOOL;
bError             : BOOL;
eError             : E_MBUS_ERROR;
dwIdNumber         : DWORD;
byStatus           : BYTE;
byGEN              : BYTE;
byCounter          : BYTE;
usiRecivedAdr      : USINT;
eMedium            : E_MBUS_Medium;
sMan               : STRING(3);
stEnergy           : ST_MBus_Info;
stVolumeInput1     : ST_MBus_Info;
stVolumeInput2     : ST_MBus_Info;
stDeviceError      : ST_MBus_Info;
stTimepoint        : ST_MBus_Info;
stEnergyTimepoint  : ST_MBus_Info;
stCounterInput1    : ST_MBus_Info;
stCounterInput2    : ST_MBus_Info;
```

bBusy: The *bBusy* output is TRUE while the meter is being read.

bReady: The *bReady* output is TRUE for one cycle, once meter reading is completed.

bError: this output goes TRUE as soon as an error occurs. This error is described via the variable *eError*.

eError: In the event of an error the output issues an error code (see E_MBUS_ERROR [▸ 198]). *bError* goes TRUE at the same time.

dwIdNumber: Serial number of the meter (secondary address).

byStatus: Device status.

byGEN: Software version of the device.

byCounter: Number of times the master has accessed data of the respective slave.

usiRecivedAdr: Received primary address (0-250).

eMedium: Medium (see E_MBUS_Medium [▸ 201]).

sMan: Manufacturer code.

stEnergy: Current heat energy (see ST_MBus_Info [▸ 205]).

stVolumeInput1: Volume input 1 (see ST_MBus_Info [▸ 205]).

stVolumeInput2: Volume input 2 (see ST_MBus_Info [▸ 205]).

stDeviceError: Error status MBus output (see ST_MBus_Info [▸ 205]).

stTimepoint: Cutoff date (date and time of the next cutoff date) (see ST_MBus_Info [▸ 205]).

stEnergyTimepoint: Heat energy on the cutoff date (see ST_MBus_Info [▸ 205]).

stCounterInput1: Count value input 1 on the cutoff date (see ST_MBus_Info [▸ 205]).

stCounterInput2: Count value input 2 on the cutoff date (see ST_MBus_Info [▸ 205]).

VAR_IN_OUT

```
stCom : ST_MBUS_Communication;
```

stCom: This structure is used to link the block FB_MBUSKL6781() [▸ 203] with the meter blocks (see ST_MBUS_Communication [▸ 203]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.1.35 Error codes

Value (hex)	Value (dec)	Value (enum)	Description
0x0000	0	eMBUS_no_error	No error is present at the block. The block is currently not querying a counter.
0x0001	1	eMBUS_busy	The block is querying a meter.
0x0003	3	eMBUS_Disabled	The block is deselected.
0x0004	4	eMBUS_FBKL6781_Disabled	The function block FB_MBUSKL6781() [▸ 19] is deselected.
0x0065	101	eMBUSERROR_CIField_wrong_72hex_expected	The 7th byte in the response telegram contains the CI field. In this byte the hexadecimal number 72 is expected. It stands for variable data structure, low byte is sent first. Only this data structure is supported.
0x0066	102	eMBUSERROR_no_data_received	No data was received.
0x0067	103	eMBUSERROR_error_checksum	The response telegram includes a checksum (sum of all bytes from byte 5). The received checksum does not match the calculated checksum. This happens if the protocol was not received cleanly (e.g. in the event of interference on the cable or if the cable is too long).
0x0068	104	eMBUSERROR_error_in_head_data	The first 4 bytes are not included in the checksum. These 4 bytes are monitored separately.
0x0069	105	eMBUSERROR_usiAddress_over_250	Addresses higher than 250 are not permitted. The input *usiAddress* of the meter block was assigned a value higher than 250 (exception: Address 254. This address can be used if only one meter is connected).
0x006A	106	eMBUSERROR_send_error	Error message for error during sending.
0x006C	108	eMBUSERROR_received_address_wrong	Received address does not match the sent address.
0x006D	109	eMBUSERROR_cMBUS_MaxCom_below_1	Reserve.
0x006E	110	eMBUSERROR_iComId_over_cMBUS_MaxCom	Reserve.
0x006F	111	eMBUSERROR_manufacturer_sign_wrong	The response telegram includes a manufacturer code. This code is allocated to the counter blocks. This message appears if the received manufacturer code does not match the block used.
0x0070	112	eMBUSERROR_baudrate_wrong	Input *eBaudrate* of the block was assigned invalid values. Only E_MBUS_Baudrate [▸ 198] are allowed.
0x0071	113	eMBUSERROR_ReceiveBufferFull	The receive buffer of the serial interface is full. This may happen with long telegrams and/or long cycle times. The PLC is unable to read the data quick enough from the receive buffer, resulting in data loss. The situation may be resolved by reducing the cycle time.
0x0072	114	eMBUSERROR_E5hex_no_received	No single character E5 hexadecimal was received after initialization of the meter.
0x0073	115	eMBUSERROR_no_stop_character	No end character in the data array.
0x0074	116	eMBUSERROR_length_wrong	Number of received characters <> the length field.

Value (hex)	Value (dec)	Value (enum)	Description
0x0075	117	eMBUSERROR_wrong_terminal	Incorrect terminal connected
0x0076	118	eMBUSERROR_Terminal_is_not_initialized	The terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Everything corrected, everything translated and re-read into the System Manager?
0x0077	119	eMBUSERROR_stSecAdr_udiIdNumber_wrong	The input variable *stSecAdr.udiIdNumber* is not assigned.
0x0078	120	eMBUSERROR_missing_parts_telegram	Not all telegram values were received.
0x0079	121	eMBUSERROR_no_stop_character_received	No stop characters were received (16hex).
0x007A	122	eMBUSERROR_too_many_characters	Too many characters were received.
0x007B	123	eMBUSERROR_TimeOut_FB_KL6781	Timeout *FB_KL6781*.
0x007C	124	eMBUSERROR_TimeOut_MeterFB	Meter block timeout.
0x00C9	201	eMBUSERROR_COM_PARAMETERCHANGED	Input parameters have changed during reception.
0x00CA	202	eMBUSERROR_COM_TXBUFFOVERRUN	String > transfer buffer.
0x00D2	210	eMBUSERROR_COM_STRINGOVERRUN	End of the string.
0x00D3	211	eMBUSERROR_COM_ZEROCHARINVALID	String may not contain any zero characters.
0x00DC	220	eMBUSERROR_COM_INVALIDPOINTER	Invalid data pointer, e.g. zero.
0x00DD	221	eMBUSERROR_COM_INVALIDRXPOINTER	Invalid data pointer for *ReceiveData*.
0x00DE	222	eMBUSERROR_COM_INVALIDRXLENGTH	Invalid length for *ReceiveData* e.g. zero.
0x00DF	223	eMBUSERROR_COM_DATASIZEOVERRUN	End of the data block.
0x1001	4097	eMBUSERROR_COM_INVALIDBAUDRATE	Invalid baud rate.
0x1002	4098	eMBUSERROR_COM_INVALIDNUMDATABITS	Invalid data bits.
0x1003	4099	eMBUSERROR_COM_INVALIDNUMSTOPBITS	Invalid stop bits.
0x1004	4100	eMBUSERROR_COM_INVALIDPARITY	Invalid parity.
0x1005	4101	eMBUSERROR_COM_INVALIDHANDSHAKE	Invalid handshake.
0x1006	4102	eMBUSERROR_COM_INVALIDNUMREGISTERS	Invalid num register.
0x1007	4103	eMBUSERROR_COM_INVALIDREGISTER	Invalid register.
0x1008	4109	eMBUSERROR_COM_TIMEOUT	COM timeout.

4.2 DUTs

4.2.1 Enums

4.2.1.1 E_MBUS_Baudrate

Configurable baud rates

```
TYPE E_MBUS_Baudrate :
(
  eMBUS_NoBaudrate := 0,
  eMBUS_Baud300    := 30,
  eMBUS_Baud600    := 60,
  eMBUS_Baud1200   := 120,
  eMBUS_Baud2400   := 240,
  eMBUS_Baud4800   := 480,
  eMBUS_Baud9600   := 960
)
END_TYPE
```

eMBUS_NoBaudrate: Standard baud rate = 2400 baud

eMBUS_Baud300: 300 baud

eMBUS_Baud600: 600 baud

eMBUS_Baud1200: 1200 baud

eMBUS_Baud2400: 2400 baud

eMBUS_Baud4800: 4800 baud

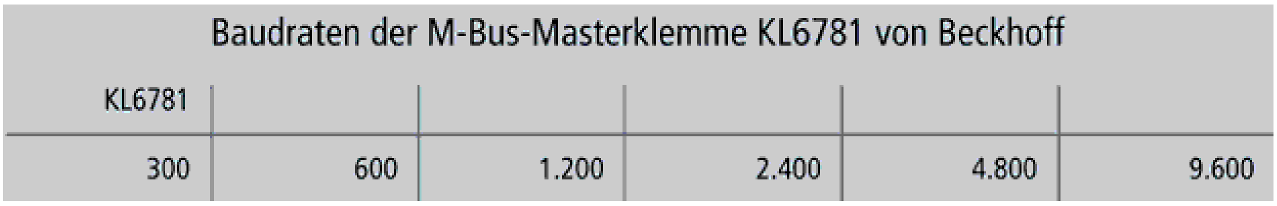
eMBUS_Baud9600: 9600 baud

M-Bus counters are generally supplied with 2400 baud.

The KL6781 supports 300, 600, 1200, 2400, 4800, 9600 baud.

The KL6781 interface is set to *eBaudrate* when the PLC starts or when the input *eBaudrate* changes.

Not all M-Bus devices support baud rates above 2400.



Baudraten der M-Bus-Masterklemme KL6781 von Beckhoff

KL6781					
300	600	1.200	2.400	4.800	9.600

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.1.2 E_MBUS_Error

Error messages.

```
TYPE E_MBUS_Error :
(
  eMBUS_no_error                         := 0,
  eMBUS_busy                             := 1,
  eMBUS_Disabled                         := 3,
  eMBUS_FBKL6781_Disabled                := 4,
  eMBUSERROR_CIField_wrong_72hex_expected := 101,
  eMBUSERROR_no_data_received             := 102,
  eMBUSERROR_error_checksum               := 103,
  eMBUSERROR_error_in_head_data           := 104,
  eMBUSERROR_usiAddress_over_250          := 105,
  eMBUSERROR_send_error                   := 106,
  eMBUSERROR_received_address_wrong       := 108,
  eMBUSERROR_cMBUS_MaxCom_below_1         := 109,
  eMBUSERROR_iComId_over_cMBUS_MaxCom     := 110,
  eMBUSERROR_manufacturer_sign_wrong      := 111,
  eMBUSERROR_baudrate_wrong               := 112,
  eMBUSERROR_ReceiveBufferFull            := 113,
  eMBUSERROR_E5hex_no_received            := 114,
  eMBUSERROR_no_stop_character            := 115,
  eMBUSERROR_length_wrong                 := 116,
  eMBUSERROR_wrong_terminal               := 117,
  eMBUSERROR_Terminal_is_not_initialized  := 118,
  eMBUSERROR_stSecAdr_udiIdNumber_wrong   := 119,
  eMBUSERROR_missing_parts_telegram       := 120,
  eMBUSERROR_no_stop_character_received   := 121,
  eMBUSERROR_too_many_characters          := 122,
  eMBUSERROR_TimeOut_FB_KL6781            := 123,
  eMBUSERROR_TimeOut_MeterFB              := 124,

  eMBUSERROR_COM_PARAMETERCHANGED         := 201,
  eMBUSERROR_COM_TXBUFFOVERRUN            := 202,
  eMBUSERROR_COM_STRINGOVERRUN            := 210,
  eMBUSERROR_COM_ZEROCHARINVALID          := 211,
  eMBUSERROR_COM_INVALIDPOINTER           := 220,
  eMBUSERROR_COM_INVALIDRXPOINTER         := 221,
  eMBUSERROR_COM_INVALIDRXLENGTH          := 222,
  eMBUSERROR_COM_DATASIZEOVERRUN          := 223,
  eMBUSERROR_COM_INVALIDBAUDRATE          := 16#1001,
  eMBUSERROR_COM_INVALIDNUMDATABITS       := 16#1002,
  eMBUSERROR_COM_INVALIDNUMSTOPBITS       := 16#1003,
  eMBUSERROR_COM_INVALIDPARITY            := 16#1004,
  eMBUSERROR_COM_INVALIDHANDSHAKE         := 16#1005,
  eMBUSERROR_COM_INVALIDNUMREGISTERS      := 16#1006,
  eMBUSERROR_COM_INVALIDREGISTER          := 16#1007,
  eMBUSERROR_COM_TIMEOUT                  := 16#1008
)
END_TYPE
```

eMBUS_no_error: No error at the block. The block is currently not querying a meter.

eMBUS_busy: The block is querying a meter.

eMBUS_Disabled: The block is deselected.

eMBUS_FBKL6781_Disabled: The block FB_MBUS_KL6781() [▶ 19] is deselected.

eMBUSERROR_CIField_wrong_72hex_expected: The 7th byte in the response telegram contains the CI field. In this byte the hexadecimal number 72 is expected. It stands for variable data structure, low byte is sent first. Only this data structure is supported.

eMBUSERROR_no_data_received: No data was received.

eMBUSERROR_error_checksum: The response telegram includes a checksum (sum of all bytes from byte 5). The received checksum does not match the calculated checksum. This happens if the protocol was not received cleanly (e.g. in the event of interference on the cable or if the cable is too long).

eMBUSERROR_error_in_head_data: The first 4 bytes are not included in the checksum. These 4 bytes are monitored separately.

eMBUSERROR_usiAddress_over_250: Addresses higher than 250 are not permitted. The input *usiAddress* of the meter block was assigned a value higher than 250 (exception: Address 254. This address can be used if only one meter is connected).

eMBUSERROR_send_error: Error message for error during sending.

eMBUSERROR_received_address_wrong: Received address does not match the sent address.

eMBUSERROR_cMBUS_MaxCom_below_1: Reserve.

eMBUSERROR_iComId_over_cMBUS_MaxCom: Reserve.

eMBUSERROR_manufacturer_sign_wrong: The response telegram includes a manufacturer code. This code is allocated to the counter blocks. This message appears if the received manufacturer code does not match the block used.

eMBUSERROR_baudrate_wrong: Input *eBaudrate* of the block was assigned invalid values. Only E_MBUS_Baudrate [▶ 198] are allowed.

eMBUSERROR_ReceiveBufferFull: The receive buffer of the serial interface is full. This may happen with long telegrams and/or long cycle times. The PLC is unable to read the data quick enough from the receive buffer, resulting in data loss. The situation may be resolved by reducing the cycle time.

eMBUSERROR_E5hex_no_received: No single character E5 hexadecimal was received after initialization of the meter.

eMBUSERROR_no_stop_character: No end character in the data array.

eMBUSERROR_length_wrong: Number of received characters <> the length field.

eMBUSERROR_wrong_terminal: Incorrect terminal connected

eMBUSERROR_Terminal_is_not_initialized: The terminal is not initialized. This message usually means that there is no connection to the terminal. Terminal linked to the variables in the System Manager? Terminal plugged in incorrectly? Everything corrected, everything translated and re-read into the System Manager?

eMBUSERROR_stSecAdr_udiIdNumber_wrong: The input variable *stSecAdr.udiIdNumber* is not assigned

eMBUSERROR_missing_parts_telegram: Not all telegram values were received.

eMBUSERROR_no_stop_character_received: No stop characters were received (16hex).

eMBUSERROR_too_many_characters: Too many characters were received.

eMBUSERROR_TimeOut_FB_KL6781: Timeout *FB_KL6781*.

eMBUSERROR_TimeOut_MeterFB: Meter block timeout.

eMBUSERROR_COM_PARAMETERCHANGED: Input parameters have changed during reception.

eMBUSERROR_COM_TXBUFFOVERRUN: String > transfer buffer.

eMBUSERROR_COM_STRINGOVERRUN: End of the string.

eMBUSERROR_COM_ZEROCHARINVALID: String may not contain any zero characters.

eMBUSERROR_COM_INVALIDPOINTER: Invalid data pointer, e.g. zero.

eMBUSERROR_COM_INVALIDRXPOINTER: Invalid data pointer for ReceiveData.

eMBUSERROR_COM_INVALIDRXLENGTH: Invalid length for ReceiveData e.g. zero.

eMBUSERROR_COM_DATASIZEOVERRUN: End of the data block.

eMBUSERROR_COM_INVALIDBAUDRATE: Invalid baud rate.

eMBUSERROR_COM_INVALIDNUMDATABITS: Invalid data bits.

eMBUSERROR_COM_INVALIDNUMSTOPBITS: Invalid stop bits.

eMBUSERROR_COM_INVALIDPARITY: Invalid parity.

eMBUSERROR_COM_INVALIDHANDSHAKE: Invalid handshake.

eMBUSERROR_COM_INVALIDNUMREGISTERS: Invalid num register.

eMBUSERROR_COM_INVALIDREGISTER: Invalid register.

eMBUSERROR_COM_TIMEOUT: COM timeout.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.1.3 E_MBus_Fct

Value function.

```
TYPE E_MBus_Fct :
(
  eMBUS_ValueNull             := -1,
  eMBUS_InstantaneousValue    := 0,
  eMBUS_Max                   := 1,
  eMBUS_Min                   := 2,
  eMBUS_ValueDuringErrorState := 3,
  eMBUS_ManufacturerSpecific  := 256
)
END_TYPE
```

eMBUS_ValueNull: Not assigned.

eMBUS_InstantaneousValue: Instantaneous value.

eMBUS_Max: Maximum value.

eMBUS_Min: Minimum value.

eMBUS_ValueDuringErrorState: Faulty value.

eMBUS_ManufacturerSpecific: Manufacturer-specific.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.1.4 E_MBUS_Medium

Medium

```
TYPE E_MBUS_Medium :
(
  eMBUS_MediumNull                     := -1,
  eMBUS_MediumOther                    := 0,
  eMBUS_MediumOil                      := 1,
  eMBUS_MediumElectricity              := 2,
  eMBUS_MediumGas                      := 3,
  eMBUS_MediumHeat_Outlet              := 4,
  eMBUS_MediumSteam                    := 5,
  eMBUS_MediumHot_Water                := 6,
  eMBUS_MediumWater                    := 7,
  eMBUS_MediumHeat_Cost_Allocator      := 8,
  eMBUS_MediumCompressed_Air           := 9,
  eMBUS_MediumCooling_load_meter_outlet := 10,
  eMBUS_MediumCooling_load_meter_intlet := 11,
  eMBUS_MediumHeat_inlet               := 12,
  eMBUS_MediumHeat_cooling_load_Meter  := 13,
  eMBUS_MediumBusSystem                := 14,
  eMBUS_MediumUnknownMedium            := 15,
  eMBUS_MediumReserved16               := 16,
  eMBUS_MediumReserved17               := 17,
  eMBUS_MediumReserved18               := 18,
  eMBUS_MediumReserved19               := 19,
  eMBUS_MediumReserved20               := 20,
  eMBUS_MediumReserved21               := 21,
  eMBUS_MediumColdWater                := 22,
  eMBUS_MediumDualWater                := 23,
  eMBUS_MediumPressure                 := 24,
  eMBUS_MediumA_D_Converter            := 25,
  eMBUS_MediumReserved26              := 26,
  eMBUS_MediumReserved27              := 27,
  eMBUS_MediumReserved28              := 28,
  eMBUS_MediumReserved29              := 29,
  eMBUS_MediumReserved30              := 30
)
END_TYPE
```

eMBUS_MediumNull: Not assigned.

eMBUS_MediumOther: Other.

eMBUS_MediumOil: Oil.

eMBUS_MediumElectricity: Electricity.

eMBUS_MediumGas: Gas.

eMBUS_MediumHeat_Outlet: Heat (return).

eMBUS_MediumSteam: Steam.

eMBUS_MediumHot_Water: Hot water.

eMBUS_MediumWater: Water.

eMBUS_MediumHeat_Cost_Allocator: Heating cost distributor.

eMBUS_MediumCompressed_Air: Compressed air.

eMBUS_MediumCooling_load_meter_outlet: Cooling (return).

eMBUS_MediumCooling_load_meter_intlet: Cooling (supply/flow).

eMBUS_MediumHeat_inlet: Heat (supply/flow).

eMBUS_MediumHeat_cooling_load_Meter: Heating / cooling.

eMBUS_MediumBusSystem: Bus / system.

eMBUS_MediumUnknownMedium: Unknown.

eMBUS_MediumReserved16: Reserved.

eMBUS_MediumReserved17: Reserved.

eMBUS_MediumReserved18: Reserved.

eMBUS_MediumReserved19: Reserved.

eMBUS_MediumReserved20: Reserved.

eMBUS_MediumReserved21: Reserved.

eMBUS_MediumColdWater: Cold water.

eMBUS_MediumDualWater: Mixed water.

eMBUS_MediumPressure: Pressure.

eMBUS_MediumA_D_Converter: A/D converter.

eMBUS_MediumReserved26: Reserved.

eMBUS_MediumReserved27: Reserved.

eMBUS_MediumReserved28: Reserved.

eMBUS_MediumReserved29: Reserved.

eMBUS_MediumReserved30: Reserved.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2 Structures

4.2.2.1 ST_KL6781outData22B

Process image of the outputs.

Linked to the terminals in the System Manager.

```
TYPE ST_KL6781outData22B :
STRUCT
  Ctrl  : WORD;
  D     : ARRAY[0..21] OF BYTE;
END_STRUCT
END_TYPE
```

Ctrl: Control word.

D: 22 bytes for the output data of the M-Bus.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.2 ST_KL6781inData22B

Process image of the inputs.

Linked to the terminals in the System Manager.

```
TYPE ST_KL6781inData22B :
STRUCT
  Status  : WORD;
  D       : ARRAY[0..21] OF BYTE;
END_STRUCT
END_TYPE
```

Status: Status word.

D: 22 bytes for the input data of the M-Bus.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.3 ST_MBUS_Communication

Interne Struktur.

This structure is used to link the block FB_MBUSKL6781() [▸ 19] with the meter blocks.
ST_MBUS_Communication).

```
TYPE ST_MBUS_Communication :
STRUCT
  bStart           : BOOL;
  bBusy            : BOOL;
  bSND_NKE         : BOOL;
  bSend            : BOOL;
  bStartManuell    : BOOL;
  bBlockadeSecAdr  : BOOL;
  usiAddress       : USINT;
  byCField         : BYTE;
  stSecAdr         : ST_MBUS_SecAdr;
  eError           : E_MBUS_ERROR;
  eBaudrate        : E_MBUS_Baudrate := eMBUS_Baud2400;
  arrMBusLongFrame : ARRAY[1..260] OF BYTE;
  bySendByte       : BYTE;
  uiMaxCount       : UINT;
  uiCount          : UINT;
  stKomRxBuffer    : ST_KL6781ComBuffer;
  stKomTxBuffer    : ST_KL6781ComBuffer;
END_STRUCT
END_TYPE
```

bStart: Start.

bBusy: This bit is set for as long as the block is active.

bSND_NKE: SND_NKE is sent.

bSend: Data is being sent.

bStartManuell: Manual start.

bBlockadeSecAdr: Secondary addressing results in blocking.

usiAddress: Primary address.

byCField: C field.

stSecAdr: Secondary address (see ST_MBUS_SecAdr [▸ 206]).

eError: Error number (see E_MBUS_ERROR [▸ 198]).

eBaudrate: Baud rate (see E_MBUS_Baudrate [▸ 198]).

arrMBusLongFrame: Sent or received bytes.

bySendByte: Number of sent bytes.

uiMaxCount: Maximum number of read commands.

uiCount: Current read command.

stKomRxBuffer: Receive buffer.

stKomTxBuffer: Send buffer.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.4 ST_MBus_Data

Value information.

```
TYPE ST_MBus_Data :
STRUCT
  sValue   : STRING(25);
  sUnit    : STRING(20);
  sInfo    : STRING;
  eFct     : E_MBus_Fct;
  iTariff  : INT;
  iStorNo  : INT;
  iUnit    : INT;
  byVIFE   : BYTE;
END_STRUCT
END_TYPE
```

sValue: Value.

sUnit: Unit.

sInfo: Information.

eFct: Function (see E_MBus_Fct [▸ 201]).

iTariff: Tariff.

iStorNo: Memory number.

iUnit: Unit (integer).

byVIFE: VIFE.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.5 ST_MBus_Data2

Structure of the output values in the block FB_MBUS_General_Ext() [▸ 80].

```
TYPE ST_MBus_Data2 :
STRUCT
  arrData : ARRAY[1..cMBUS_MaxData] OF ST_MBus_Data;
END_STRUCT
END_TYPE
```

arrData: Values.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.6 ST_MBus_Info

Value information.

```
TYPE ST_MBus_Info :
STRUCT
  sValue : STRING(25);
  sUnit  : STRING(20);
  eFct   : E_MBus_Fct;
END_STRUCT
END_TYPE
```

sValue: Value as string.

sUnit: Unit of the value as string.

eFct: Function (see E_MBus_Fct [▸ 201]).

M-Bus devices may supply very large values, which cannot be displayed or can only be displayed inaccurately as numbers on BC/BX systems. The values are therefore supplied as strings (*sValue*).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.7 ST_MBUS_SecAdr

Secondary address of a meter.

```
TYPE ST_MBUS_SecAdr :
STRUCT
  udiIdNumber    : UDINT := 16#FFFFFFFF;
  uiManufacturer : UINT  := 16#FFFF;
  usiVersion     : USINT := 16#FF;
  usiMedium      : USINT := 16#FF;
END_STRUCT
END_TYPE
```

udiIdNumber: Serial number of the meter.

uiManufacturer: Manufacturer code.

usiVersion: Counter software version.

usiMedium: Medium.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.8 ST_MBus_Scan

Scanning information.

```
TYPE ST_MBus_Scan :
STRUCT
  usiAddress : USINT;
  dwIdNumber : DWORD;
  byStatus   : BYTE;
  eMedium    : E_MBUS_Medium;
  sMan       : STRING(3);
  byGEN      : BYTE;
END_STRUCT
END_TYPE
```

usiAddress: Primary address [▶ 11] of the meter.

dwIdNumber: Serial number of the meter (secondary address)

byStatus: Status.

eMedium: Medium (see E_MBUS_Medium [▶ 201]).

sMan: Manufacturer code.

byGEN: Software version of the device.

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.9 Hydrometer

Data types	Description
ST_MBUS_DueDayHYD1 [▸ 207]	Structure of the cutoff date values in the block FB_MBUS_HYD_Sharky_00 [▸ 104]

4.2.2.9.1 ST_MBUS_DueDayHYD1

Structure of the cutoff date values in the block FB_MBUS_HYD_Sharky_00() [▸ 104].

```
TYPE ST_MBUS_DueDayHYD1 :
STRUCT
  stEnergy           : ST_MBus_Info;
  stVolume           : ST_MBus_Info;
  stTariff1          : ST_MBus_Info;
  stTariff2          : ST_MBus_Info;
  stDate             : ST_MBus_Info;
  stDateFutureDueDay : ST_MBus_Info;
END_STRUCT
END_TYPE
```

stEnergy: Energy meter reading (see ST_MBus_Info [▸ 205]).

stVolume: Volume meter reading (see ST_MBus_Info [▸ 205]).

stTariff1: Meter reading tariff 1 (see ST_MBus_Info [▸ 205]).

stTariff2: Meter reading tariff 2 (see ST_MBus_Info [▸ 205]).

stDate: Cutoff date (see ST_MBus_Info [▸ 205]).

stDateFutureDueDay: Future cutoff date (see ST_MBus_Info [▸ 205]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.2.2.10 Metrima

Data types	Description
ST_MBus_F22 [▸ 207]	Structure of the monthly values in the block FB_MBUS_SVM_F22_Ext [▸ 145]

4.2.2.10.1 ST_MBus_F22

Structure of the monthly values in the block FB_MBUS_SVM_F22_Ext() [▸ 145].

```
TYPE ST_MBus_F22 :
STRUCT
  stEnergy        : ST_MBus_Info;
  stVolume        : ST_MBus_Info;
  stVolume2       : ST_MBus_Info;
  stPulsecounter1 : ST_MBus_Info;
  stPulsecounter2 : ST_MBus_Info;
  stDate          : ST_MBus_Info;
END_STRUCT
END_TYPE
```

stEnergy: Energy meter reading (see ST_MBus_Info [▸ 205]).

stVolume: Volume meter reading (see ST_MBus_Info [▸ 205]).

stVolume2: Volume meter reading (see ST_MBus_Info [▸ 205]).

stPulsecounter1: Meter reading pulse counter 1 (see ST_MBus_Info [▸ 205]).

stPulsecounter2: Meter reading pulse counter 2 (see ST_MBus_Info [▸ 205]).

stDate: Date (see ST_MBus_Info [▸ 205]).

Requirements

Development environment	required TC3 PLC library
TwinCAT from v3.1.4020.14	Tc2_MBus from 3.3.4.0

4.3 GLVs

4.3.1 Globale_Variablen_MBUS

If they are declared in the program, a warning message is generated during program compilation, since the constant already exists. This warning can be ignored.

```
VAR_GLOBAL CONSTANT
  cMBUS_MaxData       := 40,
  cMBUS_MaxTelegrams := 5,
  cMBUS_MaxDataParam := 10,
END_VAR
```

cMBUS_MaxData: The constant applies to all instances of the blocks FB_MBUS_General() [▸ 77], FB_MBUS_General_Ext() [▸ 80] and FB_MBUS_General_Param() [▸ 84]. It indicates the maximum data volume expected in a meter telegram.

cMBUS_MaxTelegrams: The constant applies to all instances of the FB_MBUS_General_Ext block() [▸ 80]. It indicates the maximum number of telegrams to be expected.

cMBUS_MaxDataParam: The constant applies to all instances of the FB_MBUS_General_Param blocks() [▸ 84]. It indicates the maximum number of values to be displayed by the instances of block FB_MBUS_General_Param() [▸ 84].

4.4 Integration into TwinCAT

4.4.1 KL6781 with CX5120

This example describes how a simple PLC program for M-Bus can be written in TwinCAT and how it is linked with the hardware. The task is to read a counter with four digital inputs.

Example: https://infosys.beckhoff.com/content/1033/tcplclib_tc2_mbus/Resources/6218378891/.zip

Hardware

Setting up the components

The following hardware is required:

- 1x CX5120 Embedded PC
- 1x KL6781 M-Bus master terminal
- 1x KL9010 end terminal

Set up the hardware and the M-Bus components as described in the associated documentation.

This example assumes that the counter address is known.

Software

Creation of the PLC program

Create a new "TwinCAT XAE project" and a "Standard PLC project".

Add the library Tc2_MBus under **References** in the PLC project.

Generate a Global Variable List with the name GVL_MBus and create the following variables:

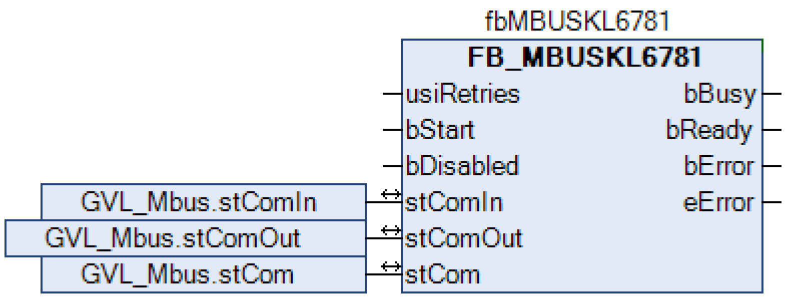
```
VAR_GLOBAL
    stComIn     AT %I* : ST_KL6781inData22B;
    stComOut    AT %Q* : ST_KL6781outData22B;
    stCom              : ST_MBUS_Communication;
END_VAR
```

stComIn: Input variable for the M-Bus terminal (see ST_KL6781inData22B [▸ 203]).

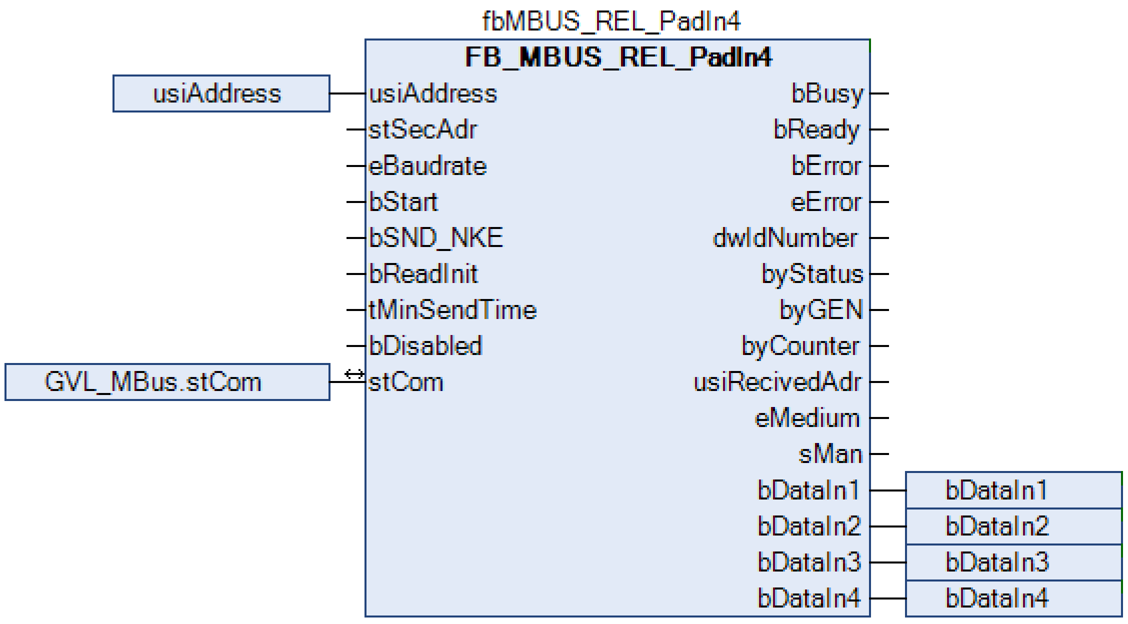
stComOut: Output variable for the M-Bus terminal (see ST_KL6781outData22B [▸ 203])

stCom: Required for the communication with M-Bus (see ST_MBUS_Communication [▸ 203]).

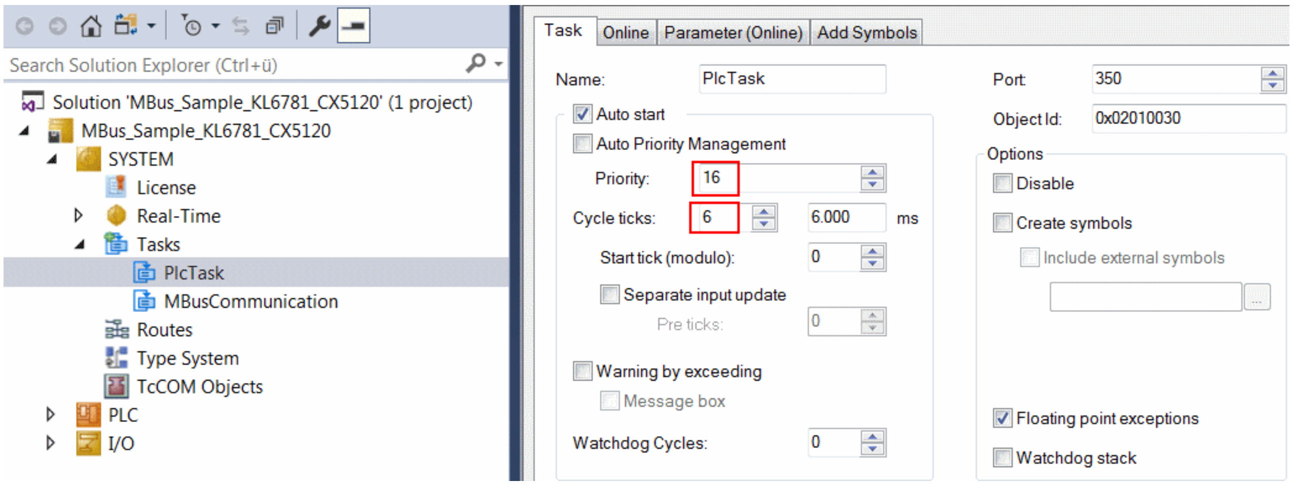
Create a program (CFC) for the background communication with M-Bus. The FB_MBUSKL6781 [▸ 19] block is called in this program. Make sure to link the communication block with stComIn, stComOut and stCom.



Create a MAIN program (CFC) in which the block FB_MBUS_REL_PadIn4 [▸ 157] is called up. Link the input *usiAddress* of the counter block with the local variable *usiAddress* and *stCom* with the global variable *stCom*.



Navigate to the task configuration section and configure the PlcTask. By way of example, the task is assigned priority 16 and a cycle time of 6 ms.



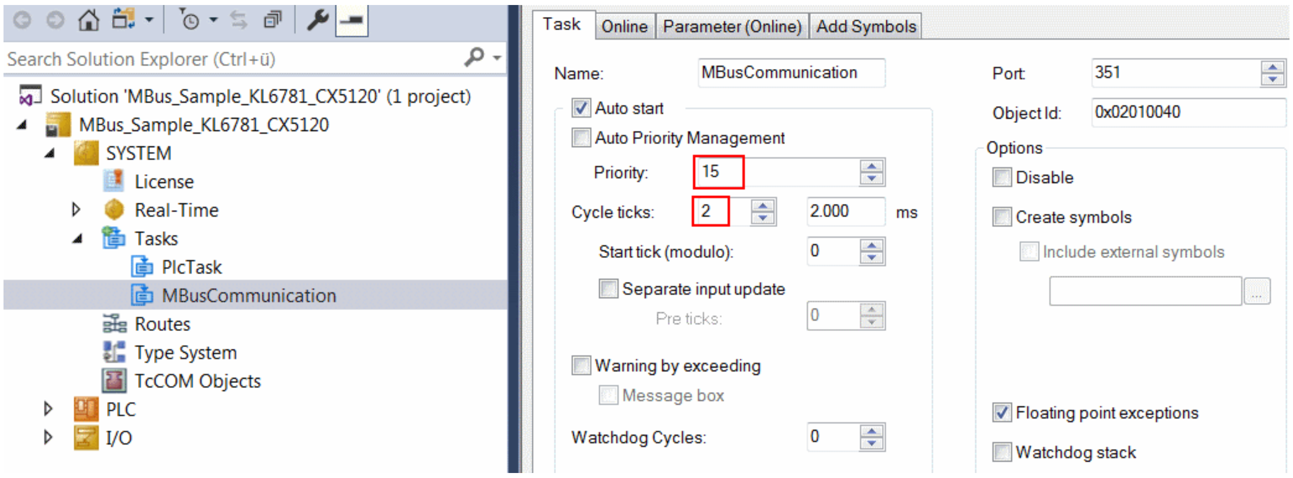
Search Solution Explorer (Ctrl+ü)

- Solution 'MBus_Sample_KL6781_CX5120' (1 project)
  - MBus_Sample_KL6781_CX5120
    - SYSTEM
      - License
      - Real-Time
      - Tasks
        - PlcTask
        - MBusCommunication
      - Routes
      - Type System
      - TcCOM Objects
    - PLC
    - I/O

Task | Online | Parameter (Online) | Add Symbols

Name:	PlcTask	Port:	350
[✓] Auto start		Object Id:	0x02010030
[ ] Auto Priority Management		Options	
Priority:	16	[ ] Disable	
Cycle ticks:	6 / 6.000 ms	[ ] Create symbols	
Start tick (modulo):	0	[ ] Include external symbols	
[ ] Separate input update			
Pre ticks:	0		
[ ] Warning by exceeding			
[ ] Message box		[✓] Floating point exceptions	
Watchdog Cycles:	0	[ ] Watchdog stack	

Create a further task for the background communication. Assign a higher priority (smaller number) and a lower interval time to this task than the PLCTask.



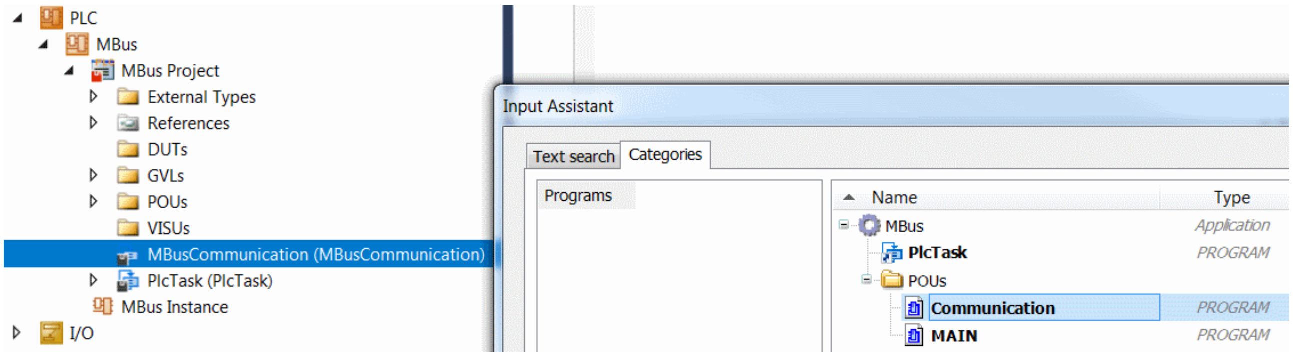
Search Solution Explorer (Ctrl+ü)

- Solution 'MBus_Sample_KL6781_CX5120' (1 project)
  - MBus_Sample_KL6781_CX5120
    - SYSTEM
      - License
      - Real-Time
      - Tasks
        - PlcTask
        - MBusCommunication
      - Routes
      - Type System
      - TcCOM Objects
    - PLC
    - I/O

Task | Online | Parameter (Online) | Add Symbols

Name:	MBusCommunication	Port:	351
[✓] Auto start		Object Id:	0x02010040
[ ] Auto Priority Management		Options	
Priority:	15	[ ] Disable	
Cycle ticks:	2 / 2.000 ms	[ ] Create symbols	
Start tick (modulo):	0	[ ] Include external symbols	
[ ] Separate input update			
Pre ticks:	0		
[ ] Warning by exceeding			
[ ] Message box		[✓] Floating point exceptions	
Watchdog Cycles:	0	[ ] Watchdog stack	

Add the program for the communication to this task. Further information on task configuration can be found in the description of the function block FB_MBUSKL6781 [▸ 19].



- PLC
  - MBus
    - MBus Project
      - External Types
      - References
      - DUTs
      - GVLs
      - POUs
      - VISUs
      - MBusCommunication (MBusCommunication)
      - PlcTask (PlcTask)
    - MBus Instance
- I/O

Input Assistant

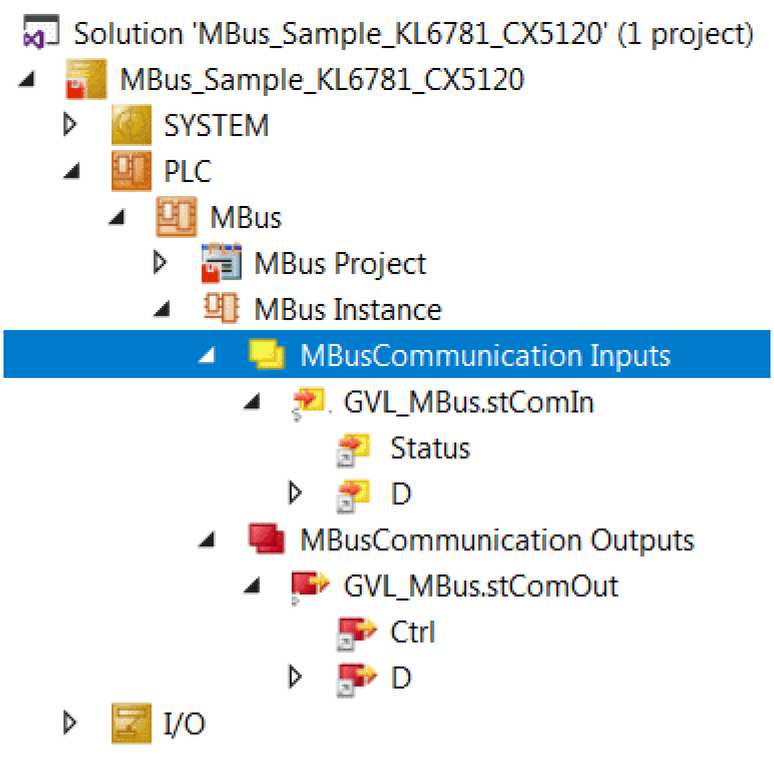
Text search | Categories

Programs

Name	Type
MBus	Application
PlcTask	PROGRAM
POUs	
Communication	PROGRAM
MAIN	PROGRAM

I/O configuration

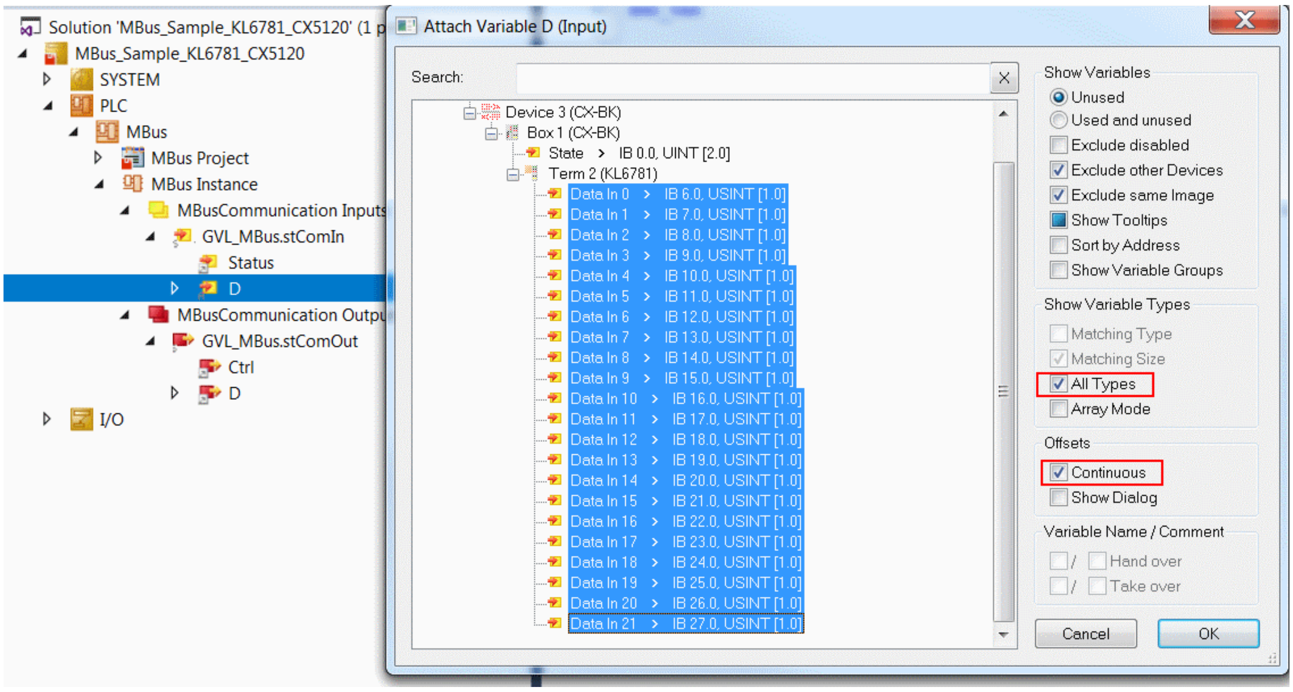
Select the CX as target system and initiate a search for its hardware. In the project instance within the PLC section, you can see that the input and output variables are assigned to the corresponding tasks.



Solution 'MBus_Sample_KL6781_CX5120' (1 project)
- MBus_Sample_KL6781_CX5120
  - SYSTEM
  - PLC
    - MBus
      - MBus Project
      - MBus Instance
        - MBusCommunication Inputs
          - GVL_MBus.stComIn
            - Status
            - D
        - MBusCommunication Outputs
          - GVL_MBus.stComOut
            - Ctrl
            - D
  - I/O

Now link the global variables of PLC program with the inputs and outputs of the Bus Terminals. Create the Solution and enable the configuration.

When linking the data array, make sure that you select both the **All types** and **Continuous** options.
Use the Shift key and the right mouse button to mark all data bytes of the terminal.



Attach Variable D (Input)

Search:

```
Device 3 (CX-BK)
  Box 1 (CX-BK)
    State      >  IB 0.0, UINT [2.0]
    Term 2 (KL6781)
      Data In 0   >  IB 6.0, USINT [1.0]
      Data In 1   >  IB 7.0, USINT [1.0]
      Data In 2   >  IB 8.0, USINT [1.0]
      Data In 3   >  IB 9.0, USINT [1.0]
      Data In 4   >  IB 10.0, USINT [1.0]
      Data In 5   >  IB 11.0, USINT [1.0]
      Data In 6   >  IB 12.0, USINT [1.0]
      Data In 7   >  IB 13.0, USINT [1.0]
      Data In 8   >  IB 14.0, USINT [1.0]
      Data In 9   >  IB 15.0, USINT [1.0]
      Data In 10  >  IB 16.0, USINT [1.0]
      Data In 11  >  IB 17.0, USINT [1.0]
      Data In 12  >  IB 18.0, USINT [1.0]
      Data In 13  >  IB 19.0, USINT [1.0]
      Data In 14  >  IB 20.0, USINT [1.0]
      Data In 15  >  IB 21.0, USINT [1.0]
      Data In 16  >  IB 22.0, USINT [1.0]
      Data In 17  >  IB 23.0, USINT [1.0]
      Data In 18  >  IB 24.0, USINT [1.0]
      Data In 19  >  IB 25.0, USINT [1.0]
      Data In 20  >  IB 26.0, USINT [1.0]
      Data In 21  >  IB 27.0, USINT [1.0]
```

Show Variables
- ● Unused
- ○ Used and unused
- ☐ Exclude disabled
- ☑ Exclude other Devices
- ☑ Exclude same Image
- ■ Show Tooltips
- ☐ Sort by Address
- ☐ Show Variable Groups

Show Variable Types
- ☐ Matching Type
- ☑ Matching Size
- ☑ All Types
- ☐ Array Mode

Offsets
- ☑ Continuous
- ☐ Show Dialog

Variable Name / Comment
- ☐ / ☐ Hand over
- ☐ / ☐ Take over

After starting the PLC, the current values are regularly read by the counter.

5 Appendix

5.1 Support and Service

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

Beckhoff's branch offices and representatives

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

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

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

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

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

More Information:
www.beckhoff.com/te1000/

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