

WAGO→I/O→SYSTEM 750

Library for Building Automation

Function Block Description for M-Bus

Last Update: 25.04.2019



Copyright © 2019 by WAGO Kontakttechnik GmbH & Co. KG
All rights reserved.

WAGO Kontakttechnik GmbH & Co. KG

Hansastraße 27
32423 Minden
Germany

Phone: +49 (0) 571/8 87 – 0
Fax: +49 (0) 571/8 87 – 1 69
E-mail: info@wago.com
Web: <http://www.wago.com>

Technical Support

Phone: +49 (0) 571/8 87 – 4 45 55
Fax: +49 (0) 571/8 87 – 84 45 55
E-mail: support@wago.com

Every conceivable measure has been taken to ensure the correctness and completeness of this documentation. However, as errors can never be fully excluded we would appreciate any information or ideas at any time.

We wish to point out that the software and hardware terms as well as the trademarks of companies used and/or mentioned in the present manual are generally trademark or patent protected.

WAGO-I/O-PRO Library for M-Bus

List of contents

Important Comments	4
Copyright.....	4
Personnel Qualification	4
Intended Use.....	4
Communication	5
M-Bus Master (FbMbusMaster)	5
Meter	7
M-Bus Water Meter (FbMbus_Water)	7
M-Bus Electricity Meter (FbMbus_Electricity)	9
M-Bus Heat Meter (FbMbus_Heat)	11
Additional function	14
General M-Bus Meter (FbMbus_General)	14
M-Bus Multi Telegram Meter (FbMbus_MultiTel)	16
M-Bus RawDevice (FbMbus_RawDevice)	18
M-Bus Send data (FbMbusSend)	20
Unit Converter	23
M-Bus Unit Converter (FbUnitConverter)	23
Addition	25
M-Bus Version.....	25

Important Comments

To ensure fast installation and start-up of the units described in this manual, we strongly recommend that the following information and explanation is carefully read and adhered to.

Copyright

This manual is copyrighted, together with all figures and illustrations contained therein. Any use of this manual which infringes the copyright provisions stipulated herein, is not permitted. Reproduction, translation and electronic and photo-technical archiving and amendments require the written consent of WAGO Kontakttechnik GmbH & Co. KG. Non-observance will entail the right of claims for damages.

Personnel Qualification

The use of the product detailed in this manual is exclusively geared to specialists having qualifications in PLC programming, electrical specialists or persons instructed by electrical specialists who are also familiar with the valid standards. WAGO Kontakttechnik GmbH & Co. KG declines all liability resulting from improper action and damage to WAGO products and third party products due to non-observance of the information contained in this manual.

Intended Use

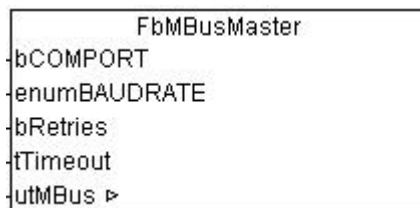
For each individual application, the components supplied are to work with a dedicated hardware and software configuration. Modifications are only admitted within the framework of the possibilities documented in the manuals. All other changes to the hardware and/or software and the non-conforming use of the components entail the exclusion of liability on part of WAGO Kontakttechnik GmbH & Co. KG.

Please direct any requirements pertaining to a modified and/or new hardware or software configuration directly to WAGO Kontakttechnik GmbH & Co. KG.

Communication

M-Bus Master (FbMbusMaster)

WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMbusMaster	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:		Mbus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
bCOMPORT		BYTE	Serial interface port number (M-Bus level converter module) Default setting = 2
enumBAUDRATE		COM_BAUDRATE	Baud rate of the M-Bus meter BAUD_1200 := 120 BAUD_2400 := 240 BAUD_4800 := 480 BAUD_9600 := 960 BAUD_19200 := 1920 Default setting = BAUD_2400
bRetries		BYTE	M-Bus retries Default setting = 3
tTimeout		TIME	M-Bus communication timeout Default setting = t#1000ms
Input/output parameter:		Data type:	Comment:
utMbus		typMbus	M-Bus parameter and receive buffer
.bPADR		BYTE	M-Bus primary address
.xSND_NKE		BOOL	Normalization command first
.xSTART		BOOL	Start toggle variable (sets caller)
.xBUSY		BOOL	Interface is busy (sets caller)
.xCOM_BUSY		BOOL	M-Bus communication is busy
.aobBUFFER		ARRAY [0..261] OF BYTE	M-Bus telegram receive buffer
.bBUFFERLENGTH		BYTE	Characters received in “aobBUFFER“
.xERROR		BOOL	Error indication
.bCOM_ERROR		BYTE	Interface error (see SERCOMM)
.bL1_ERROR		BYTE	M-Bus layer 1 error (Timeout, CRC, ...)

Graphical display:

Function description:

The function block **FbMbusMaster** is used for the communication with the serial (parameterizable) module 750-650/003-000 in connection with the fieldbus controllers 750-8xx. A level converter is required for the M-Bus connection.

The port numbers of the connected serial modules are identified and assigned independently by the controller from the left starting with COM2. The service interface of the controller (COM1) cannot be used for the M-Bus communication.

To address the function block to the level converter, the appropriate number (e.g. "2" for COM2) must be entered as a constant at input "**bCOMPORT**".

The input/output variable "**utMbus**" contains all necessary request parameters and data from the M-Bus telegrams received. This variables structure can be used by other function blocks for start and data processing. Using "**utMbus.xStart**", another function block can first start a M-Bus query if "**utMbus.xBusy**" is not set. This function block, on its part, must in turn set "**utMbus.xBusy**" to indicate that it is just using the communication module. The address of the M-Bus participant to be addressed must be entered in "**utMbus.bPAdr**". "**utMbus.COM_Busy**" shows that the serial communication is active. After readout, the possible errors are indicated by the variable "**utMbus.xERROR**" set. As a result, a better error description can be returned by the variables "**utMbus.bCOM_ERROR**" and "**utMbus.bL1_Error**". If the "**utMbus.bCOM_ERROR**" value is different than 0, then there is a serial interface error. The error codification is the same as in the library "SERCOMM.LIB".

The "**utMbus.bL1_ERROR**" value is different than 0, when there is a M-Bus layer 1 error. The individual bit positions of these bytes are assigned to different errors. The following error assignment is obtained by numbering the bit positions from 0 to 7:

- Bit 7 – Timeout (meter gives no response)
- Bit 6 – not used
- Bit 5 – not used
- Bit 4 – Stop character is wrong
- Bit 3 – M-Bus CRC error
- Bit 2 – 2nd start character (0x68) wrong
- Bit 1 – different lengths – bytes
- Bit 0 – 1st start character (0x68) wrong

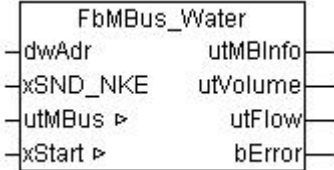
Important:

The function block **FbMbusMaster** can only be used in connection with the library "Serial_Interface_01.lib".

Meter

M-Bus Water Meter (FbMbus_Water)

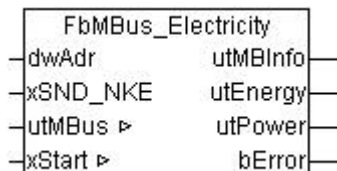
WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMBus_Water	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:		MBus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
dwAdr		DWORD	M-Bus primary address (>255 Sec.-Adr.)
xSND_NKE		BOOL	M-Bus normalisation command first
Input/output parameter:		Data type:	Comment:
utMBus		TypMBus	M-Bus parameter and receive buffer (see function block FbMBusMaster)
xStart		BOOL	Toggle variable for starting to read out the water meter. Automatically reset after read-out.
Feedback value:		Data type:	Comment:
utMBInfo		TypMBusInfo	Fixed meter data (header in data telegram with ID, manufacturer, medium, ..)
.bPAdr		BYTE	Primary address read out
.stID		STRING(8)	Secondary address (ID)
.stMAN		STRING(3)	3-digit manufacturer code
.stMED		STRING(14)	Medium to be measured
.bMGen		BYTE	Version number (generation)
.bMStatus		BYTE	M-Bus status byte
.bAllRecords		BYTE	Number of data records in the telegram
.bGoodRecords		BYTE	Analysed data records
utVolume		typMBusRecord	Measured water volume (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as atring
.Enum_Unit		typMBusUnit	Unit as constant (enumeration)
utFlow		typMBusRecord	Effective water flow (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		typMBusUnit	Unit as constant (enumeration)

bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...) 0x04 = no analysable records
Graphical display:		
		
Function description:		
<p>The function block FbMbus_Water is used to read out and decode M-Bus water meter data. Heat, cold, heat/cold, gas, steam, oil and mass meters also belong to this category. The function block can only be used in connection with the FbMbusMaster M-Bus communication block.</p> <p>Both function blocks are synchronized via the “utMbus” variable structure and must therefore be connected to each other.</p> <p>The input variable “dwAdr” is preallocated with the primary (<256) or secondary (>256) address of the M-Bus water meter. By setting “xSND_NKE”, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequential telegrams can respond with their 1st (important) telegram at the next request.</p> <p>The “xStart” input/output variable starts reading out the water meter. This variable is automatically reset after read-out.</p> <p>“bError” indicates whether the read-out has been performed without error.</p> <p>The volume and effective flow of the water meter are contained in the output variables “utVolume” and “utFlow” when the read-out has been performed successfully. Note that the flow is not delivered by all M-Bus water meters in their telegram. In this case, the string variables remain empty.</p> <p>To convert the meter reading into both a REAL type value and a desired unit, the function block FbUnitConverter described further below can be connected downstream by the user.</p> <p>Important:</p> <p>Only the meters with the following medium setting are analysed:</p> <p>water, hot water, cold water, dual water,</p> <p>oil, gas, steam, compressed air,</p> <p>heat, heat inlet, heat outlet, heat/cooling, cooling inlet, cooling outlet.</p>		

M-Bus Electricity Meter (FbMbus_Electricity)

WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMbus_Electricity	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:		Mbus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
dwAdr		DWORD	M-Bus primary address (>255 Sec.-Adr.)
xSND_NKE		BOOL	M-Bus normalisation command first
Input/output parameter:		Data type:	Comment:
xStart		BOOL	Toggle variable for starting to read out the electricity meter. Automatically reset after read-out.
utMbus		typMbus	M-Bus parameter and receive buffer (see function block FbMbusMaster)
Feedback value:		Data type:	Comment:
utMBInfo		typMbusInfo	Fixed meter data (header in data telegram with ID, manufacturer, medium, ..)
.bPAdr		BYTE	Primary address read out
.stID		STRING(8)	Secondary address (ID)
.stMAN		STRING(3)	3-digit manufacturer code
.stMED		STRING(14)	Medium to be measured
.bMGen		BYTE	Version number (generation)
.bMStatus		BYTE	M-Bus status byte
.bAllRecords		BYTE	Number of data records in the telegram
.bGoodRecords		BYTE	Analysed data records
utEnergy		typMbusRecord	Meter reading of consumed energy (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		TypMbusUnit	Unit as constant (enumeration)
utPower		TypMbusRecord	Effective electrical power (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		TypMbusUnit	Unit as constant (enumeration)

bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...) 0x04 = no analysable records
--------	------	--

Graphical display:

Function description:

The function block **FbMbus_Electricity** is used to read out and decode M-Bus electricity meter data. It can only be used in connection with the **FbMbusMaster** M-Bus communication block.

Both function blocks are synchronized via the **"utMbus"** variable structure and must therefore be connected to each other.

The input variable **"dwAdr"** is preallocated with the primary (<256) or secondary (>256) address of the M-Bus electricity meter. By setting **"xSND_NKE"**, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequence telegrams can respond with their 1st (important) telegram at the next request.

The **"xStart"** input/output variable starts reading out the electricity meter. This variable is automatically reset after read-out.

"bError" indicates whether the read-out has been performed without error.

The effective energy consumption and effective power of the electricity meter are contained in the output variables **"utEnergy"** and **"utPower"** contain when the read-out has been performed successfully. Note that the power is not delivered by all M-Bus electricity meters in their telegram. In this case, the string variables remain empty. To convert the meter reading into both a REAL type value and a desired unit, the function block **FbUnitConverter** described further below can be connected downstream by the user.

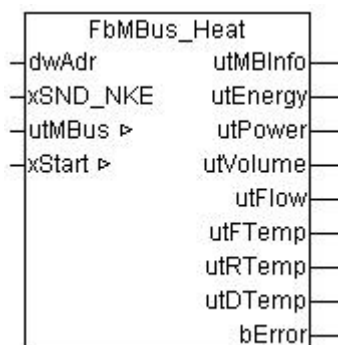
Important:

Only the meters with "Electricity" medium setting are analysed.

M-Bus Heat Meter (FbMBus_Heat)

WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMBus_Heat	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:		MBus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
dwAdr		DWORD	M-Bus primary address (>255 Sec.-Adr.)
xSND_NKE		BOOL	M-Bus normalisation command first
Input/output parameter:		Data type:	Comment:
utMBus		typMBus	M-Bus parameter and receive buffer (see function block FbMBusMaster)
xStart		BOOL	Toggle variable for starting to read out the heat meter. Automatically reset after readout.
Feedback value:		Data type:	Comment:
utMBInfo		typMBusInfo	Fixed meter data (header in data telegram with ID, manufacturer, medium, ..)
.bPAdr		BYTE	Primary address read out
.stID		STRING(8)	Secondary address (ID)
.stMAN		STRING(3)	3-digit manufacturer code
.stMED		STRING(14)	Medium to be measured
.bMGen		BYTE	Version number (generation)
.bMStatus		BYTE	M-Bus status byte
.bAllRecords		BYTE	Number of data records in the telegram
.bGoodRecords		BYTE	Analysed data records
utEnergy		typMBusRecord	Meter reading of consumed energy (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		typMBusUnit	Unit as constant (enumeration)
utPower		typMBusRecord	Effective heat power (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		typMBusUnit	Unit as constant (enumeration))

utVolume	typMBus Record	Measured water volume (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
utFlow	typMBus Record	Effective flow (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
utFTemp	typMBus Record	Flow temperature (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
utRTemp	typMBus Record	Return temperature (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
utDTemp	typMBus Record	Differential temperature (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...) 0x04 = no analysable records

Graphical display:


Function description:

The function block **FbMBus_Heat** is used to read out and decode M-Bus heat meter data. It can only be used in connection with the M-Bus communication block **FbMBusMaster**.

Both function blocks are synchronized via the variable structure **"utMBus"** and must therefore be connected to each other.

The input variable **"dwAdr"** is preallocated with the primary (<256) or secondary (>256) address of the M-Bus heat meter. By setting **"xSND_NKE"**, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequence telegrams can respond with their 1st (important) telegram at the next request.

The **"xStart"** input/output variable starts reading out the heat meter. This variable is automatically reset after read-out.

"bError" indicates whether the read-out has been performed without error.

The effective energy consumption and effective power of the heat meter are contained in the output variables **"utEnergy"** and **"utPower"** when the read-out has been performed successfully. The effective volume and effective flow are delivered by **"utVolume"** and **"utFlow"**. The measured flow as well as return and differential temperatures are represented by the variables **"utFTemp"**, **"utRTemp"** and **"utDTemp"**. Note that the performance is not delivered by all M-Bus heat meters in their telegram. In this case, the string variables remain empty. To convert the meter reading into both a REAL type value and a desired unit, the function block **FbUnitConverter** described further below can be connected downstream by the user.

Important:

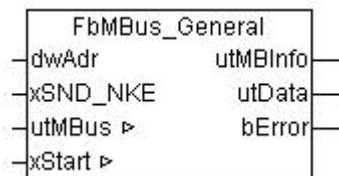
Only the meters with the following medium setting are analysed:

heat, heat inlet, heat outlet, heat/cooling, cooling inlet, cooling outlet,
hot water

Additional function

General M-Bus Meter (FbMBus_General)

WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMBus_General	
Type:		Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:		MBus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
dwAdr		DWORD	M-Bus primary address (>255 Sek.-Adr.)
xSND_NKE		BOOL	M-Bus normalisation command first
Input/output parameter:		Data type:	Comment:
utMBus		typMBus	M-Bus parameter and receive buffer (see function block FbMBusMaster)
xStart		BOOL	Toggle variable for starting to read out the M-Bus meter. Automatically reset after read-out.
Feedback value:		Data type:	Comment:
utMBInfo		typMBusInfo	Fixed meter data (header in data telegram with ID, manufacturer, medium, ..)
.bPAdr		BYTE	Primary address read out
.stID		STRING(8)	Secondary address (ID)
.stMAN		STRING(3)	3-digit manufacturer code
.stMED		STRING(14)	Medium to be measured
.bMGen		BYTE	Version number (generation)
.bMStatus		BYTE	M-Bus status byte
.bAllRecords		BYTE	Number of data records in the telegram
.bGoodRecords		BYTE	Analysed data records
utData		typMBus Record	Meter reading (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Enum_Unit		typMBusUnit	Unit as constant (enumeration)
bError		BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...) 0x04 = no analysable records

Graphical display:**Function description:**

The function block **FbMBus_General** is used to read out and decode the data of a M-Bus meter. It returns exactly one meter reading. The function block can only be used in connection with the M-Bus communication block **FbMBusMaster**.

Both function blocks are synchronized via the variable structure **"utMBus"** and must therefore be connected to each other.

The input variable **"dwAdr"** is preallocated with the primary (<256) or secondary (>256) address of the M-Bus meter. By setting **"xSND_NKE"**, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequence telegrams can respond with their 1st (important) telegram at the next request.

The **"xStart"** input/output variable starts reading out the meter. This variable is automatically reset after read-out.

"bError" indicates whether the read-out has been performed without error.

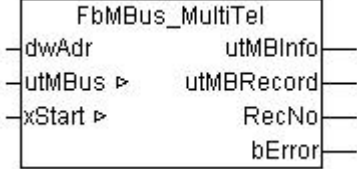
The consumption value of the M-Bus meter is contained in the output variable **"utData"** when the read-out has been performed successfully. The most important record is automatically determined by the program when several records are contained in the telegram. The importance order is specified as follow:

1. Energy value
2. Volume
3. HCA (Heat Cost Allocator, nondimensional)
4. Power
5. Volume flow
6. Temperature
7. Differential temperature
8. Pressure

To convert the meter reading into both a REAL type value and a desired unit, the function block **FbUnitConverter** described further below can be connected downstream by the user.

M-Bus Multi Telegram Meter (FbMbus_MultiTel)

WAGO-I/O-PRO Library Elements			
Category:		Building Automation	
Name:		FbMbus_MultiTel	
Type:		Funktion <input type="checkbox"/>	Funktionsblock <input checked="" type="checkbox"/> Programm <input type="checkbox"/>
Library name:		Mbus_03.lib	
Applicable to:		See release note	
Input parameter:		Data type:	Comment:
dwAdr		DWORD	M-Bus primary address (>255 Sec.-Adr.)
Input/output parameter:		Data type:	Comment:
xStart		BOOL	Toggle variable for starting to read out the M-Bus meter. Automatically reset after read-out.
utMbus		typMbus	M-Bus parameter and receive buffer (see function block FbMbusMaster)
Feedback value:		Data type:	Comment:
utMBInfo		typMbusInfo	Fixed meter data (header in data telegram with ID, manufacturer, medium, ..)
.bPAdr		BYTE	Primary address read out
.stID		STRING(8)	Secondary address (ID)
.stMAN		STRING(3)	3-digit manufacturer code
.stMED		STRING(14)	Medium to be measured
.bMGen		BYTE	Version number (generation)
.bMStatus		BYTE	M-Bus status byte
.bAllRecords		BYTE	Number of data records in the telegram
.bGoodRecords		BYTE	Analysed data records
utMBRecord		ARRAY[1..20] OF typMbusRecord	Byte array with meter reading (value, power of 10, unit)
.Value		STRING(21)	Value as string
.Pof10		STRING(3)	Power of 10 as string
.Unit		STRING(7)	Unit as string
.Descriptor		STRING(6)	Description (e.g.. L1 for phase 1)
.Enum_Unit		typMbusUnit	Unit as constant (enumeration)
RecNo		BYTE	Number of entries

bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...) 0x04 = no analysable records
Graphical display:		
		
Function description:		
<p>The function block FbMBus_MultiTel is used to read out and decode the data of a M-Bus multi telegram meter (e.g.. electricity meter DELTA+ Serie of ABB DAM13000, DZ4000 and Berg BAM13000 The function block can only be used in connection with the M-Bus communication block FbMBusMaster.</p> <p>Both function blocks are synchronized via the variable structure „utMBus“ and must therefore be connected to each other.</p> <p>The input variable „dwAdr“ is preallocated with the primary (<256) or secondary (>256) address of the M-Bus meter.</p> <p>By setting „xSND_NKE“, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequence telegrams can respond with their 1st (important) telegram at the next request.</p> <p>The „xStart“ input/output variable starts reading out the meter. This variable is automatically reset after read-out.</p> <p>„bError“ indicates whether the read-out has been performed without error.</p> <p>The output variable „utMBRecord“ is an array of type „typMBusRecord“. It contains a maximum entry of 20 M-Bus data sets.</p> <p>„RecNo“ indicates the number of decoded data sets.</p> <p>Die Variablen „utMBRecord.Descriptor“ might provides an additional description for the individual M-Bus data set. Possible discriptions are:</p> <p>T1, T2, ... tariff</p> <p>L1, L2, L3 Phases of electrical network</p> <p>LF power factor</p> <p>FR Frequency</p> <p>To convert the meter reading into both a REAL type value and a desired unit, the function block FbUnitConverter described further below can be connected downstream by the user.</p> <p>It is possible to read meter values of any medium types.</p>		

M-Bus RawDevice (FbMbus_RawDevice)

WAGO-I/O-PRO Library Elements		
Category:	Building Automation	
Name:	FbMbus_RawDevice	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Mbus_03.lib	
Applicable to:	See release note	
Input parameter:	Data type:	Comment:
dwAdr	DWORD	M-Bus primary address
xSND_NKE	BOOL	M-Bus normalisation command first
Input/output parameter:	Data type:	Comment:
utMbus	typMbus	M-Bus parameter and receive buffer (see function block FbMbusMaster)
xStart	BOOL	Toggle variable for starting to read out the meter. Automatically reset after read-out.
Feedback value:	Data type:	Comment:
MbusBuffer	typMbus Buffer	Byte array with M-Bus raw data
.aobData	ARRAY [0..261] OF BYTE	Data buffer
.bLength	BYTE	Occupied length in data buffer
bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...)
Graphical display:		
<div><div>FbMbus_RawDevice</div><div><div>dwAdr</div><div>xSND_NKE</div><div>utMbus ▸</div><div>xStart ▸</div></div><div><div>MbusBuffer</div><div>bError</div></div></div>		

Function description:

The function block **FbMbus_RawDevice** is used to read out a M-Bus meter and returns the raw data without analyse in a byte array. It can only be used in connection with the M-Bus communication block **FbMbusMaster**.

Both function blocks are synchronized via the variable structure **“utMbus”** and must therefore be connected to each other.

The input variable **“dwAdr”** is preallocated with the primary address of the M-Bus meter. By setting **“xSND_NKE”**, a M-Bus normalisation command is sent to the meter before the data is read out. As a result, meters with sequence telegrams can respond with their 1st (important) telegram at the next request.

The **“xStart”** input/output variable starts reading out the M-Bus meter. This variable is automatically reset after read-out.

“bError” indicates whether the read-out has been performed without error.

Both a byte array with M-Bus raw data and the length of this array are contained in the output variable **“MbusBuffer”** when the read-out has been performed successfully.

M-Bus Send data (FbMbusSend)

WAGO-I/O-PRO Library Elements		
Category:	Building Automation	
Name:	FbMbusSend	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	Mbus_03.lib	
Applicable to:	See release note	
Input parameter:	Data type:	Comment:
dwAdr	DWORD	M-Bus primary address
bCField	BYTE	M-Bus C-Feld (16#53 for send data)
xSND_NKE	BOOL	M-Bus normalisation command first
MbusBuffer	typMbus Buffer	Byte array with M-Bus raw data without frame (from CI-Feld)
.aobData	ARRAY [0..261] OF BYTE	Data buffer
.bLength	BYTE	Occupied length in data buffer
Input/output parameter:	Data type:	Comment:
utMbus	typMbus	M-Bus parameter and receive buffer (see function block FbMbusMaster)
xStart	BOOL	Toggle variable for starting to read out the meter. Automatically reset after read-out.
Feedback value:	Data type:	Comment:
bError	BYTE	Error code 0x00 = no error 0x01 = interface error (SERCOMM) 0x02 = M-Bus timeout 0x03 = M-Bus layer 1 error (CRC, ...)
Graphical display:		
<div><div>FbMbusSend</div><div><div><div>dwAdr</div><div>bCField</div><div>xSND_NKE</div><div>MbusBuffer</div><div>utMbus ▶</div><div>xStart ▶</div></div><div>bError</div></div></div>		

Function description:

The function block **FbMbusSend** is used to transmit data to an M-Bus meter. It must be executed in conjunction with the **FbMbusMaster** communications module.

Synchronisation of several instances of the module or with other modules that request data is achieved by means of the variable structure **"utMbus"**, which is also used by the **FbMbusMaster** communications module.

The M-Bus data to be transmitted are made available in the **"MbusBuffer"** input variables from the CI field up to the last data byte. A data record to be sent appears as follows (68 = start character, L = length byte, Dat1, ..., Datn = data, CRC = checksum, 16 = stop character):

68 L L 68 C-Field PAdr CI-Field Dat1 Dat2 ... Datn CRC 16 (numbers hexadecimal).

"MbusBuffer" is determined by the program together with (CI-Field, Dat1, Dat2, ..., Datn) and the buffer length n+1 before transmission. **FbMbusSend** automatically inserts start character, length, CRC and stop character. The meter addressed by means of **"dwAdr"** acknowledges the received data record with the individual character 16#E5. If all M-Bus meters on the bus are to receive the data record, then the broadcast address 255 can be used without acknowledgement. In this case, the **FbMbusMaster** communications module automatically ensures that no telegrams are repeated and that there is therefore no unnecessary waiting time due to timeouts.

"bCField" is the C-Field of the M-Bus telegram to be transmitted. The C-Field has the value 16#53, for example, for "send data to slave" (or 16#73 when FCB is set). 16#52 (or 16#72 when FCB is set) is used for selecting a meter with the help of its 8-digit ID (secondary address) on the primary address 253. The M-Bus CI-Field is the first byte of the **"MbusBuffer"** input variables. The CI-Field 16#51 signals to the addressed meter that data is to be sent to it from the master. With CI-Field 16#50, the addressed meter executes the "Application Reset" command.

When **"xSND_NKE"** is set, an additional M-Bus normalisation command is sent to the addressed meter before the actual data are sent.

The input/output variable **"xStart"** starts the transmission of data to the M-Bus meter. This variable is automatically reset when transmission is complete.

"bError" indicates whether the read-out has been performed without error.

Examples:
1st practical example:

A Sensus COSMOS hybrid water meter only has the consumption and a few historical values in its telegram and not the instantaneous flow. In order to output consumption and flow at the same time in one telegram, it requires two set-up telegrams from the M-Bus master:

68 07 07 68 53 02 51 0F 07 55 4D 5E 16 switches the meter with address 2 to mixed protocol (hexadecimal telegram mode)

68 07 07 68 53 02 51 0F 07 55 4D 5E 16 selects consumption and flow for the meter with address 2 (hexadecimal telegram mode)

Two constants of type "typMbusBuffer" can be created in the program, which are transmitted consecutively to the address of the meter by two instances of the function block "*FbMbusSend*". If there are several COSMOS hybrid water meters on the M-Bus network, then the telegram can be sent to the broadcast address 255 (= primary address 255). In doing so, it must be noted that, in this case, none of the meters acknowledges the received telegram with the individual character 16#E5. **FbMbusSend** gives a timeout error, which in this case is of no significance however.

VAR CONSTANT

StoMP_Hybrid: typMbusBuffer := (aobData:=16#51,16#0F,16#07, 16#55,16#4D, 251(0),bLength:=5);

MProt_Hybrid: typMbusBuffer := (aobData:=16#51,16#0F,16#07,16#4D,16#3A, 16#6A, 250(0),bLength:=6);

END_VAR

The meter can then be read with the function model "*FbMbus_Water*". When doing so, the SND_NKE option must be deactivated, as otherwise the meter will fall back on its standard telegram without the instantaneous flow.

2nd practical example:

An "Application Reset" command is to be output to a meter. In rare cases, this command may be necessary to ensure that the meter replies to the next query with its standard telegram.

68 03 03 68 53 02 50 A5 16 the meter with address 2 receives an "Application Reset" command (hexadecimal telegram mode)

VAR CONSTANT

AppRes: typMbusBuffer := (aobData:=80,255(0),bLength:=1);

END_VAR

The constant "AppRes" is passed to the **FbMbusSend** function block for transmission.

Unit Converter

M-Bus Unit Converter (FbUnitConverter)

WAGO-I/O-PRO Library Elements		
Category:	Building Automation	
Name:	FbUnitConverter	
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/> Program <input type="checkbox"/>
Library name:	MBus_03.lib	
Applicable to:	See release note	
Input parameter:	Data type:	Comment:
UtMBRecord	TypMBusRecord	M-Bus data records (value, power of 10, unit)
.Value	STRING(21)	Value as string
.Pof10	STRING(3)	Power of 10 as string
.Unit	STRING(7)	Unit as string
.Enum_Unit	typMBusUnit	Unit as constant (enumeration)
UtUnit typMBusUnit Target unit desired		
Feedback value:	Data type:	Comment:
RValue	REAL	Value of target unit "utUnit"
XError	BOOL	Error when converting to the target unit
Graphical display:		
<div><div>FbUnitConverter</div><div><div>utMBRecord</div><div>rValue</div><div>utUnit</div><div>xError</div></div></div>		

Function description:

The function block **FbUnitConverter** is used to convert a M-Bus record **"utMBRecord"** into both a desired target unit **"utUnit"** and a REAL type value **"rValue"**. **"xError"** indicates whether the conversion into the target unit has been performed without error.

The unit converter can be connected downstream to the data outputs of the function blocks "FbMBus_Water", "FbMBus_Electricity", "FbMBus_Heat" and "FbMBus_General".

The variable **"utUnit"** can assume the following values:

TYPE typMBusUnit :

```
(miW, W, kW, MW, GW,
miJps, Jps, kJps, MJps, GJps,
miJpmin, Jpmin, kJpmin, MJpmin, GJpmin,
miJph, Jph, kJph, MJph, GJph,
miWs, Ws, kW, MWs, GWs,
miJ, J, kJ, MJ, GJ,
miWmin, Wmin, kWmin, MWmin, GWmin,
miWh, Wh, kWh, MWh, GWh,
mil, l, m3,
milps, lps, m3ps,
milpmin, lpmin, m3pmin,
milph, lph, m3ph,
g, kg,
gps, kgps,
gpmin, kgpmin,
gph, kgph,
mibar, bar, kbar,
miC, C,
miK, K,
HCA,
V
mA, A
none);
```

END_TYPE

mi = milli, k = kilo, M = Mega, G = Giga

ps = per second, pmin = per minute, ph = per hour

l = Litre, m3 = m³, W = Watt, J = Joule

HCA = Heat Cost Allocator (without unit)

None = invalid

Note:

Please note that the converted meter values may have rounding errors due to the REAL format. It is not possible to convert input values outside the range of $\pm 3\,999\,999\,999$

Addition

M-Bus Version

WAGO-I/O-PRO Library Elements			
Category:	Building Automation		
Name:	MBus_Version		
Type:	Function <input type="checkbox"/>	Function block <input checked="" type="checkbox"/>	Program <input type="checkbox"/>
Library name:	MBus_03.lib		
Applicable to:	See release note		
Input parameter:	Data type:	Comment:	
EN	BOOL	The function block is triggered by a positive going edge at this input	
Feedback value:	Data type:	Comment:	
MBUS_VERSION	WORD	Library version	
Graphical display:			
<div><div>MBUS_VERSION</div><div>—ENMBus_Version—</div></div>			
Function description:			
The function block MBUS_VERSION returns the current version number to the library. This function block can be used for information during the program development. Furthermore version conflicts can be prevented at runtime.			



WAGO Kontakttechnik GmbH & Co. KG
Postfach 2880 • D-32385 Minden
Hansastraße 27 • D-32423 Minden
Telefon: 05 71/8 87 – 0
Telefax: 05 71/8 87 – 1 69
E-Mail: info@wago.com

Internet: <http://www.wago.com>
