ekinex

CONTROL YOUR LIVING SPACE

Gateway configuration manual
M-Bus master - KNX TP
EK-BM1-TP-20
EK-BM1-TP-40
EK-BM1-TP-80
EK-BM1-TP-160

Indice

Scope of the document

This document describes the gateway (protocol converter) M-Bus master – KNX TP. The gateway finds its ideal applicaton in the integration of M-Bus devices over a M-Bus serial network in a KNX-based automation system for homes and buildings. This product belongs to a broad line of ekinex® gateways designed to meet the needs for integration of the building automation most widely used protocols, based on serial, Ethernet or proprietary infrastructures. For further informations about the available technical solutions, please visit www.ekinex.com.

1 Product description

The ekinex® M-Bus master EK-BM1-TP gateway is a KNX modular unit for panel mounting. It allows you to exchange informations with one or more slave devices over a serial network through M-Bus protocol. The ekinex gateway acts as M-Bus Master. The informations exchanged over the M-Bus network are updated over the KNX network by means of a twisted pair (TP) communication cable.

The device manages a two-way data stream: the M-Bus registers can be cyclically read and their value sent as a communication object over the KNX TP network through a multicast communication to configured group addresses. The data update over the KNX network can be done cyclically and/or on event of change of the data acquired by the M-Bus network.

Likewise, the ekinex gateway can make requests to cyclically readings KNX communication objects or acquire their values during data exchange over the bus. Cyclically or on event of change of the communication objects, data are written on the M-Bus registers of one or more configured devices.

The ekinex gateway supports the entire M-Bus master protocol. The maximum number of supported slave devices depends on the gateway model. It allows to link to KNX communication objects different measures and variables such as meters, energy totalizers, flow, power, absorbed current and so on.

These are the available gateway sizes and models:

EK-BM1-TP-20: up to 20 devices

EK-BM1-TP-40: up to 40 devices

EK-BM1-TP-80: up to 80 devices

EK-BM1-TP-160: up to 160 devices

As for KNX communication, 1-bit, 1-byte, 2-byte and 4-byte communication objects can be acquired: internal conversion functions allow you to convert the informations from and to 16-bit floating point values (DPT 9.xxx) starting from integer M-Bus registers.

Configuration is performed through a PC application software which communicates through the integrated Ethernet port. The application software CGEKBM1TP is available for download at www.ekinex.com.

1.1 Main functions

The gateway acts as a bidirectional protocol converter. Data streams are the following:

- M-Bus network – Cyclical reading from one or more slaves. Refresh time starts from 1 s (configurable)
- KNX TP network – Sending of writing multicasting frames (APCI = write)[1] to configured group addresses. Data can be sent cyclically over the bus (configurable refresh time), on event of change of the data contained in the "M-Bus memory image", or both cyclically and on change. Internal conversion functions to the most common types of KNX Datapoints are present.
- KNX TP network – Multicasting frame listening from configured group addresses (with selectable filters on the area or network of interest) or cyclical sending of read request frames (APCI = read). The values of the acquired communication objects are stored in a 1440-byte volatile memory buffer ("KNX image memory"). This buffer is independent from the "M-Bus image memory" buffer.
- M-Bus network – Writing of registers to one or more slave devices. Registers can be sent cyclically over the serial network (configurable refresh time), on event of change of the data contained in the "KNX memory image", or both cyclically and on change.

1.2 Technical data

Characteristic	Value
Power supply	8…24 Vac 12…35 Vdc
Power Absorption	At 24 Vdc: 3,5 VA
Application area	dry indoor environment
Environmental conditions	• Operating temperature: - 40 ... + 85°C • Stock temperature: - 25 ... + 55°C • Transportation temperature: - 25 ... + 70°C • Relative humidity: 93% non-condensing
Programming elements	1 pushbutton and 1 LED (red) on the front
Display elements	4 status LEDs + 1 Ethernet connector LED
Configuration elements	1-way microswitches • Microswitch A: OFF normal mode; ON Boot mode
Safety class	II
Installation	35 mm DIN rail (according to EN 60529)
Protection degree	IP20
Dimensions (WxHxD)	82 x 75 x 35 mm
Ethernet interface (IEEE 802.3)	
Connector	RJ45, minimum cable category: 5E
M-Bus interface	
Communication port	M-Bus (Twisted pair), electrically isolated from power supply and KNX communication port
Baud rate	Configurable, from 300 to 38400 baud
KNX TP interface	
Communication port	KNX TP (twisted pair), 9600 baud, electrically isolated from power supply and RS485 communication port
Power supply	SELV 30 Vdc through bus KNX
Current absorption from bus	< 13 mA

[1] APCI = Application Layer Protocol Control Information. Information contained in the frame sent to the application layer of the receiving device. It is defined by the KNX standard.

1.3 Supply

The supply includes the device and terminal blocks to connect to the KNX bus. An instruction sheet is also supplied within the package.

1.4 System requirements for configuration software

Configuration and commissioning of the ekinex® gateway must be performed using the application program CGEKBM1TP, available for download at www.ekinex.com.

The PC where the application program is installed must meet the following requirements:

- Desktop o laptop PC with Ethernet IEEE 802.3 port.
- 32/64 bit operating system, Microsoft Windows® XP, 7, 8.0, 8.1 e 10.

.NET Framework 4.0 system library installation is required.

1.5 Certifications

Compliance with the European directives is certified by the CE symbol on the product label and on the documentation.

2 Switching, display and command elements

The device is equipped with a pushbutton and a KNX programming LED, with a status LED and terminal blocks for KNX and M-Bus network connection. A port for RJ45 connector for device configuration via Ethernet as well as one 1-way microswitch are also present.

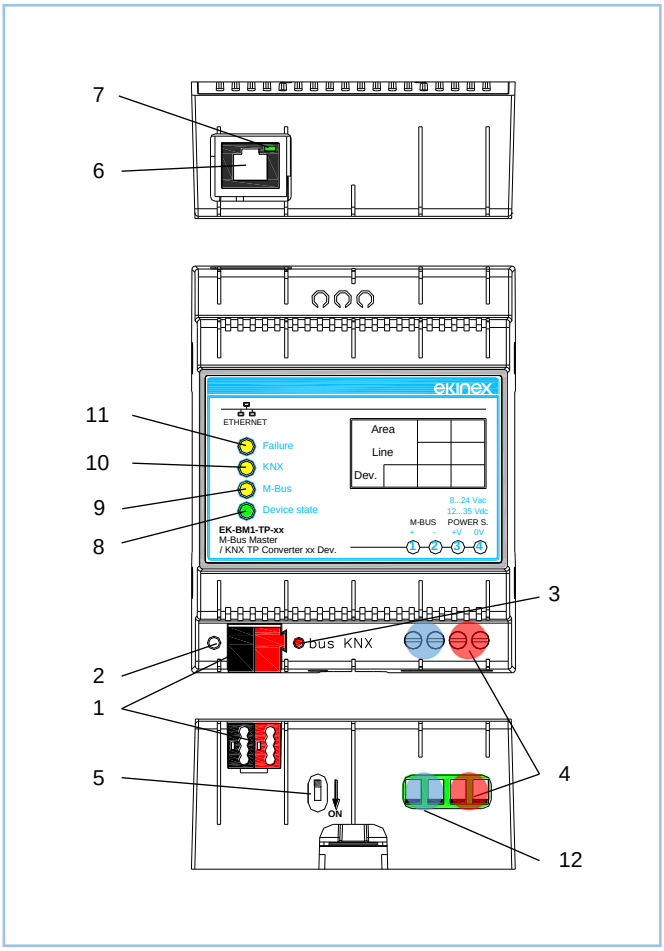


Figure 1 - Switching, display and connection elements

1)	KNX bus line terminal blocks
2)	KNX programming pushbutton
3)	KNX programming LED
4)	Power supply terminal blocks (3-4)
5)	1-way microswitch A
6)	Ethernet port
7)	Ethernet port LED
8)	Device satus LED
9)	M-Bus communication LED
10)	KNX communication LED
11)	Device error LED
12)	M-Bus serial line terminal blocks (1-2)

Command elements

- Pushbutton that switches between normal mode and KNX physical address programming.

1-way microswitches

- A - OFF: normal mode active. ON: Boot mode active

Display elements

The device can run according to two operating modes: Normal mode (configuration loaded, M-Bus and KNX communication running) and Boot mode (no configuration or still loading configuration)

LED	Normal mode	Boot mode
Green LED (8) – Device status	Slow blinking (~1 Hz)	**ON**: device on **OFF**: device off
Yellow LED (9) – M-Bus communication	Blinks when a frame is received on the M-Bus port.	**Fast blinking**: no configuration **Very slow blinking (~0,5 Hz)**: loading configuration.
Yellow LED (10) – KNX communication	Blinks when a frame is received.	**Fast blinking**: no configuration **Very slow blinking (~0,5 Hz)**: loading configuration.
Yellow LED (11) – Device error	**ON**: at least one M-Bus request did not get a correct answer **OFF**: no error	**Fast blinking**: no configuration **Very slow blinking (~0,5 Hz)**: loading configuration.
Green LED (7) – Ethernet port	**ON**: Ethernet connector plugged **OFF**: Ethernet connector unplugged	**ON**: Ethernet connector plugged **OFF**: Ethernet connector unplugged
Red LED (3) – KNX programming	**ON**: physical address programming mode on **OFF**: physical address programming mode off	**Fast blinking**: no configuration **Very slow blinking (~0,5 Hz)**: loading configuration.

In the current version of the device, both KNX physical address programming and configuration download must be performed through the configuration program: for KNX physical address please refer to “Communication parameters” paragraph, “ID Device” parameter.

3 Configuration and commissioning

The device configuration requires the following tools:

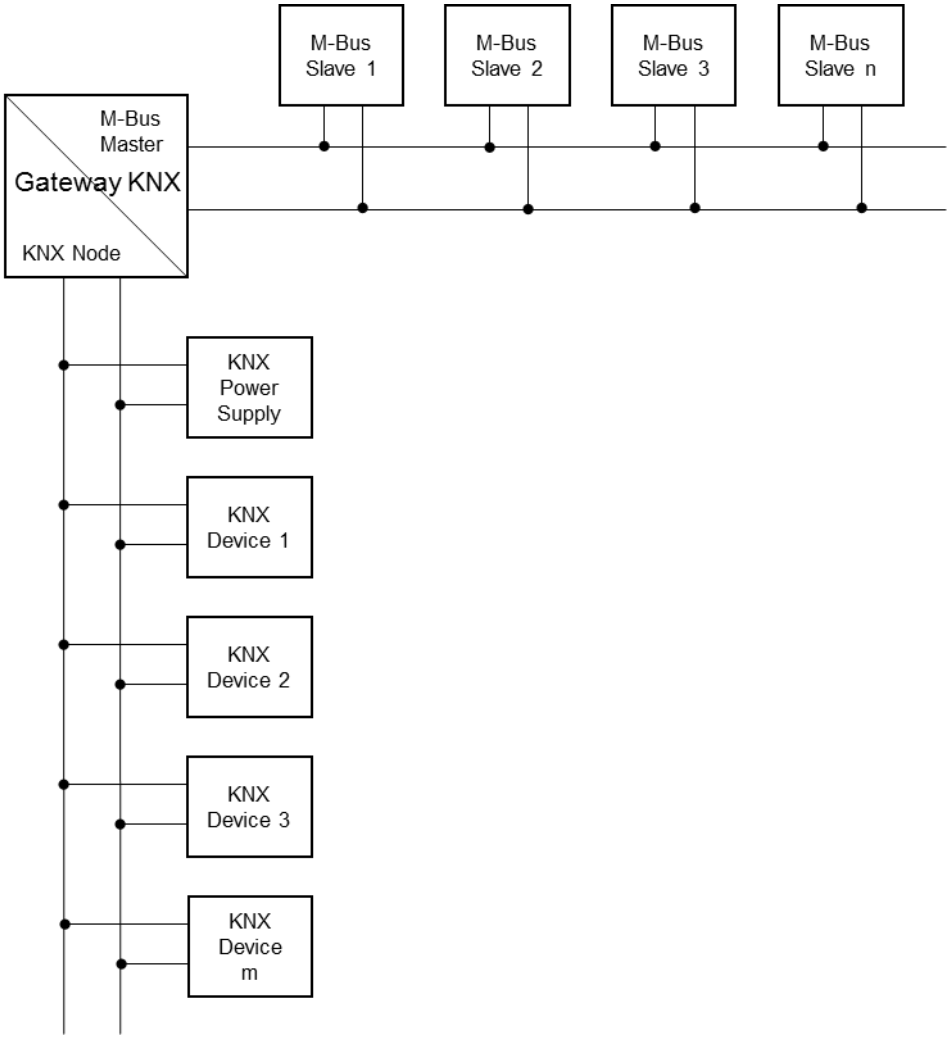
- The documentation of the M-Bus products, specifically the database of each product to be integrated, containing the addresses of the registers and physical parameters of the M-Bus serial communication (baud rate, parity check, delays, physical addresses of the devices to be integrated).
- CGEKBM1TP application software to properly configure the gateway.
- Knowledge of the ETS automation project, with particular attention to communication objects and group addresses passing on the bus during the multicast communication between sensors and actuators.

Configuration and commissioning of the ekinex® gateway require specialized skills about KNX networks and knowledge of the specific ETS automation project. In order to acquire such skills, it is essential to attend trainings and workshops organized at KNX-certified training centers. For further information: www.knx.it.

4 Generalità sul protocollo M-Bus

M-Bus, in its wired version, is a two-wire, low-cost bus system designed to read consumption meters for heat, water, gas and electricity. Protocol physical levels and basic elements are defined by EN1434-3 European standard. The KNX gateway acts as a Master of the wired M-Bus communication. The maximum number of supported slave devices depends on the gateway model (please refer to the chapter related to the available product codes). It allows to link to KNX communication objects different measures and variables such as meters, energy totalizers, flow, power, absorbed current and so on.

M-Bus system was developed to meet some important communication requirements for applications where billing of consumptions towards the end user is necessary. In that case it is necessary to connect a high number of meters (up to 250 devices per dorsal) distributed on high distances and with a high integrity level in data transmission. It is not necessary to achieve high transmission speeds since usually there is a low number of information to be transmitted. A further requirement is that the system must be low-cost: the two-wired version does not need a shielding sheath and all slave devices have their communication logic directly powered by the bus.



4.1 The physical layer

M-Bus is a hierarchical system where the communication is controlled by a master. A M-Bus system is made of a single master, slaves and a 2-wire connection cable (in the wired version supported by the KNX gateway). All slaves are connected in parallel.

Communication between the master and the slaves is a serial data transmission. To allow to remotely connect the slaves, the binary informations are represented as follows:

⇒ The bit transfer from the master to the slave is carried out by modulating a voltage level. Logical level “1” equals a nominal voltage of +36V; when logical level “0” is sent, the driver reduces the bus voltage by 12V to a nominal value of +24V.

⇒ The bit transfer from the slave to the master is carried out by modulating a current absorbtion by the slave. Logical level “1” equals a constant current value less than 1.5mA; transmission of the logical level “0” corresponds to an added current absorbtion of 11-20mA.

The only difference between different codes of KNX gateway is the power that the internal power supply is able to provide on the M-Bus, thus changing the number of devices that the gateway is able to support. The number of supported devices takes into account the number of slaves that can simultanously transit a “0” logical level during a collision.

The KNX gateway settings of the M-Bus serial communication port which are suitable for most factory-configured meters are the following: 2400 baud, 8E1 (8 data bits, even parity check, 1 stop bit).

4.2 Device addressing

The KNX gateway suopports 2 M-Bus slave addressing modes:

- By primary address (*Primary ID Node* in *M-Bus* form, section *Nodes* of the configuration program), with values in range 0-250. This kind of addressing reqires that the primary address is programmed inside the meter through an external application program, usually suplied by the device manufacturer.
- By secondary address (*Secondary ID Node* in *M-Bus* form, section *Nodes* of the configuration program). The secondary address is a 8-digit numeric code, usually printed on the front or accessible inside the display’s menu, if available.

In case of substitution of a meter over the M-Bus network with an equivalent product, in order for the KNX gateway to properly work, a system update is required, depending on the addressing type used:

- Primary addressing: the previously used address must be reprogrammed;
- Secondary address: the gateway must be reconfigured according to the new secondary address.

4.3 Data mapping

Mapping the M-Bus variables in the configuration software requires knowledge about the communication protocol defined by EN1434-3 european standard. Particularly, one topic to properly understand is the *long frame* structure containing the data to be transfered to the KNX-based automation system by the slave meters upon master’s request.

Start 68H	L Field	L Field	Start 68H	C Field	A Field	C1 Field	User Data (0-252 Byte)	Check Sum	Stop 16H

Table 1 - Long Frame structure

Within the *User Data* field, informations are enveloped in a *Fixed Data Header* and in a sequence of bytes (*Variable Data Blocks*) containing the real data to be exported as KNX communication objects.

Ident. Nr.	Manufact.	Version	Medium	Access No.	Status	Signature
4 Byte	2 Byte	1 Byte	1 Byte	1 Byte	1 Byte	2 Byte

Tabele 2 - Fixed Data Header

DIF	DIFE	VIF	VIFE	Data
1 Byte	0-10 Byte	1 Byte	0-10 Byte	0-N Byte
Data Information Block DIB		Value Information Block VIB		
Data Record Header				

Table 3 - Variable Data Block

Data within the frames are formatted with some fields indicating the reading “keys”. The DIF (Data Information Field) contains the function code (instant value, error value, max or min value), length and data type (e.g. 4-digit BCD). These informations are requested by the configuration program in the M-Bus form sections. The VIF (Value Information Field) contains the measurement unit (e.g.: kWh or °C) and the scale factor.

Value	Field	Data structure	Notes
0x68	Start Byte		
0xBD	L Field		
0xBD	L Field		
0x68	Start Byte		
0x08	C Field		
0x02	A Field		
0x72	C1 Field		
0x71	Identification Number (4° Byte)	Fixed Data Header	Corresponds to the secondary address. If in the *M-Bus* form the box *Identification Number* is checked, the secondary address is stored in a reserved support memory area.
0x65	Identification Number (3° Byte)		
0x45	Identification Number (2° Byte)		
0x28	Identification Number (1° Byte)		
0x4D	Manufacturer (2° Byte)		
0x6A	Manufacturer (1° Byte)		
0x81	Version		
0x04	Medium		
0x3E	Access Number		
0x27	Status		Corresponds to the meter status. If in the *M-Bus* form, section *Nodes*, the box *Node State* is checked, the device status is stored in a reserved support memory area.
0x00	Signature (2° Byte)		
0x00	Signature (1° Byte)		
0x04	DIF	Variable 1	The variable index must be used in the *Position* field of the section *Variables*, when *Variable List = by Position* is selected.
0x79	VIF - Identification		
0x00	Data (4° Byte)		
0x00	Data (3° Byte)		
0x00	Data (2° Byte)		
0x00	Data (1° Byte)		
0x04	DIF	Variable 2	
0x06	VIF - Energy		

Value	Field	Data structure	Notes
0x00	Data (4° Byte)		
0x00	Data (3° Byte)		
0x00	Data (2° Byte)		
0x00	Data (1° Byte)		
0x44	DIF	Variable 3	
0x06	VIF - Energia		
0x00	Data (4° Byte)		
0x00	Data (3° Byte)		
0x00	Data (2° Byte)		
0x00	Data (1° Byte)		
…		Other variables	
0x55	Check Sum		
0x16	Stop Byte		

Table 4 - Long Frame example

In the KNX gateway configuration program it is possible to insert the variable list which must be scquired by the meter according to 2 modes:

- By Type: it is necessary to insert all variables contained in the fram, filling up all the fields in the Variables section
- By Position: it is possible to map only the variables which are relevant for the applicatin by selecting the data's proper position index within the frame.

Mapping a meter's data is an activity that must be performed with extreme care. A track of the device to integrate in the KNX automation system is required.
For further information about M-Bus protocol please refer to the documentation on the website http://www.m-bus.com.

5 Configuration software

The ekinex® configuration software CGEKBM1TP allows you to perform the following operations:

- Selection of physical parameters of the M-Bus communication;
- Selection of physical address of the device over the KNX TP network;
- Selection of Ethernet parameters (for configuration download only);
- KNX network: communication objects definition and relative group addresses to be acquired;
- KNX network: communication objects definition and relative group addresses to be sent over the KNX network;
- M-Bus network: definition of the registers to be read from the network devices;
- M-Bus network: definition of the registers to be written on the network devices;
- Firmware and/or configuration update.

The application program consists in multiple modal windows called “forms”: each form must be closed before accessing the following form. The buttons on the main form (see Figure 2 – Main form of the application program) are ordered according to the proper sequence to follow in order to perform a correct configuration.

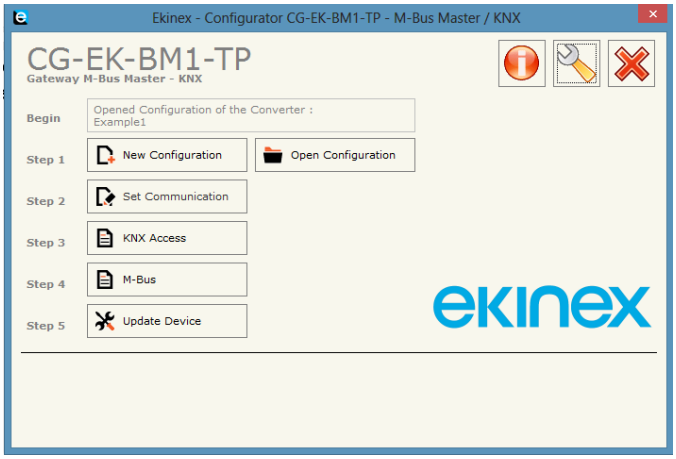


Figure 2 – Main form of the application program

Starting from the main form, by accessing the *About...* window, you can check the current version of the installed program.

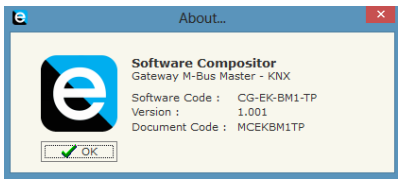


Figure 3 – About form

Please visit the section about communication gateways on www.ekinex.com in order to check the current version of the application program and download the latest version.

5.1 Creating a new project or modifying a saved project

The application program allows you to create a new configuration or open an existing one using the buttons called *New Configuration* and *Open Configuration* (see Figure 2 – Main form of the application program): the configuration files are stored on the hard drive in XML format.

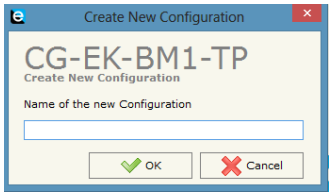


Figure 4 – Create new configuration form

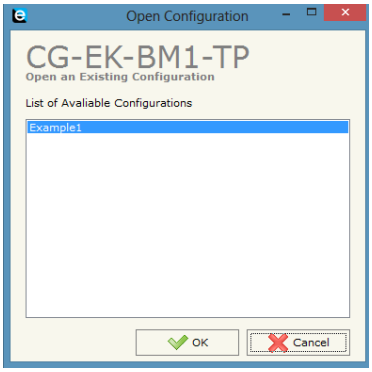


Figure 5 – Open configuration form

In order to duplicate an existing project, you must find the project folder containing the XML files and copy them in a new folder. Project files can be found by the following path:

"C:\Program Files(x86)\Ekinex\Compositor_CG-EK-BM1-TP\Projects".

Once the project has been duplicated, simply restart the application program and open the form *Open configuration* (see Figure 6 - Open configuration form): you will see the name of the duplicated project in the list of available configurations.

5.2 Software Options

The *Software Options* form allows you to select a different language for the application program.

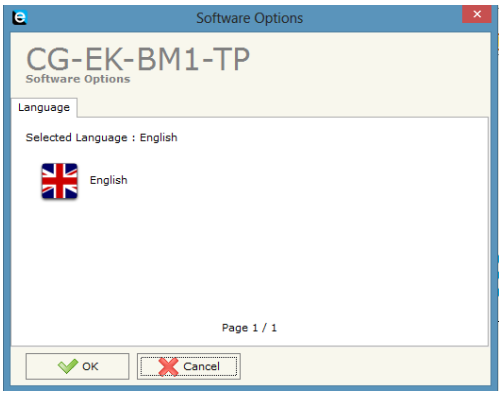


Figure 6 – Options form, Language tab

5.3 Communication parameters

In this section we define the basic communication parameters for the KNX TP network, for the M-Bus network and for Ethernet connection. Ethernet connection is required in order to perform the configuration update on the device.

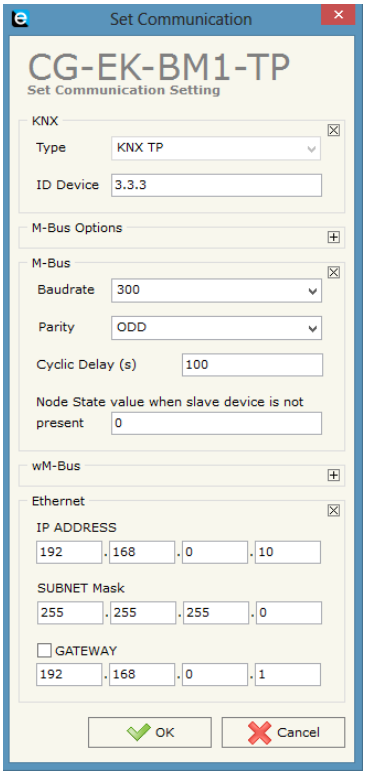


Figure 8 – Set communication form

You can access the form by pressing the *Set Communication* button in the main form (see Figure 2 – Main form of the application program).

Description of fields in *Set communication* form.

Parameter name	*Values*	*Description*
KNX		
Type	**KNX TP**	Type of connection used for KNX communication. The parameter has a constant value “KNX TP”. The device supports KNX communicaton over a twisted pair communication cable.
ID Device		This parameter identifies the physical address assigned to the KNX device. The format requires the use of a dot “.” as a separator beween the 3 fields: area, line and device address. Here are the conventions used for physical addressing and the values used for each field: Area field: = 0 reserved for backbone, values [1...15] Line field: = 0 reserved for main line, values [1...15] Device address field: = 0 reserved for coupler, values [1...255], range [1..64] for devices belonging to the line, above 64 for device belonging to extensions or other segments of the line. Example: 1.3.5: Area = 1; Line = 3; Device address = 5.
M-Bus Options		
M-Bus on wire	Not Present **Present**	Specifies if the wired version of M-Bus is present.

Parameter name	*Values*	*Description*
M-Bus wireless	**Not Present** Present	Specifies if the wireless version of M-Bus is present (not used)
M-Bus		
Baudrate	300 600 1200 2400 4800 **9600** 19200 38400	Baudrate of the M-Bus communication
Parity	**NONE** ODD EVEN	Parity check
Cyclic Delay (s)		Minimum delay between requests (in milliseconds) performed by the master.
Node state value when slave device is not present		It is possible to specify a value to be inserted in the “Node state” field when te gateway dos not find the polled M-Bus slave
wM-Bus (non usato)		
Mode	**S1** T1 N1 N2	Wireless communication mode (S1 or T1 for 868 MHz version and N1 or N2 for 169 MHz version)
Radio channel	**1a @ 4800 bps** 1b @ 4800 bps 2a @ 2400 bps 2b @ 2400 bps 3a @ 4800 bps 3b @ 4800 bps 0 @ 19200 bps 2a @ 4800 bps 2b @ 4800 bps	Radio Channel for wM-Bus communication (only 169 MHz version)
Ethernet		
IP ADDRESS		IP Address (4-octet format) assigned to the device. Each octet is set in an Edit box. Default IP Address is: **192.168.2.205**. This is the address assigned to the device before the first configuration or after a complete restore.
SUBNET Mask		Subnet mask assigned to the device.
GATEWAY		Gateway address used for Ethernet communication. The gateway can be enabled or disabled through the control check-box placed at the right side of the field.

Please refer to the technical documentation of the slave device in order to set the correct parameters of the serial communication. Incompatible values of these parameters may prevent the correct exchange of frames.

5.4 KNX communication object configuration

In this section we define communication objects sent or acquired over the KNX network. You can access the form by pressing the *KNX Access* button in the main form (see Figure 2 – Application program main form).

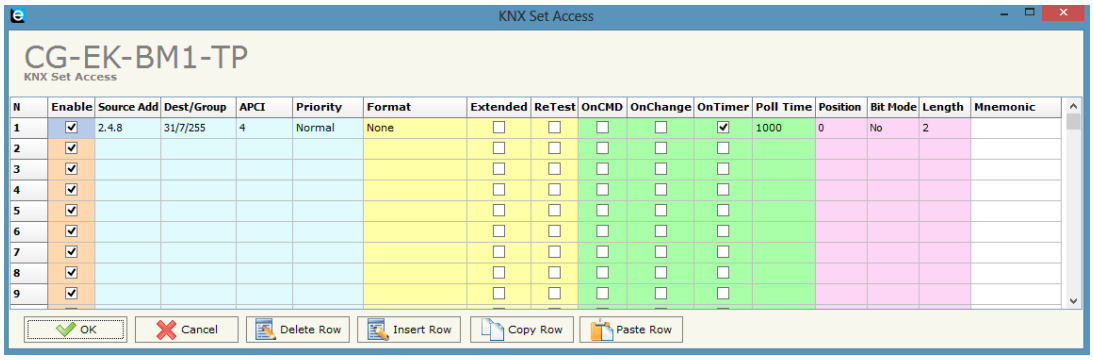


Figure 9 – KNX Set Access form

The form contains a configurable grid. Each record allows you to assign the properties for each communication object exchanged over the KNX network. In order to make the management of a significant number of data easier, after selecting a record it is possible to delete it from the project, insert a new record in a specific position and perform copy/paste of a previously configured record.

Description of fields in *KNX Set Access* form

Field name	***Values***	***Description***
N		Progressive number of the configuration record
Enable	**checked** / unchecked	Configuration record enabling. If a record is disabled, the corresponding data points will not be acquired or changed over the KNX bus
Source Address		In case of writing frames (field APCI=write) the physical address may correspond to the physical address of the gateway (*Device ID* field in the *Set Communication* form), in the format Area.Line.Address (each field must be separated by a dot). In case of reading frames (field APCI=read), *Source Address* acts as a filter. Through this field you can acquire datapoints of all lines over the KNX bus (0.0.0 value) or you can select one specific line (e.g. 4.3.0) or a single device identified by a specific physical address (e.g. 4.3.1).
Dest/Group		A Group Address (2-level, 3-level or free structure) or a Physical Address can be set. In case of a group address the fields must be separated through a "/", while in case of physical address the separator will be a ".".
APCI	read / write	The "read" option is used to send a request in order to read a communication object over the KNX bus. The "write" option must be selected if you want to change the value of a communication object over the KNX bus. Other services can be configured by editing the value of the corresponding service. The name used in the field refers to a 4-bit code (APCI = Application Layer Protocol Control Information) which defines the type of service required in KNX communication standard.
Priority	System/ Urgent / Normal / Low	KNX frames priority. In multicast communication (exchange of frames from/to group addresses), the default priority is Low.
Format	None / Swap16 / Swap32 / Swap All / Int to Float / Float to Int / Float 16 to Float 32	In case of a frame containing a data (in response to a reading request frame APCI = read), the Format field determines the data type conversion from the received frame to the support internal memory area. In case of a writing frame (APCI = write), the Format field determines the data type conversion from the support internal memory area to the frame.
Extended	**checked** / unchecked	Enables extended frame format for KNX communication (cEMI = Common Extended Message Interface)

Field name	*Values*	*Description*
ReTest	**checked** / unchecked	Enables the re-send of a frame in case of wrong response message
OnCMD	**checked** / unchecked	Not used
OnChange	**checked** / unchecked	Event which enables the automatic sending of command frames over the KNX bus when the data on the Modbus device changes their values.
OnTimer	**checked** / unchecked	Event which enables the cyclical sending of command frames over the KNX bus.
Poll Time		Cyclic poll time (in ms) when OnTimer event is enabled.
Position	Value in range [0…1439]	Position of the first byte where a data is stored, in the internal support memory buffer. In case of a record where APCI=read, *Position* refers to the "KNX image" buffer; in case of APCI=write, *Position* refers to the "M-Bus image" buffer. Please refer to the paragraph concerning the structure of the memory image to perform a correct addressing and avoid overlaps between the two data buffers.
Bit Mode	No / 0 / 1 / 2 / 3 / 4 / 5 / 6 / 7	Position, inside the first byte of the internal support memory buffer, where a 1-bit data is stored.
Length		Size (in number of bytes) of the data stored inside the internal memory.
Mnemonic		Text to comment the record and/or the datapoint over the KNX bus.

If *OnChange* field is selected, *OnTimer* in selected and *Poll Time* ≠ 0, the gateway will send the commands both cyclically and on change of the data acquired over the M-Bus network.

If *OnChange* and *OnTimer* fields are not selected, the gateway will only store the communication objects exchanged through the multicast frames over the KNX network ("sniffer" function).

Conversion types of internal data selectable through *Format* field:

Conversion	APCI = read	APCI = write
None	The value of the communication object is transferred in raw mode to the "KNX image" buffer and sent as register to the M-Bus network.	The value of the communication object acquired over the M-Bus network and stored in the "M-Bus image" buffer is transferred in raw mode as communication object over the KNX network.
Swap16	16-bit swap inside the stored data	16-bit swap inside the stored data
Swap32	32-bit swap inside the stored data	32-bit swap inside the stored data
Swap All	All bit swap inside the stored data	All bit swap inside the stored data
Int to Float		The integer value acquired over the M-Bus network is converted to a 2-byte (DPT 9.xxx) floating point value in order to be sent as communication object over the KNX network.
Float to Int	The 2-byte (DPT 9.xxx) floating point communication object value acquired over the M-Bus network is converted to integer in order to be sent as Holding register over the M-Bus network.	
Float 16 to Float 32	The 2-byte (DPT 9.xxx) floating point communication object value acquired over the M-Bus network is converted to a 32-bit floating point value (according to standard IEEE 754) in order to be sent as double Holding register over the M-Bus network.	

5.5 M-Bus registers configuration

In this section we define the registers read or written over the M-Bus network. You can access the form *M-Bus Network* by pressing the *M-Bus* button in the main form (see Figure 2 – Application program main form).

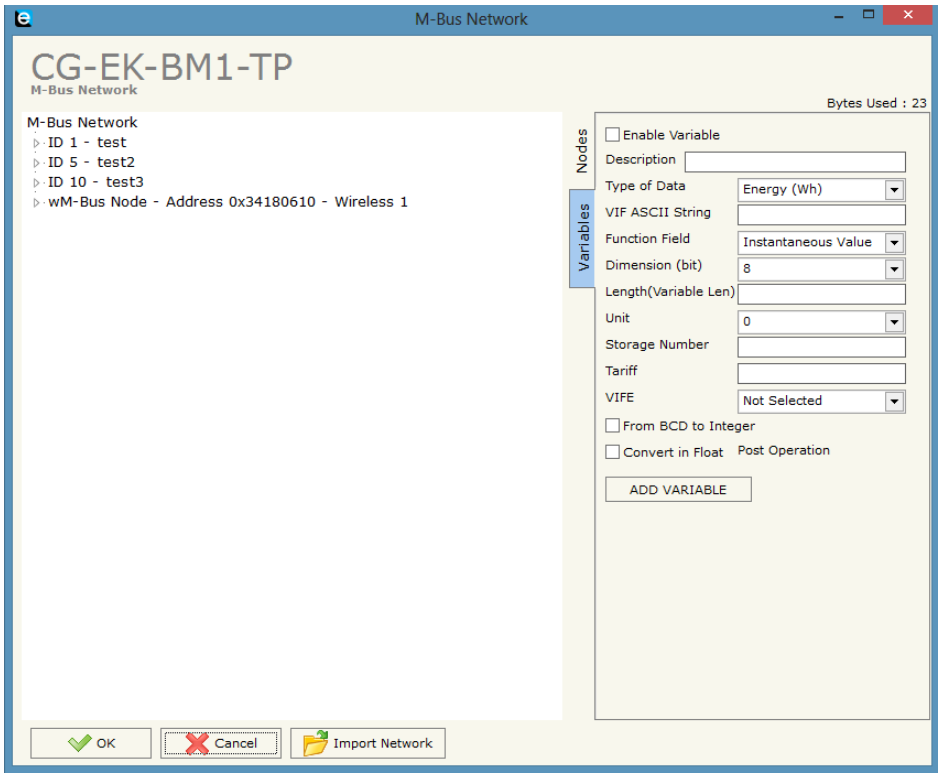


Figure 10a - M-Bus Network form – Nodes when M-Bus = M-Bus on wires

In the *Nodes* section it is possible to create the nodes of the M-Bus network or modify existing ones.

Description of fields in *M-Bus Network – Nodes* form when *M-Bus Type = M-Bus on wires:*

Field name	***Values***	***Description***
Enable Node	**checked** / unchecked	Enable of the node on the M-Bus network. If this option is disabled, the node will not be acquired by the gateway.
Description		Description of slave M-Bus node
M-Bus Type	**M-Bus on wires** wM-Bus	Slave M-Bus node type (wired or wireless)
Primary ID Node	Value in range [1…250]	Primary ID of the M-Bus slave node
Secondary ID Node	Value in range [0…99999999]	Secondary ID of the M-Bus slave node
Node State	checked / **unchecked**	If this field is checked, the gateway reserves one byte at the start of internal data array and saves the status of the counter.
Identification Number		If this field is checked, the gateway reserves four bytes at the start of internal data array and saves the Secondary Address of the device.
Swap identification number	**None** Type 1	Defines the identification number swapping mode Type 1 = ABCD -> CDAB
Convert BCD in Integer Identification Number	checked / **unchecked**	If this field is checked, the identfication number will be converted from BCD to Integer
Send SND_NKE	checked / **unchecked**	If this field is checked, then the communication starts, the gateway will send the SND_NKE frame
Send Reset App.	checked / **unchecked**	If this field is checked, the gateway sends the *Application Reset* command to the slave

Field name	Values	Description
Variables list	**By Type** By Position	Through this field it is possible to decide which definition to use to declare variables. Choosing *By Type,* i twill be necessary to fill up all fields with proper values, while choosing *By Position* i twill be possible to insert a progressive number corresponding to the desired variable.
Cut after	Value in range [1…250]	Through this field it is possible to decide after how many frames cut out the data request to the slave. It is used when the slave has too many data and not all of them are important to read.

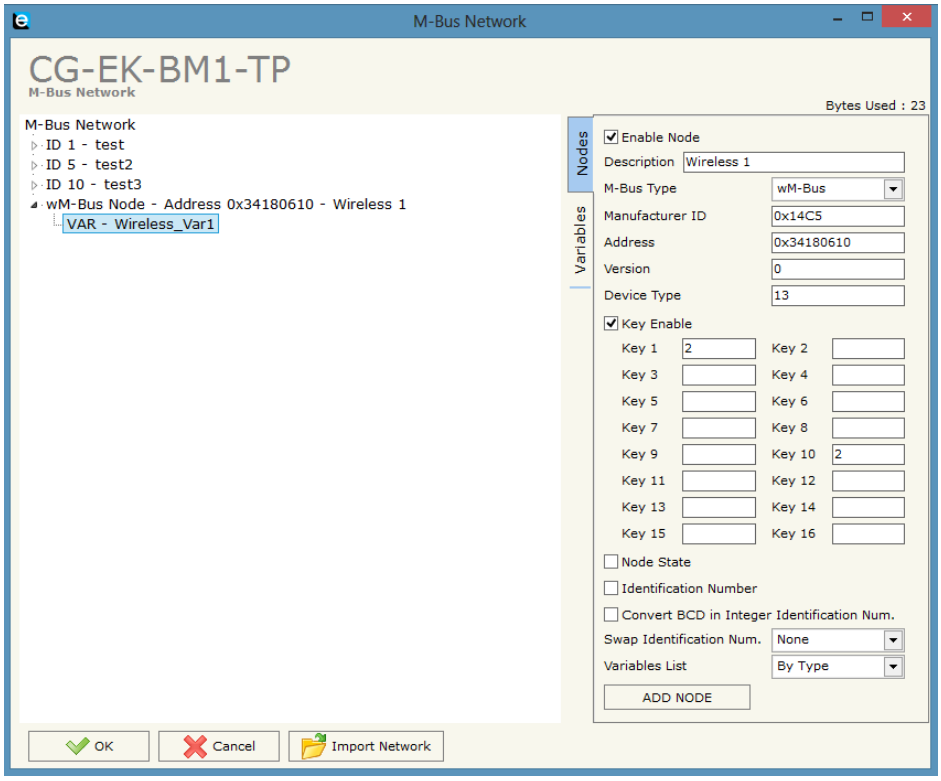


Figure 10b - M-Bus Network Form – Nodes when M-Bus = wM-Bus

Description of fields in *M-Bus Network – Nodes* form when *M-Bus Type = wM-Bus:*

Field name	Values	Description
Enable Node	**checked** / unchecked	Enable of the node on the M-Bus network. If this option is disabled, the node will not be acquired by the gateway.
Description		Description of slave M-Bus node
M-Bus Type	**M-Bus on wires** wM-Bus	Slave M-Bus node type (wired or wireless)
Manufacturer ID		ID of the manufacturer of the wM-Bus slave
Address		wM-Bus slave node ID
Version		Version of the wM-Bus slave
Device Type		Type of the wM-Bus slave
Key Enable		This field is used to decode the M-Bus frame sent by the wM-Bus slave node if an encrypted communication is used. In its 16 subfields the key for decodification is required.
Node State	checked / **unchecked**	If this field is checked, the gateway reserves one byte at the start of internal data array and saves the status of the counter.
Identification Number		If this field is checked, the gateway reserves four bytes at the start of internal data array and saves the Secondary Address of the device.

Field name	Values	Description
Convert BCD in Integer Identification Number	checked / **unchecked**	If this field is checked, the identfication number will be converted from BCD to Integer
Swap identification number	**None** Type 1	Defines the identification number swapping mode Type 1 = ABCD -> CDAB
Variables list	**By Type** By Position	Through this field it is possible to decide which definition to use to declare variables. Choosing *By Type,* i twill be necessary to fill up all fields with proper values, while choosing *By Position* i twill be possible to insert a progressive number corresponding to the desired variable.

After filling up the above fields, by pressing *Add Node* button the new node will appear on the left. To edit an existing node you have to select it from the table on the left, edit the desired properties then press *Modify Node* button.

After selecting the desired node, by pressing on *Variables* you can access the *Variables* section, which is used to create new variables or edit existing ones.

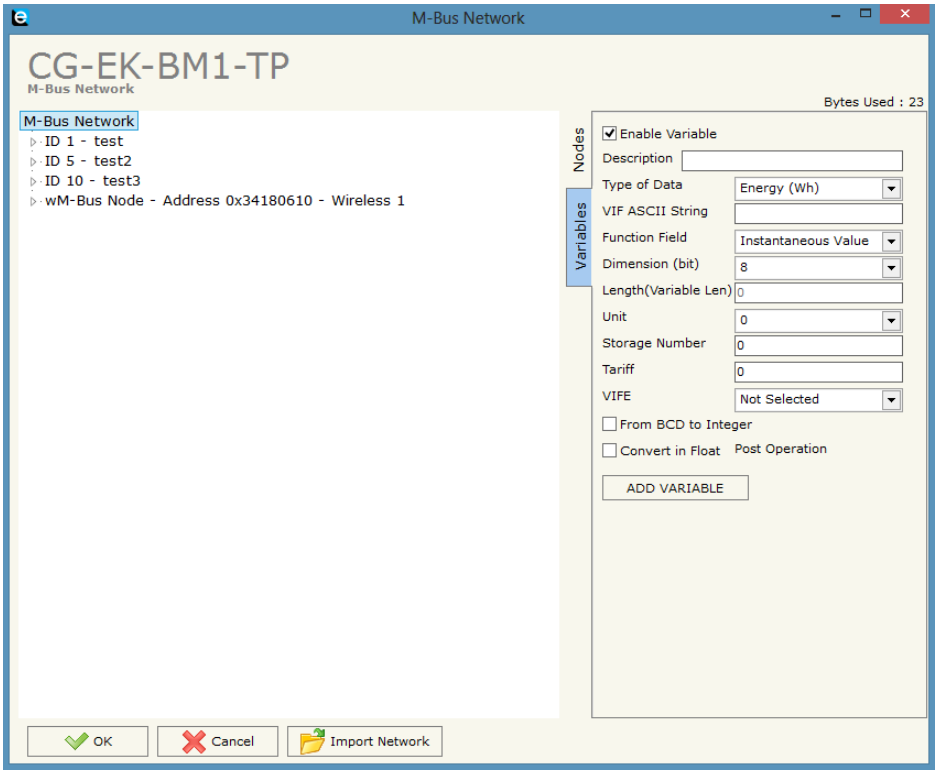


Figure 11a - M-Bus Network Form – Variables when Variables List = By Type

Description of fields in *M-Bus Network – Variables* form when *Variables List = By Type*

Field name	Values	Description
Enable Variable	**checked** / unchecked	In order to use the variable this field must be checked. If a variable is not used it is possible to dsable it without deleting it by unchecking this field.
Description		Variable description (optional)
Type of Data	**Energy (Wh)** Energy (J) Volume (m^3) Mass (Kg)	Variable measurement unit

Field name	*Values*	*Description*
	On Time Operating Time Power (W) Power (J/h) Volume Flow (m^3/h) Volume Flow Ext. (m^3/min) Volume Flow Ext. (m^3/s) Mass Flow (Kg/h) Flow Temperature (°C) Return Temperature (°C) Temperature Difference (K) External Temperature (°C) Pressure (bar) Averaging Duration Actuality Duration Type of Data in VIFE Time Point VIF is in ASCII Unit for H.C.A. Fabrication No (Enhanced) Identification Bus Address	
VIF ASCII String		This field is enabled only if *Type of Data = VIF is in ASCII*
Function Field	**Instantaneous Value** Minimum Value Maximum Value Vaue During Error State	Function of acquired value
Dimension (bit)	**8** 16 32 64 24 48 32 Real Variable Length	Data dimension (bit)
Length (Variable Len)		Data dimension (bit) if *Dimension (bit) = Variable Length*
Unit	0 1 2 3 4 5 6 7 8 9 10	If necessary, it is possible to assign a progressive number to the variable, to distinguish which slave it belongs to.
Storage Number		With this value the slave can transmit different values associated to meters or historical values in the same order they arrive.
Tariff		Indicates which device data is coming from.
VIFE		Data subtype, complementary to *Type of Data*
From BCD to Integer	checked / **unchecked**	If this field is checked, the identfication number will be converted from BCD to Integer
Convert in Float Post Operation	checked / **unchecked**	If this field is checked, the variable value will be converted to floating point

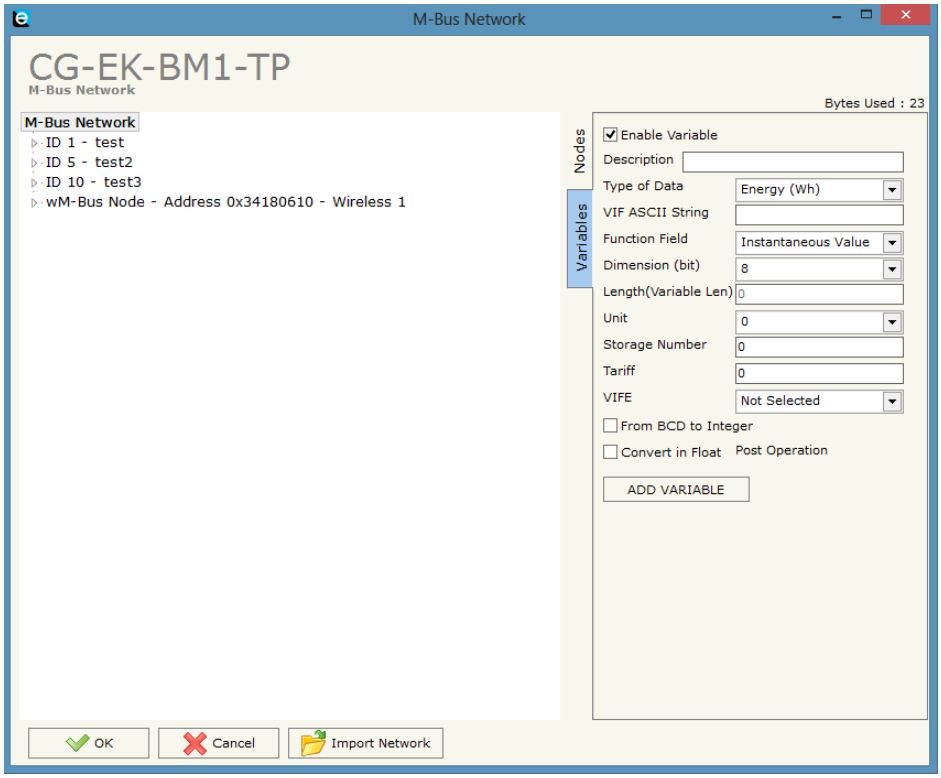


Figure 11b - M-Bus Network Form – Variables when Variables List = By Position

Description of fields in *M-Bus Network – Variables* form when *Variables List = By Position*

Field name	***Values***	***Description***
Enable Variable	**checked** / unchecked	In order to use the variable this field must be checked. If a variable is not used it is possible to dsable it without deleting it by unchecking this field.
Dimension (bit)	**8** 16 32 64 24 48 32 Real Variable Length	Data dimension (bit)
Length (Variable Len)		Data dimension (bit) if *Dimension (bit) = Variable Length*
From BCD to Integer	checked / **unchecked**	If this field is checked, the identfication number will be converted from BCD to Integer
Position		Nuber of the variable to be sent over the KNX network
Convert in Float	checked / **unchecked**	If this field is checked, the variable value will be converted to floating point

After filling up the above fields, by pressing *Add Variable* button the new variable will appear on the left, under the corresponding node. To edit an existing variable you have to select it from the table on the left, edit the desired properties then press *Modify Variable* button.

It is possible to copy, paste or delete a node or a variable by right clicking with the mouse.

It is possible to copy and paste a variable from one node to another, or from one project to another, as well as copy and paste a whole node with all its variables.

5.6 Configuration update

The implemented configuration and possibly the updated firmware can be downloaded by pressing the *Update Device* button in the main form of the application program (see Figure 2 – Main form of the application program).

There can be 2 possible update sequences, the first in case the IP address assigned to the device is unknown, the second in case the IP address is known.

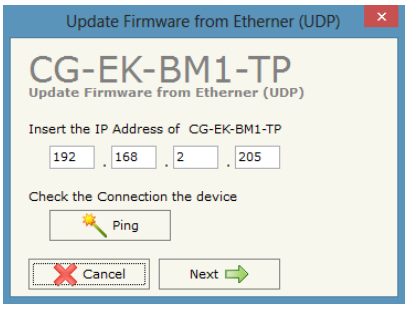


Figure 12 - Update configuration form

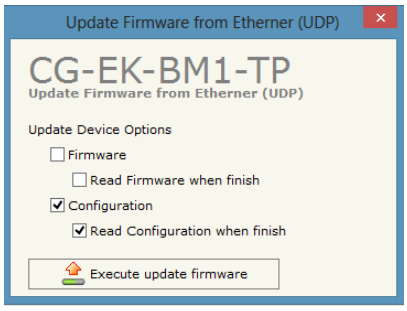


Figure 13 - Download options form

Sequence to follow in case of unassigned or unknown IP address:

- Power off the device
- Set the 1-way microswitch A (see Figure 1 – Switching, display and connection elements) to ON position
- Power on the device
- Connect PC and device by means of an Ethernet cable. Make sure that the PC's network parameters are consistent with the IP address assigned to the device in Boot Mode **192.168.2.205.** Otherwise, change the PC's network settings
- Write the IP address **192.168.2.205** inside the Update Configuration form (see Figure 12 – Update configuration form)
- Press *Ping* button; if you correctly applied the procedure, the text "*Device found!*" will appear
- Press *Next* button
- Select the desired options (see Figure 13 – Download options form): firmware update, configuration update or both
- Press *Execute update firmware* button
- When all operations are completed (see Figure 14 – Update in progress) shut down the device
- Set the 1-way microswitch A (see Figure 1 – Switching, display and connection elements) to OFF position
- Power on the device

If the sequence is successful, this means that firmware and/or configuration has been correctly downloaded on the device.

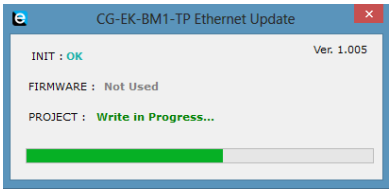


Figure 14 - Update in progress

Sequence to follow in case of known IP address:

- Power on the device with PC and device connected by means of an Ethernet cable
- Provide the device IP address (see Figure 12 – Update configuration form). Make sure that the PC's network parameters are consistent with the IP address assigned to the device**.** Otherwise, change the PC's network settings
- Press *Ping* button; if you correctly applied the procedure, the text "*Device found!*" will appear (see Figure 12 – Update configuration form)
- Press *Next* button (see Figure 12 – Update configuration form)
- Select the desired options (see Figure 13 – Download options form): firmware update, configuration update or both
- Press *Execute update firmware* button
- When all operations are completed (see Figure 14 – Update in progress) the device automatically switches back to Normal mode.

If the sequence is successful, this means that firmware and/or configuration has been correctly downloaded on the device.

It is recommended to update the firmware when a new version of the application program is installed or when configuring the device for the first time.

In case the update procedure goes into PROTECTION mode (see Figure 15 – Update error, "Protection" mode), you may want to check the following:

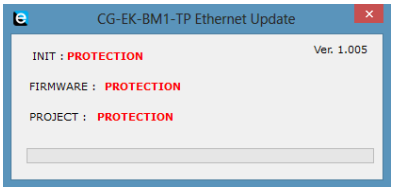


Figure 15 - Update error, "Protection" mode

- Repeat the update sequence
- Reboot your PC
- When running the program on a Virtual Machine, close it and rerun the program using the primary OS
- When using Windows 7 or later, make sure the user has administrator privileges
- Pay attention to firewall settings
- Check LAN configuration

In case of manual firmware update, replace “Sim67811.sim” file in the system folder “C:\Program Files (x86)\Ekinex\Compisitor_CG-EK-BM1-TP\Master”. After replacing, open *Update configurazione* form (see Figure 12 – Update configuration form) in the application program and start the proper sequence.

6 Warning

- Installation, electrical connection, configuration and commissioning of the device can only be carried out by qualified personnel.
- Opening the housing of the device causes the immediate end of the warranty period.
- ekinex® KNX defective devices must be returned to the manufacturer at the following address:

SBS S.p.A. Via Circonvallazione s / n, I-28010 Miasino (NO) Italy.

7 Other information

- This application manual is aimed at installers, system integrators and planners
- For further information on the product, please contact the ekinex® technical support at the e-mail address: support@ekinex.com or visit the website www.ekinex.com
- ekinex® is a registered trademark of SBS S.p.A.
- KNX® and ETS® are registered trademarks of KNX Association cvba, Brussels

•