User manual

vesion 1.0

KNXnet

Converter KNX / Ethernet

Passive PoE or 9-32V power

Indication of communication on KNX

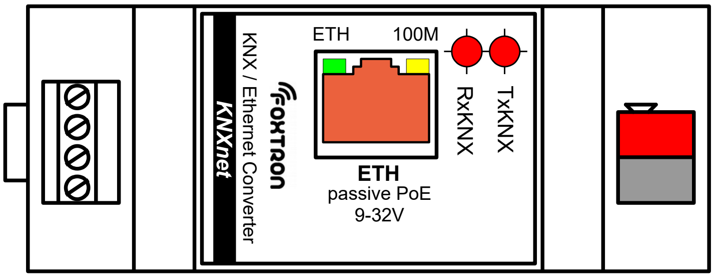
Galvanic separation KNX/ETH

DIN rail mount (2 modules)

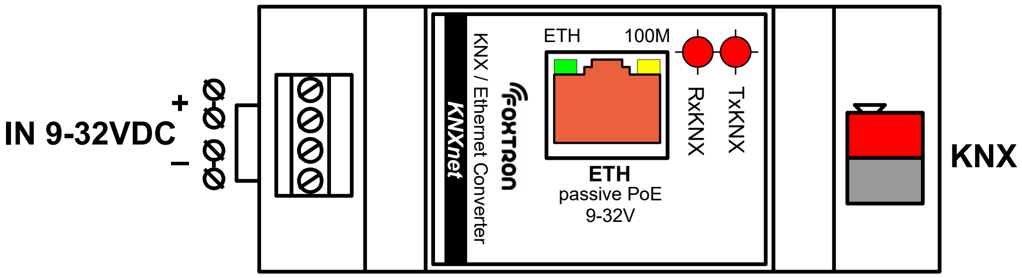
KNXnet is a KNX to Ethernet (TCP / IP) converter. TCP / IP communication is implemented using the simple ASCII protocol described in this manual. The transmitter can be powered by a passive PoE using an Ethernet data cable or by power on screw terminals.

Technical specification		
Bus	KNX / Ethernet	
KNX addresses	1000	
Speed (ETH)	10/100M	
Protocol (ETH)	TCP/IP + ASCII	
Power	9-32	V
Consumption	1,5	W
	50-150 (power supply 32-9V)	mA
Consumption (from KNX bus)	4	mA
Wires cross section (power)	0,08 – 1,5	mm^2
Wires cross section (KNX)	0,6 – 0,8	mm^2
Ingress protection	IP20	
Galvanic separation KNX/ETH	3,75	kV
Ambient working temperature	0 ÷ 50	°C
Storing temperature	-10 ÷ 50	°C
Weight	80	g

Signaling		
ETH	Connection to the ethernet network	
100M	Ethernet speed	
	off	10Mbit
	on	100Mbit
TxKNX	Sending data on KNX bus	
RxKNX	Receiving communication on KNX bus	

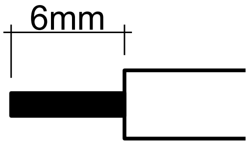


Terminals connection

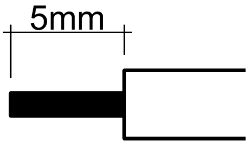


designation	*description*
KNX	KNX bus
ETH	Ethernet, powered by passive PoE
IN 9-32VDC	external DC 9-32V (alternative to PoE)

Wire preparation for power supply:



Wire preparation for KNX:

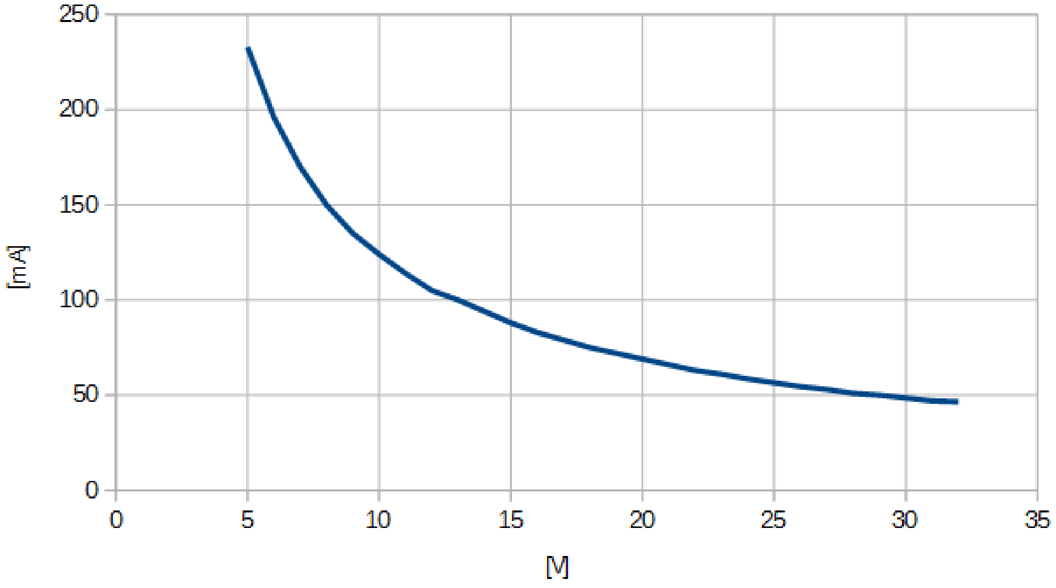


Power

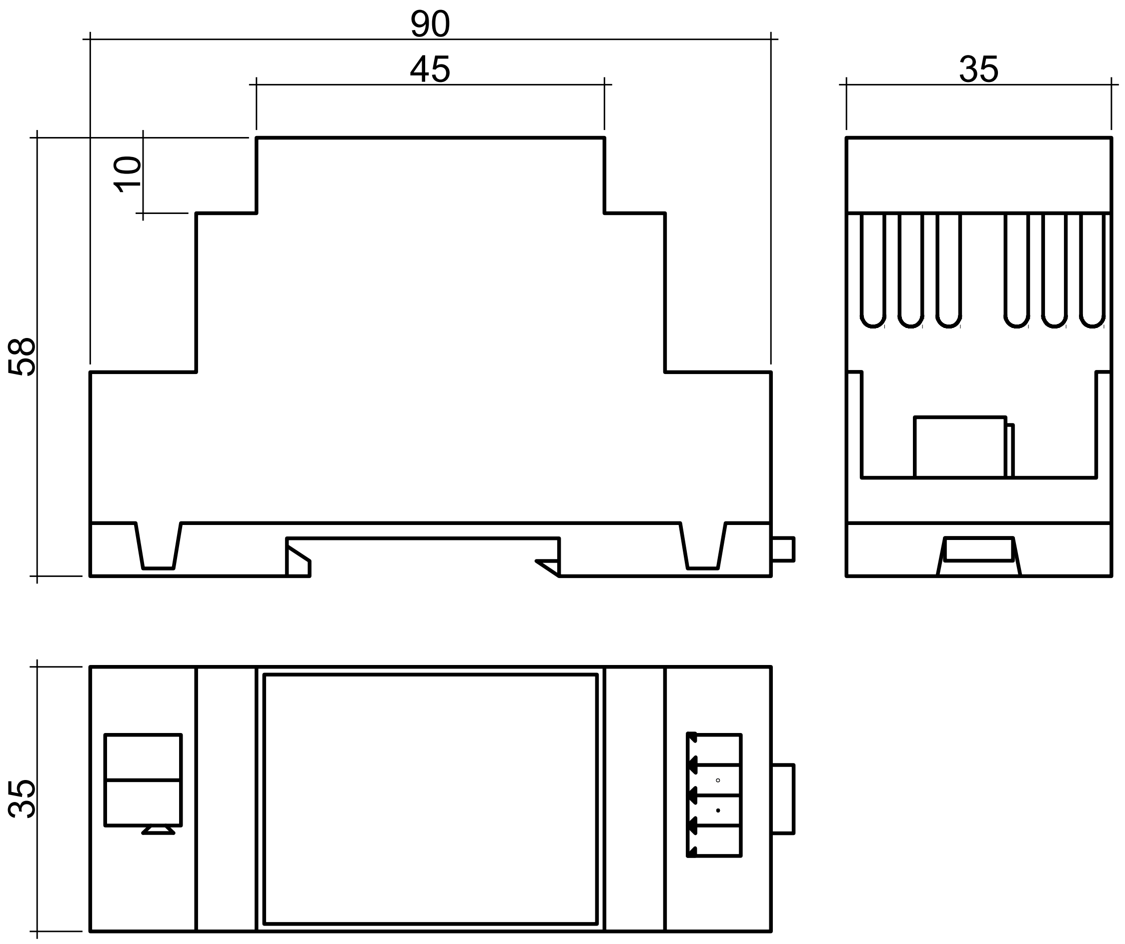
The unit has two power options, a passive PoE or an external power supply to the terminal IN 9-32VDC. The power consumption in both cases is 1.5W (the supply current is dependent on the connected supply voltage).

Passive PoE	The power is brought together via Ethernet to the ETH connector. The power is inserted into the data cable using a common passive PoE injector. The supply voltage can range from 9-32V.
IN 9-32VDC	DC power supply 9-32V to "IN 9-32VDC"

Consumtion dependent on voltage



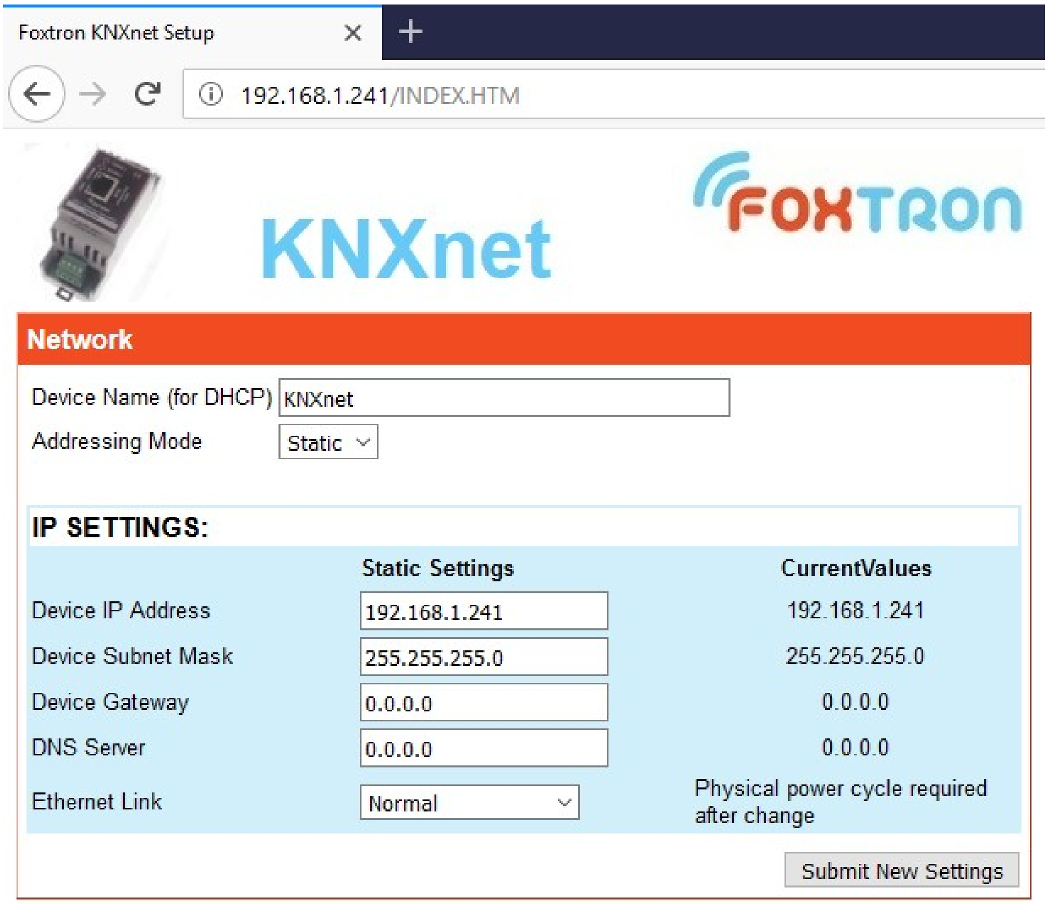
Dimensions (in mm)



Setup

Setting the IP address of the KNXnet converter is done using a web browser. Enter the IP address of the converter in the address bar. The default IP address is 192.168.1.241. If the address is not known, the converter can be found using software FOXnetFinder.exe, which can be downloaded from www.foxtron.eu
The communication port number is 23.

Default ethernet setting is shown in the following figure.



Network

Device Name (for DHCP): KNXnet
Addressing Mode: Static

IP SETTINGS:

	Static Settings	CurrentValues
Device IP Address	192.168.1.241	192.168.1.241
Device Subnet Mask	255.255.255.0	255.255.255.0
Device Gateway	0.0.0.0	0.0.0.0
DNS Server	0.0.0.0	0.0.0.0
Ethernet Link	Normal	Physical power cycle required after change

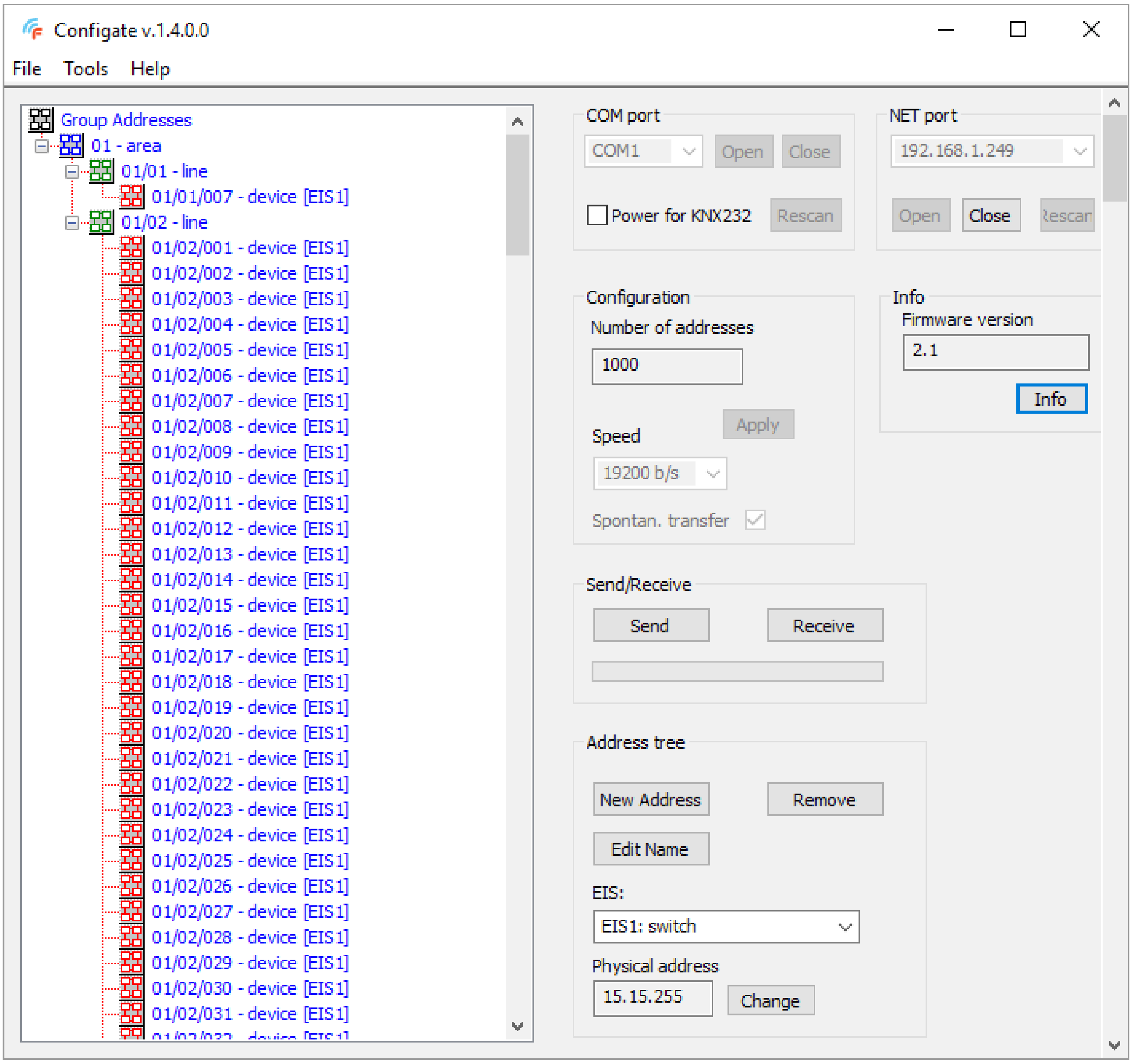
Device Name (for DHCP)	Name of KNXnet for DHCP server
Addressing Mode	Static – manual setting of IP address
	DHCP – automatic setting of IP address by DHCP server
Device IP Address	IP address of KNXnet converter (for Addressing Mode = Static)
Device Subnet Mask	
Device Gateway	Default gateway
DNS Server	Domain server name
Ethernet Link	Normal – automatic speed choice
	100BT Half duplex –Ethernet speed 100Mbps
	10BT Half duplex – Ethernet speed 10Mbps

Setting up listened and managed addresses is done using software Configate.exe, which can be downloaded from http://www.foxtron.eu/

KNXnet sets the listened KNX addresses, and then they are forwarded to Ethernet. Messages with an unlisted KNX address or bad DPT / EIS will be filtered and will not be forwarded from KNX to Ethernet or vice versa.

In Configate.exe, you need to use the Ethernet (NET port) setting.
The serial line is listed because the program is common to both the KNXnet and KNXnet drives.

By using Configate, it is possible to update it if a new firmware is released.
"Tools" → "Upload firmware .."



Configate v.1.4.0.0

- Group Addresses
  - 01 - area
    - 01/01 - line
      - 01/01/007 - device [EIS1]
    - 01/02 - line
      - 01/02/001 - device [EIS1]
      - 01/02/002 - device [EIS1]
      - 01/02/003 - device [EIS1]
      - 01/02/004 - device [EIS1]
      - 01/02/005 - device [EIS1]
      - 01/02/006 - device [EIS1]
      - 01/02/007 - device [EIS1]
      - 01/02/008 - device [EIS1]
      - 01/02/009 - device [EIS1]
      - 01/02/010 - device [EIS1]
      - 01/02/011 - device [EIS1]
      - 01/02/012 - device [EIS1]
      - 01/02/013 - device [EIS1]
      - 01/02/014 - device [EIS1]
      - 01/02/015 - device [EIS1]
      - 01/02/016 - device [EIS1]
      - 01/02/017 - device [EIS1]
      - 01/02/018 - device [EIS1]
      - 01/02/019 - device [EIS1]
      - 01/02/020 - device [EIS1]
      - 01/02/021 - device [EIS1]
      - 01/02/022 - device [EIS1]
      - 01/02/023 - device [EIS1]
      - 01/02/024 - device [EIS1]
      - 01/02/025 - device [EIS1]
      - 01/02/026 - device [EIS1]
      - 01/02/027 - device [EIS1]
      - 01/02/028 - device [EIS1]
      - 01/02/029 - device [EIS1]
      - 01/02/030 - device [EIS1]
      - 01/02/031 - device [EIS1]

COM port: COM1

□ Power for KNX232

NET port: 192.168.1.249

Configuration

Number of addresses: 1000

Speed: 19200 b/s

Spontan. transfer ☑

Info

Firmware version: 2.1

EIS: EIS1: switch

Physical address: 15.15.255

Protocol description

The data between the external device and the converter is sent via a simple protocol in ASCII format.
The TCP / IP communication port is **23**.

The communication is of Master - Slave type. Communication initiates question with an external device and the converter just responds. Only forwarded messages from the KNX can be sent to Ethernet without a response in case of spontaneous transmission of messages. With spontaneous transmission disabled, the rights that come from the KNX bus and are intended for external devices are stored in the encoder buffer. The external device reads it from the converter. The cache memory of the transmitter can hold up to 8 messages.

Structure of data telegram

Question or reply contains data

<STX>	function code	data	checksum	<CR>
1 character (0x02)	2 ASCII characters (1 byte)	Length by message type	2 ASCII characters (1 byte)	1 character (0x0D)

Reply with no data (confirmation)

Command / Question accepted:

<ACK>
0x06

Command / Question denied:

<NAK>
0x15

All othe bytes (function code, data, checksum) are numbers written in hex format (0x00 – 0xFF) in ASCII coding – in two bytes which can get only values
„0“ – „9“ (0x30 – 0x39) or „A“ – „F“ (0x41 – 0x46)

Example: function code 0x04 (Question on received data from KNX bus) is written in two bytes „0“ – 0x30 a „4“ – 0x34.
Whole message for Question on received data from KNX bus which is further described <STX> 0x04 0xFB <CR> is sent in 6 bytes (1B <STX>, 2B function code, 2B checksum and 1B <CR>). On the bus is therefore sent 0x02, 0x30, 0x34, 0x46, 0x42, 0x0D.

Calculation of the checksum

In the checksum are being counted all byte messages except STX and CR
Checksum is negation (negation of each bits) of sum function code and data.

Example: Question on address 1/1/1
STX 0x0C 0x09 0x01 0xE9 CR
0x0C + 0x09 + 0x01 = 0x16
Negation 0x16 is final - therefore 0xE9.

List of functions

Basic functions for communication with KNXnet converter.

Function code	Description
0x01	Converter firmware version question
0x04	Question on received data from KNX bus
0x0B	Write data to a KNX group address
0x0C	Question on group address on KNX

Configuration messages for the KNXnet converter setting. If settings are made by *Configate.exe*, you do not need to use the messages.

Function code	Description
0x40	Retrieve listened addresses of KNXnet converter
0x41	Set the KNXnet listened addresses
0x43	Deleting the list of listened addresses
0x44	Retrieve the communication speed
0x45	Set the communication speed
0x46	Retrieve spontaneous data transfer mode
0x47	Set spontaneous data transfer mode
0x48	Retrieve serial number
0x49	Retrieve hardware version
0x4A	Retrieve data format DPT/EIS
0x4B	Set data format DPT/EIS
0x4C	Retrieve maximal number of KNX addresses
0x4E	Retrieve of the ordering validity of the input addresses
0x70	Firmware overwrite mode
0x71	Firmware overwrite (just in firmware overwrite mode)
0x72	Bootloader version (just in firmware overwrite mode)
0x73	Memory size (just in firmware overwrite mode)

Converter firmware version question

Question	function code	**0x01**
	data	*not used*

Reply	function code	**0x81**
	data	Converter firmware version (2 byte) Version is stated in x.y format. First byte represents x, second y.

Example	Question	<STX> 0x01 0xFE <CR>
	Reply	<STX> 0x81 0x01 0x15 0x68 <CR> Firmware version is 1.21 (0x01.0x15)

Question on received data from KNX bus

Question	function code	**0x04**
	data	*not used*

Reply	function code	**0xFC**
	data	Group address (2 byte) data – n byte (depending on data type on KNX)

or	<ACK>	No new data from the KNX was received.

Example	Question	<STX> 0x04 0xFB <CR>
	Reply	<STX> 0xFC 0x09 0x01 0x07 0xF2 <CR> Data received from 1/1/1 address. Received byte 0x07.
		<ACK> No new data from the KNX was received.

If the spontaneous transmission mode is turned on, Question is not sent, but Answer with the 0xFC function code is sent automatically when bus data is received.

Write data to a KNX group address

Command	function code	**0x0B**
	data	Group address (2 byte) sending priority (1 byte) 0x0C – lowest priority 0x08 – alarm functions 0x04 – high priority 0x00 – system functions *data (n byte)* length according to data type on KNX

Reply	<ACK>	Data sent
or	<NAK>	Data sending error or the address is not in the list of listened address

Example	Question	<STX> 0x0B 0x09 0x01 0x0C 0x07 0xD7 <CR>
	Reply	<ACK> To address 1/1/1 data 0x07 written.

Question on group address on KNX

Question	function code	**0x0C**
	data	Group address (2 byte)

Reply	<ACK>	Data sent
or	<NAK>	Data sending error or the address is not in the list of listened address

Example	Question	<STX> 0x0C 0x09 0x01 0xE9 <CR>
	Reply	<ACK> Question to Group address 1/1/1 sent.

Setting of listened group addresses list

The message is used to configure the KNXnet converter. One message can set one listened group address. The addresses must be entered from the smallest to the largest into the memory. If configuration is made by *Configate.exe*, you do not need to use the message.

Command	function code	**0x41**
	data	Position of the item in the memory of the KNXnet converter (2 byte) 0-249 – list of listened group addresses 65535 (0xFFFF) physical address of KNXnet converter KNX address (2 byte) group or physical address 0xFFFF means in the case of a group address that this memory position will not be used (will delete the item) Data type (1 byte) group address format (1=DPT1 atd.) In the case of physical address 0 can be set.

Reply	function code	**0xC1**
	data	position of the item in the memory of the KNXnet converter (2 byte) 0-249 – list of listened group addresses 65535 (0xFFFF) physical address of KNXnet converter KNX address (2 byte) group or physical address Data type (1 byte) group address format (1=DPT1/EIS1 etc.) 0 means that this position is unused

Example	Question	<STX> 0x41 0x00 0x00 0x09 0x03 0x01 0xB1 <CR> Save group address 1/1/3 (0x09,0x03) with type DPT1 (0x01) in the first position of memory (0x00,0x00).
	Reply	<STX> 0xC1 0x00 0x00 0x09 0x03 0x01 0x31 <CR> In the first position of memory(0x00,0x00) is saved group address 1/1/3 (0x09,0x03) with type DPT1 (0x01).

Retrieve the list of listened addresses

The message is used to configure the KNXnet converter. One message can retrieve one listened group address. If configuration is made by *Configate.exe*, you do not need to use the message.

Question	function code	**0x40**
	data	Position of the item in the memory of the KNXnet converter (2 byte) 0-249 – list of listened group addresses 65535 (0xFFFF) physical address of KNXnet converter KNXnet

Reply	function code	**0xC0**
	data	position of the item in the memory of the KNXnet converter (2 byte) 0-249 – list of listened group addresses 65535 (0xFFFF) physical address of KNXnet converter KNX address (2 byte) group or physical address 0xFFFF in the case of group address means that this memory item is unused Data type group address format (1=DPT1/EIS1 etc.)

Example	Question	<STX> 0x40 0x00 0x00 0xBF <CR> Question to group address which is first in the first item of the converters memory
	Reply	<STX> 0xC0 0x00 0x00 0x09 0x03 0x01 0x32 <CR> In the first memory item is saved group address 1/1/3 (0x09,0x03) with type DPT1 (0x01).

Delete of the listened addresses list

The message is used to delete the configuration of the KNXnet converter. All the listened addresses are deleted by the message. Deletion can take up to 1.5 seconds.

Question	function code	**0x43**
	data	*not used*

Reply	function code	**0xC3**

data	*not used*

Example

Command	<STX> 0x43 0xBC <CR>
Reply	<STX> 0xC3 0x3C <CR>

Setting spontaneous data transfer mode

The message is used to enable or disable the spontaneous data transfer mode. If the mode is inactive, data received from the KNX bus are stored in the buffer, and the master system retrieves it using the "KNX Received Data" message (function code 0x04).When the mode is active, messages received on the KNX bus are sent spontaneously to the serial line. When the mode is activated, the contents of the message stacker are cleared.

Command

function code	**0x47**
data	spontaneous transfer (1 byte) 0 – off *1 – on*

Reply

function code	**0xC7**
data	spontaneous transfer (1 byte) 0 – off *1 – on*

Example

Command	<STX> 0x47 0x01 0xB7 <CR>
Reply	<STX> 0xC7 0x01 0x37 <CR> Spontaneous data transfer mode is on.

Read of spontaneous data transfer mode

The message is used to read the state of spontaneous data transfer mode.

Question

function code	**0x46**
data	*not used*

Reply

function code	**0xC6**
data	spontaneous transfer (1 byte) 0 – off 1 – on

Example	Question	<STX> 0x46 0xB9 <CR>
	Reply	<STX> 0xC6 0x01 0x38 <CR> Spontaneous data transfer mode is on.

Read of the serial number

The message is used to read the serial number of the KNXnet converter.

Question	function code	**0x48**
	data	*not used*

Reply	function code	**0xC8**
	data	Serial number (2 byte)

Example	Question	<STX> 0x48 0xB7 <CR>
	Reply	<STX> 0xC8 0x01 0x01 0x35 <CR> serial number 0x0101 = 257.

Read of HW version

The message is used to read hardware version of the converter KNXnet.

Question	function code	**0x49**
	data	*not used*

Reply	function code	**0xC9**
	data	hw version (2 byte) Version is stated in the x.y format, where first byte represents x and second y.

Example	Question	<STX> 0x49 0xB6 <CR>
	Reply	<STX> 0xC9 0x00 0x01 0x35 <CR> HW version is 0.1

Read of data format DPT/EIS

The message is used to read used data format for addresses stored in list of listened addresses.

Question	function code	**0x4A**
	data	*not used*

Reply	function code	**0xCA**
	data	Data format (1 byte) 0 – DPT 1 – EIS

Example	Question	<STX> 0x4A 0xB5 <CR>
	Reply	<STX> 0xCA 0x00 0x35 <CR> DPT data type is used.

Setting data format DPT/EIS

The message is used to set the used data format for the addresses stored in the listened addresses list.
The message is used to enable or disable the spontaneous data transfer mode.

Command	function code	**0x4B**
	data	Data format (1 byte) 0 – DPT 1 – EIS

Reply	function code	**0xCB**
	data	Data format (1 byte) 0 – DPT 1 – EIS

Example	Command	<STX> 0x4B 0x00 0xB4 <CR>
	Reply	<STX> 0xCB 0x00 0x34 <CR> The spontaneous data transfer mode is on.

Retrieve the maximum number of addresses on KNX

The message is used to retrieve the maximum number of controlled addresses on the KNX. The KNXnet is manufactured only in one type with 1000 addresses.

Question	function code	**0x4C**
	data	*not used*

Reply	function code	**0xCC**
	data	Maximum number of controlled addresses on KNX (2 byte)

Example	Question	<STX> 0x4C 0xB3 <CR>
	Reply	<STX> 0xCC 0x03 0xE8 0x48 <CR> Maximum number 0x03E8 = 1000 KNX addresses

Retrieving the validity of the ordering KNX addresses

The message is used to read if the data is properly sorted. For KNXnet it is necessary to write individual KNX addresses from the lowest to the highest. If settings are made by Configate.exe, the addresses are automatically sorted before uploading.

Command	function code	**0x4E**
	data	*not used*

Reply	function code	**0xCE**
	data	Information about the validity of the ordered KNX addresses(1 byte) 0 – Unsorted data. Some addresses might be ignored. 1 – The data is sorted correctly.

Example	Command	<STX> 0x4E 0xB1 <CR>
	Reply	<STX> 0xCE 0x01 0x30 <CR> The data is sorted correctly.

Firmware overwriting mode

The message switches the unit to firmware update mode. After switching, you need to wait 100ms before the firmware overwriting mode initializes. If overwriting does not begin within 2s, the unit automatically switches back to the standard function.

Povel	function code	**0x70**
	data	*not used*

Reply	function code	**0xF0**
	data	*not used*

Example	Question	<STX> 0x70 0x8F <CR>
	Reply	<STX> 0xF0 0x0F <CR> Firmware overwriting mode is on.

Firmware overwrite

After switching to firmware overwrite mode, individual lines are sent in the Intel HEX format. The initial character ":" and the Intel HEX checksum are not sent (last 2 characters). After complete upload, the new firmware is started automatically (firmware overwrite mode).

Command	function code	**0x71**
	data	Intel HEX file line without and initial character “:” and checksum.

Reply	function code	**0xF1**
	data	Data (1 byte) 0 – Data received 1 – Maximum message length exceeded 2 – Command is out of memory length 3 – Error writing to memory

Example	Command	<STX> 0x71 0x04 0x00 0x00 0x00 0xF8 0x94 0xFF 0xCF 0x30 <CR>
	Reply	<STX> 0xF1 0x00 0x0E <CR> Line „:04000000F894FFCF“ Intel HEX file successfully written.

Note: For the first line of the HEX file a 4s timeout is required. For other lines 100ms is sufficient.

Bootloader version

Inquiry on the firmware version of the program in the unit. Available only in firmware overwrite mode.

Question	function code	**0x72**
	data	*not used*

Reply	function code	**0xF2**
	data	Bootloader version (2 byte) Version is stated in the x.y format, where first byte represents x and second y.

Example	Question	<STX> 0x72 0x8D <CR>
	Reply	<STX> 0xF2 0x00 0x01 0x0C <CR> Bootloader version is 0.1

Memory size

Question on the memory size for firmware upload. The return value indicates the number of 128B blocks. Available only in firmware overwrite mode.

Question	function code	**0x73**
	data	*not used*

Reply	function code	**0xF3**
	data	Memory size (2 byte)

Example	Question	<STX> 0x73 0x8C <CR>
	Reply	<STX> 0xF3 0x00 0xDE 0x2E <CR> Memory for firmware is 222*128 = 28416B

tel: +420 274 772 527
e-mail: info@foxtron.cz
web: www.foxtron.eu