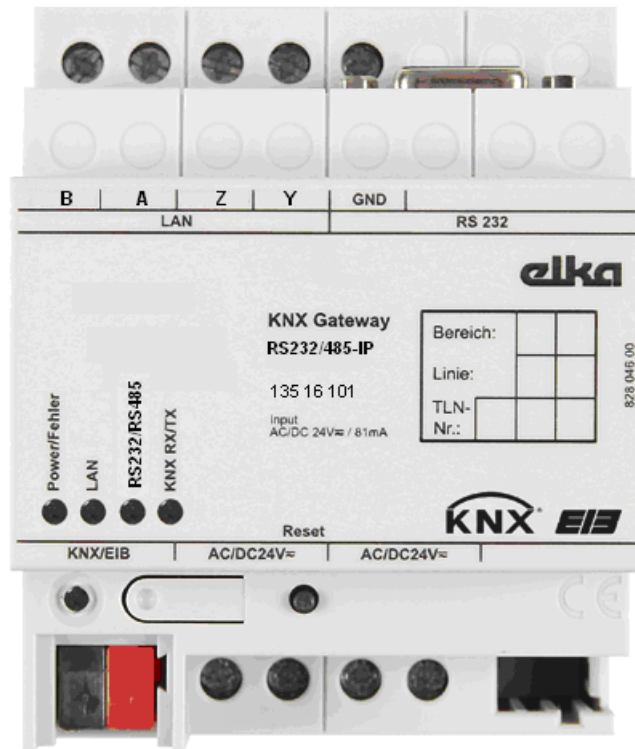


# 1 KNX Gateway RS232/485-IP, Tool software



All rights reserved. Reproduction, publication or other use is subject to prior written approval.

© 2015 ELKA-Elektronik GmbH

- 1 KNX Gateway RS232/485-IP, Tool software..... 1
- 2 Description of functions..... 4
- 3 Hardware description ..... 5
  - 3.1 Device configuration ..... 5
  - 3.2 Status displays..... 6
  - 3.3 Technical data ..... 6
- 4 KNX Gateway RS232/485-IP ..... 8
  - 4.1 Operating modes ..... 8
    - 4.1.1 Standard ASCII protocol..... 8
    - 4.1.2 Protocol for string processing..... 8
  - 4.2 Interfaces ..... 8
    - 4.2.1 KNX ..... 8
    - 4.2.2 RS485..... 8
      - 4.2.2.1 4-wire-operation (full duplex) ..... 9
      - 4.2.2.2 2-wire-operation (half duplex) ..... 9
      - 4.2.2.3 Terminating resistor ..... 10
    - 4.2.3 LAN..... 10
    - 4.2.4 RS232..... 10
  - 4.3 Diagnosis function ..... 10
  - 4.4 Use of interfaces..... 11
- 5 KNX-Gate2..... 12
  - 5.1 Program start..... 12
  - 5.2 Choice of the protocol..... 13
  - 5.3 Device parameters..... 14
    - 5.3.1 Interface settings ..... 14
      - 5.3.1.1 IP parameters ..... 15
      - 5.3.1.2 RS232 and RS485 parameters ..... 16
    - 5.3.2 KNX parameters ..... 17
  - 5.4 KNX objects and group addresses ..... 18
    - 5.4.1 Create KNX objects ..... 18
    - 5.4.2 Import group addresses ..... 19
    - 5.4.3 Create and edit group addresses ..... 20
    - 5.4.4 Connect group addresses with objects ..... 21
  - 5.5 Operation mode Strings..... 22
    - 5.5.1 Creating Strings..... 23
      - 5.5.1.1 Text strings ..... 23
      - 5.5.1.2 Hexadecimal Strings..... 25
    - 5.5.2 Send and receive Strings ..... 25
    - 5.5.3 Export and import Strings..... 26
  - 5.6 Documentation..... 27
  - 5.7 Commissioning ..... 27
    - 5.7.1 Choice of interface ..... 27
    - 5.7.2 IP settings..... 27
    - 5.7.3 RS232-settings ..... 29
    - 5.7.4 Download ..... 29
  - 5.8 Reconstruction..... 29
  - 5.9 Diagnosis ..... 29
- 6 Appendix ..... 32
  - 6.1 ASCII protocol..... 32
    - 6.1.1 Spontaneous sending / polling ..... 32
    - 6.1.2 Telegram structure ..... 32
    - 6.1.3 Version inquiry of the gateway firmware ..... 33
    - 6.1.4 Inquiry on incoming bus telegrams ..... 33
    - 6.1.5 Reset Gateway..... 34
    - 6.1.6 Describe KNX group address..... 34
    - 6.1.7 Read out KNX group address ..... 35
    - 6.1.8 Error message..... 35
  - 6.2 Formatting of strings ..... 37

6.2.1	General description .....	37
6.2.2	KNX data point types.....	38
6.2.2.1	Simple data point types (1 bit, 2 bits, 4 bits).....	39
6.2.2.2	Integer types .....	39
6.2.2.3	Floating point types .....	40
6.2.2.4	Date and time .....	40
6.2.2.5	Special types .....	40

## 2 Description of functions

The *KNX gateway RS232/485-IP* is an intelligent DIN-rail mounted system device for coupling of subsystems via a serial RS232 or RS485 interface to the KNX. To this effect, the KNX gateway *RS232/485-IP* is connected with the KNX and the subsystem via the corresponding interface. The gateway requires 24 V AC/DC supply voltage.

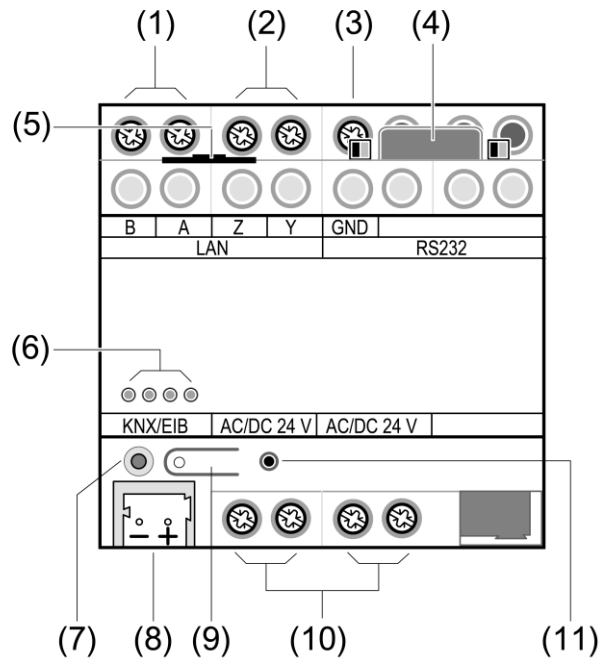


For connection of the KNX gateway RS232/485-IP the mounting and operating instructions must be observed.

The functionality of the KNX gateway RS232/485-IP is set by programming.

### 3 Hardware description

#### 3.1 Device configuration



**Fig. 3.1-1: Device configuration**

- (1) RS485-input with 4-wire-operation (B, A)  
Not used with 2-wire-operation
- (2) RS485-output with 4-wire-operation (Z, Y)  
RS485-input/output with 2-wire-operation
- (3) RS485-ground (GND)
- (4) RS232 port
- (5) Ethernet/IP port
- (6) Status LED
- (7) Programming LED
- (8) KNX port
- (9) Programming key
- (10) External supply port
- (11) Reset key

## 3.2 Status displays

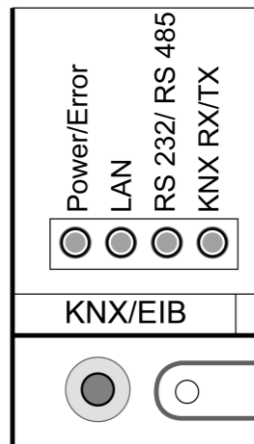


Fig. 3.2-1: Status LED

Power/Error	Shining green: normal operation Flashing orange: invalid or no project Flashing red: invalid firmware
LAN	Shining yellow: Reception via Ethernet/IP interface
RS232/RS485	Flashing green: Receiving/transmitting via RS232 interface Flashing red: Receiving/transmitting via RS485 interface
KNX RX/TX	Flashing red: Receiving from KNX bus. Flashing green: Transmitting to KNX bus. Flashing red-green: No KNX bus detected

## 3.3 Technical data

### External supply

Mains voltage AC/DC  
Mains frequency  
Power consumption

AC/DC 24 V SELV ( $\pm 10\%$ )  
50 / 60 Hz  
max. 2 VA

**Ambient conditions**

Ambient temperature	- 5 °C to + 45 °C
Storage and transport temp.	- 25 °C to + 70 °C
Humidity (ambient / storage / transport)	
Protection class	III
Installation width	72 mm / 4 pitches
Weight	approx. 175 g

**Network communication**

Bit rate IP	10 / 100 Mbit/s
Ethernet/IP port	RJ45 socket

**RS232**

Bit rate RS232	1.2 kbit/s ... 115.2 kbit/s
RS232 port	9-pol. D-Sub socket
Protective circuit	DCE

**KNX**

KNX medium	TP 1
Commissioning mode	S-mode
Nominal voltage KNX	DC 21 V ... 32 V SELV
KNX port	Standard KNX / EIB bus connection terminals
Power consumption KNX	typ. 150 mW

**RS485**

Bit rate RS485	1.2 kbit/s ... 115.2 kbit/s
----------------	-----------------------------

**Supply and RS485 port**

Port type	Screw terminal
Single-core	0.5 ... 4 mm <sup>2</sup>
Finely stranded without wire-end sleeve	0.34 ... 4 mm <sup>2</sup>
Finely stranded with wire-end sleeve	0.14 ... 2.5 mm <sup>2</sup>

## 4 KNX Gateway RS232/485-IP

### 4.1 Operating modes

The *KNX gateway RS232/485-IP* is used for bi-directional data exchange between KNX and a subsystem.

It supports the two operating modes

- ELKA standard ASCII protocol
- Freely definable strings

The operating mode used is defined in the *KNX-Gate2* software when a new project is created.

#### 4.1.1 Standard ASCII protocol

Via KNX, the gateway receives the telegrams of the bus line. It filters the desired information and passes them on to the connected device in ASCII format. Since the gateway receives all group telegrams, the group addresses are filtered on the basis of projected data.

In opposite direction, the connected device transmits the KNX addresses and the assigned information to the gateway in ASCII format. In the gateway, the transmitted information are validated, converted into KNX telegrams and transmitted to the KNX.

#### 4.1.2 Protocol for string processing

In this operating mode strings can be defined which are sent to the subsystem when a defined KNX telegram is received. In opposite direction strings can be defined releasing a defined KNX telegram when sent by the subsystem.

### 4.2 Interfaces

The gateway has got four different interfaces which are used depending on the active operating mode.

#### 4.2.1 KNX

KNX connection is effected via a standard KNX bus terminal (8).

#### 4.2.2 RS485

The RS485 port can be used to connect a subsystem. This port can either be used in 4-wire-operation (full duplex), or in 2-wire-operation (half duplex). The operating mode is defined in the *KNX-Gate2*.

#### 4.2.2.1 4-wire-operation (full duplex)

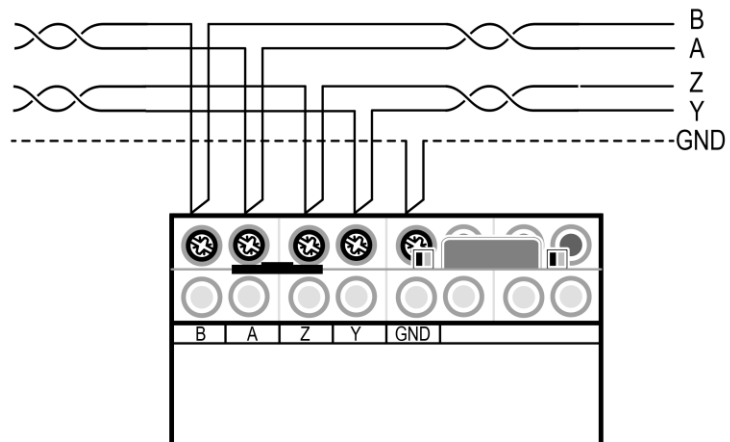


Fig. 4.2-1: RS485, 4-wire-operation

In 4-wire-operation the gateway receives via terminals B and A and transmits via terminals Z and Y.

#### 4.2.2.2 2-wire-operation (half duplex)

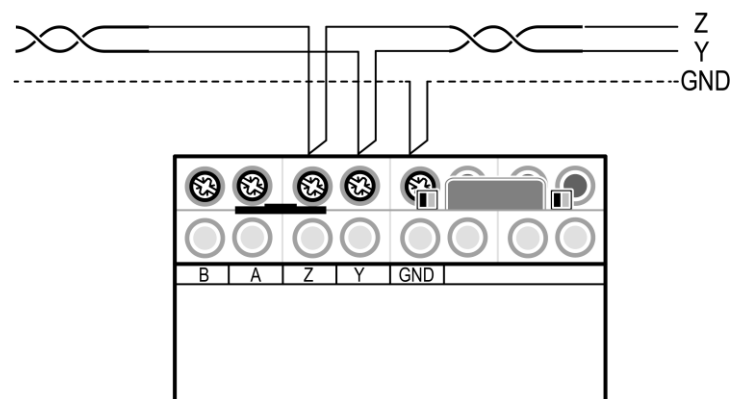


Fig. 4.2-2:RS485, 2-wire-operation

In 2-wire-operation the gateway sends and receives via terminals Z and Y. Terminals B and A are not used.

### 4.2.2.3 Terminating resistor

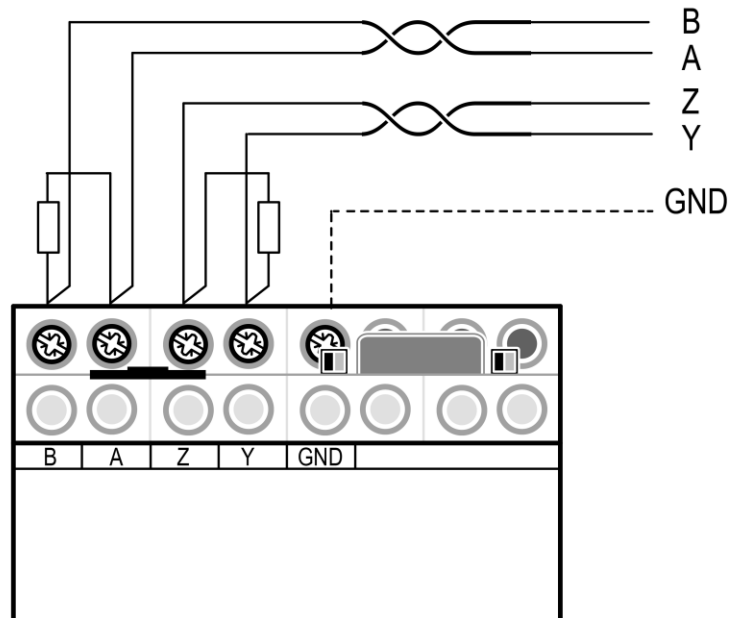


Fig. 4.2-3: RS485, terminating resistors at lead end

At the last device within the RS485 installation, the RS485 bus should be terminated with the terminating resistor supplied.

### 4.2.3 LAN

The LAN port can be used for commissioning of the gateway and for connection of a subsystem. For connection of a subsystem the serial data are tunnelled by IP.

The connection to the network or to the PC is effected via the RJ45 socket LAN. If necessary, for direct connection of a PC a crosslink network cable must be used.

### 4.2.4 RS232

The RS232 port can be used for connection of a subsystem or for commissioning of the gateway.

The RS232 port is designed as 9-pole DCE (data communication end unit = modem). PC connection is effected via an RS232 cable (plug and socket) with 1:1 configuration (no null modem cable)

## 4.3 Diagnosis function

By means of the diagnosis function, the commissioning engineer can find the cause for faulty telegrams or those that have not been passed on (KNX->target system and vice versa). To this effect data can be emitted via an interface which is not used for gateway operation. *KNX-Gate2* is required for evaluation of the data.

## 4.4 Use of interfaces

The gateway has got three interfaces RS232, RS485 and Ethernet for communication with a PC or a subsystem.

In parameter setting one of the three interfaces can be used as operation interface for bi-directional communication to the KNX.

The RS232 interface and the Ethernet interface can be used as diagnosis interface. One interface cannot be used as operation and diagnosis interface at the same time.

Independent of being set as operation or diagnosis interface, the RS232 interface and the Ethernet interface can be used for commissioning and reconstruction.

**Please note:** The IP-specific data (IP address, network template) can be loaded only via the Ethernet interface.

## 5 KNX-Gate2

### 5.1 Program start

With the first start, *KNX-Gate2* shows an empty project.

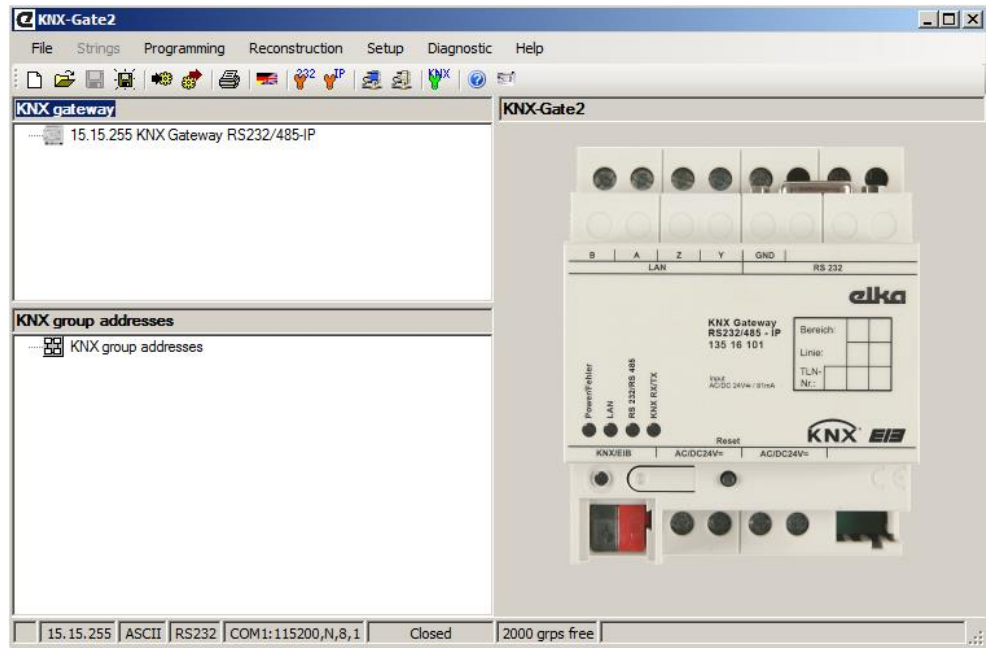


Fig. 5.1-1: Start screen

The start screen consists of six areas:

- **Menu bar**  
In the menu bar all commands for administration of projects and for direct access to the current gateway are available.
- **Tool bar**  
The tool bar offers quick access to frequently used functions.
- **Object tree**  
In the object tree KNX objects can be created and edited.
- **Group address tree**  
The group address tree shows all group addresses which can be imported from ETS or created manually. By drag and drop the group addresses can be assigned to the KNX objects.
- **Operation and information area**  
The operation and information area gives further information on the element marked in the object tree or in the group address tree.
- **Status bar**  
The status bar shows basic data on the projected gateway, on the commissioning interface, on the available group addresses and on the communication between PC and gateway. The fields with the physical address and the configuration data of the interface can also be used to open the related configuration windows.

In the menu *Settings* the option *Start with last project* can be activated so that the current project is opened automatically when the program is started.

## 5.2 Choice of the protocol

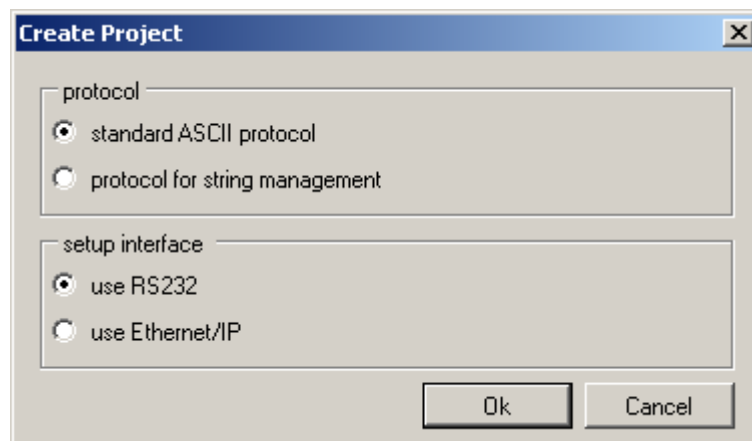


Fig. 5.2-1: Create project

When a new project is created, the user has to choose the protocol in which the gateway is to be operated. This setting cannot be modified within an existing project.

If an existing project is opened, the suitable operation mode is set automatically.

Old projects of previous *KNX-Gate* versions can also be imported in eg2-format.

On request, the commissioning interface can also be modified later.

### 5.3 Device parameters

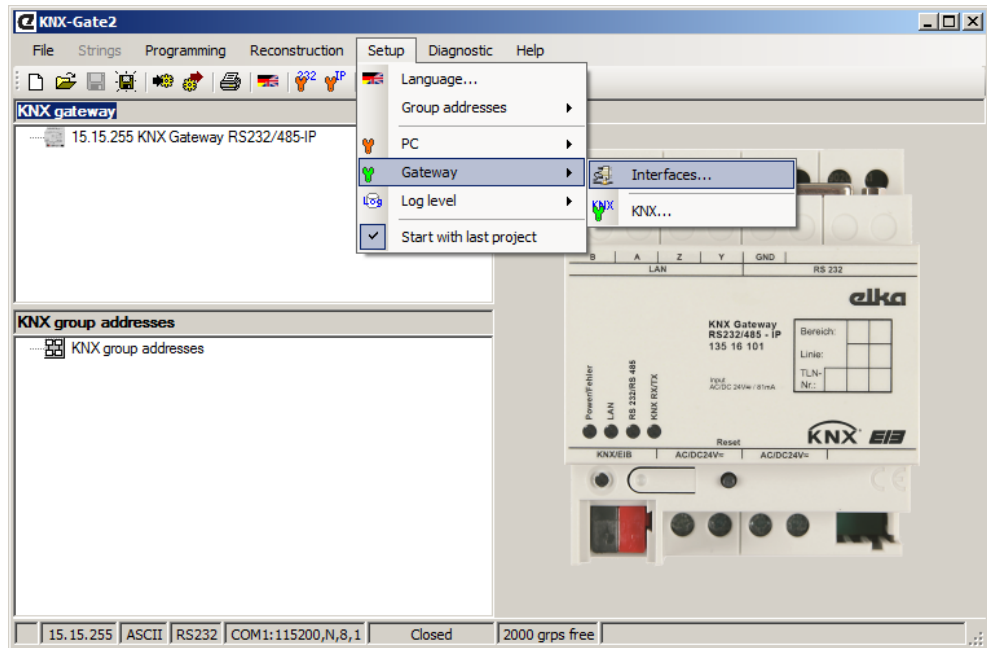


Fig. 5.3-1: Setting of device parameters

Via the menu item *Settings/Gateway* the communication settings of the gateway are made.

#### 5.3.1 Interface settings

With the dialogue *Setup Gateway interfaces* the functions of the different interfaces can be chosen. Here different ports have to be used for the operation interface for connection with a subsystem and for the interface for diagnosis with *KNX-Gate2*.

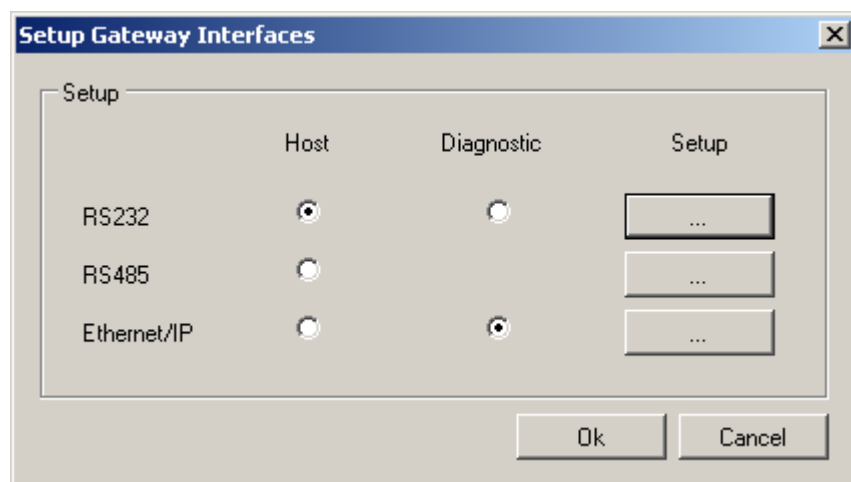


Fig. 5.3-2: Choice of interface

Afterwards, the parameters of the interfaces are stored in different dialogues.

Independent of the operation interface and the diagnosis interface, commissioning can be effected via the RS232 interface or the IP-interface.

### 5.3.1.1 IP parameters

In the dialogue *Settings Gateway Ethernet/IP* the basic communication parameters for network connection of the gateway are set.

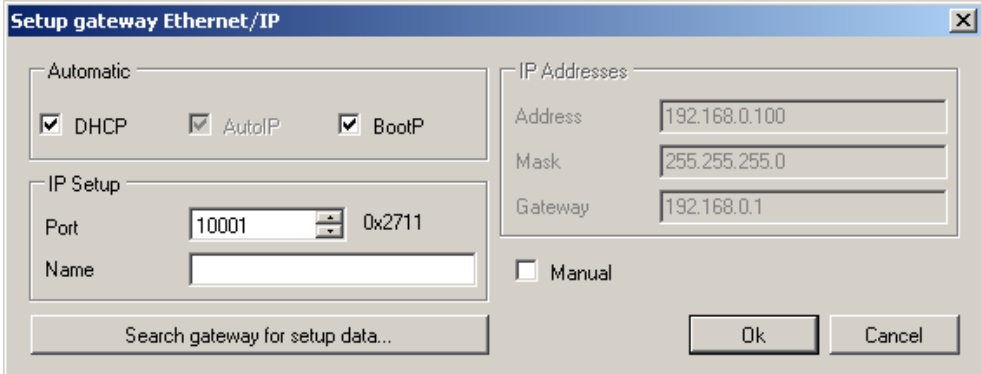
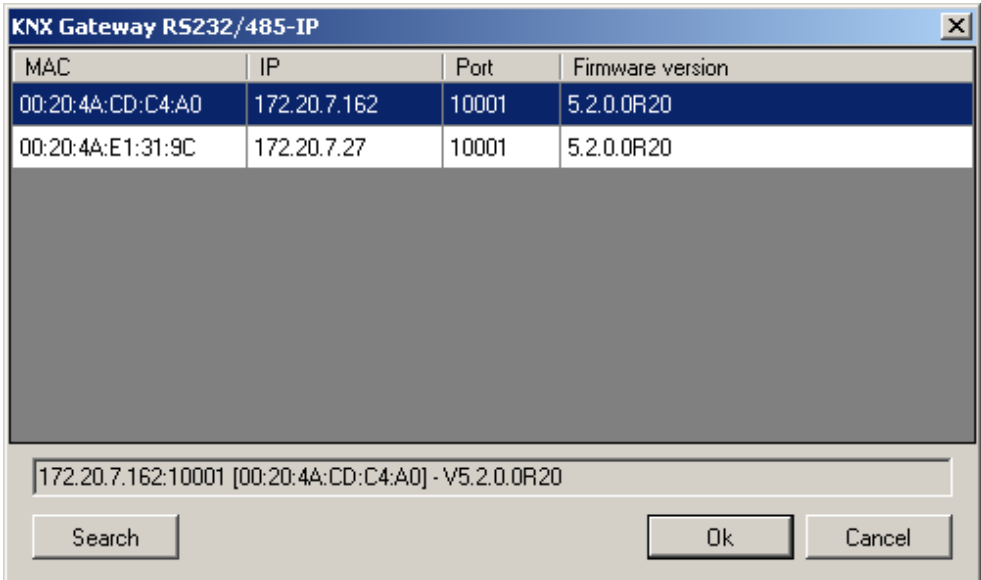


Fig. 5.3-3: Setup IP parameters

In most cases, the gateway will normally receive an automatic IP address. Available processes: DHCP, AutoIP and BootP. Alternatively, the IP settings can also be set manually.



MAC	IP	Port	Firmware version
00:20:4A:CD:C4:A0	172.20.7.162	10001	5.2.0.0R20
00:20:4A:E1:31:9C	172.20.7.27	10001	5.2.0.0R20

Fig. 5.3-4: Search gateway and select

The button *Search Gateway for setup data* opens the dialogue *KNX Gateway RS232/485-IP*, in which all gateways found are listed with their MAC addresses.

If the gateway is not listed in this dialogue, it might be possible that the network cable of the gateway was not connected when the power

supply was switched on. In this case the gateway might have set an IP address which does not match the PC settings. In this case switch off the supply voltage of the gateway and switch it on again when the network cable has been connected.

The list is updated by pressing the *Search* button .

### 5.3.1.2 RS232 and RS485 parameters

The settings of the two serial interfaces RS232 and RS485 are nearly identical.

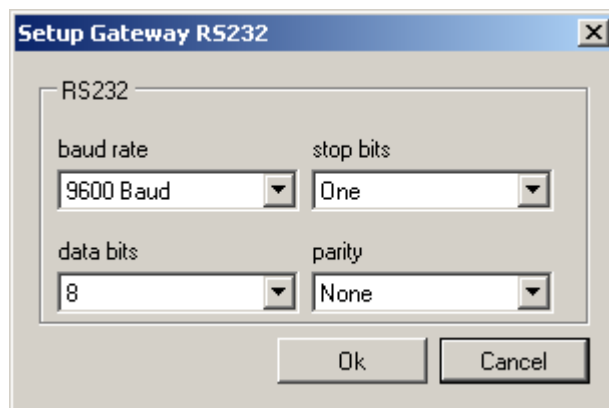


Fig. 5.3-5: Settings Gateway RS232

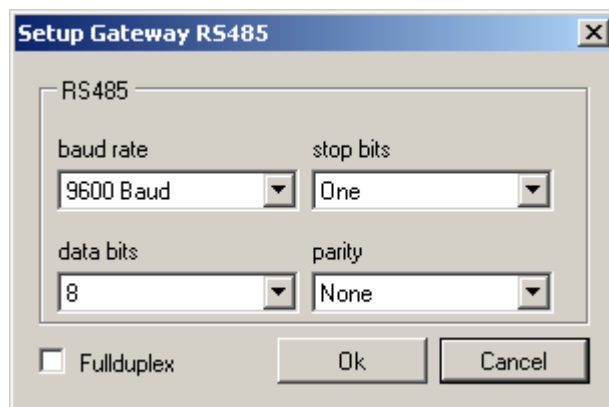


Fig. 5.3-6: Settings Gateway RS485

Transmission speed (baud rate) can be chosen between 1200 baud and 115200 baud. In string operation also the data format can be adapted. In ASCII operation the data format is preset to 8 data bits, one stop bit without parity bit.

For the RS485 interface additionally the option *Full duplex* (= 4-wire-operation) can be activated.

### 5.3.2 KNX parameters

The dialogue *Settings Gateway KNX* provides different options to control the transmission behaviour of the gateway in both directions.

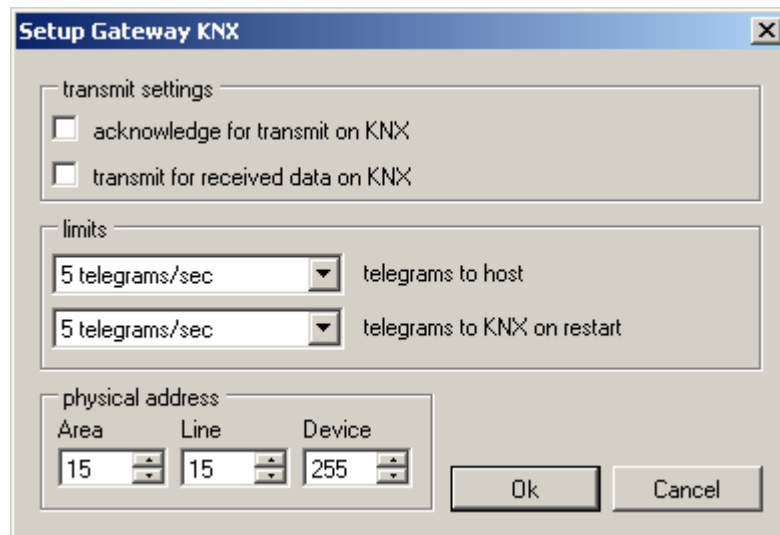


Fig. 5.3-7: Setup KNX parameters

If the subsystem / the PC is to control the communication of the gateway, the option *acknowledge for transmit on KNX* should be activated. In this case the gateway gives a confirmation to the subsystem when a transmission command has been executed successfully.

If the option *transmit for received data on KNX* is deactivated, the gateway sends new data to the subsystem only on demand. If the option is activated, the gateway can automatically send new data to the subsystem.

When a telegram has been sent to the subsystem, the subsystem should confirm receipt of the telegram with a positive acknowledgement *ACK*. If no positive acknowledgement is received or if the gateway receives a negative acknowledgement *NAK*, the gateway waits for a time that can be set via the parameter *Telegrams to subsystem*. With the settings *2 telegrams/sec* or *5 telegrams/sec* the gateway correspondingly waits 500 ms or 200 ms for an acknowledgement. If the subsystem does not give out an acknowledgement in general, the setting *no transmit limitation* is sensible. The default value is 5 telegrams/sec.

After a restart the gateway can inquire the values of selected KNX group addresses. Here the bus load in context with the response telegram to be expected has to be considered. For this reason the gateway here also has got an adjustable transmission limitation with the options *2 telegrams/sec* or *5 telegrams/sec*. The default value is 5 telegrams/sec.

In the dialogue *Setup Gateway KNX* it is set which physical address the gateway uses on the KNX bus. This address is automatically written into the gateway when the project is loaded with KNX-Gate2.

## 5.4 KNX objects and group addresses

### 5.4.1 Create KNX objects

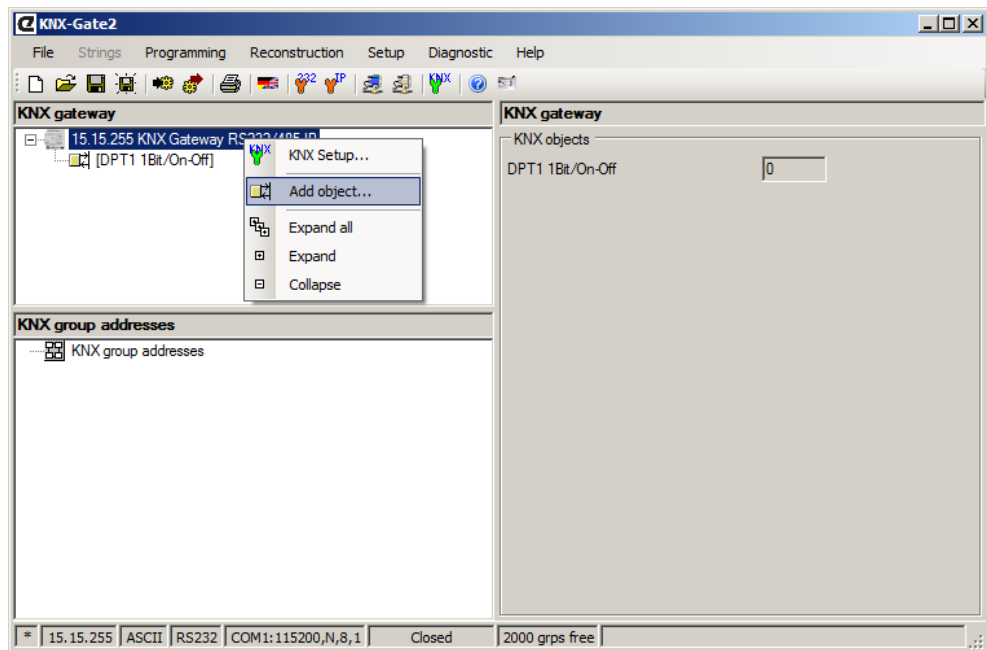


Fig. 5.4-1: Add KNX object

In the object tree the required KNX objects can be created via the context menu. Here the suitable data point type must be assigned to each object.

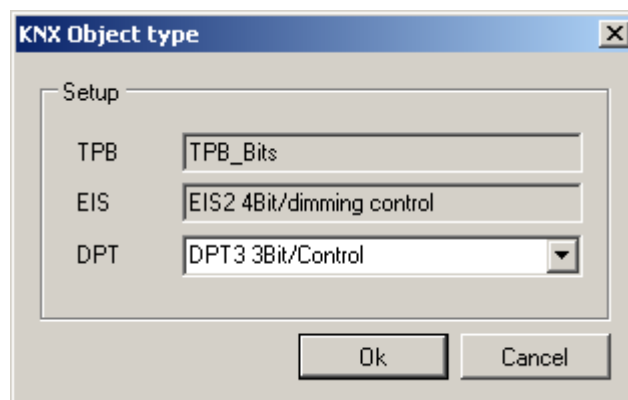


Fig. 5.4-2: Select KNX object type

## 5.4.2 Import group addresses

Matching the presentation used in the ETS, the group addresses can be used with a 2-level, 3-level or free structure. Switch-over can be effected via the menu *Settings -> Group addresses*. This only changes the display. The internal administration of the group addresses remains unchanged.

The KNX group addresses can either be taken over from an existing project of ETS3 or ETS4 or created manually.

To take over the group addresses from an ETS3 project, they must be exported in ETS3 into a CSV-file.

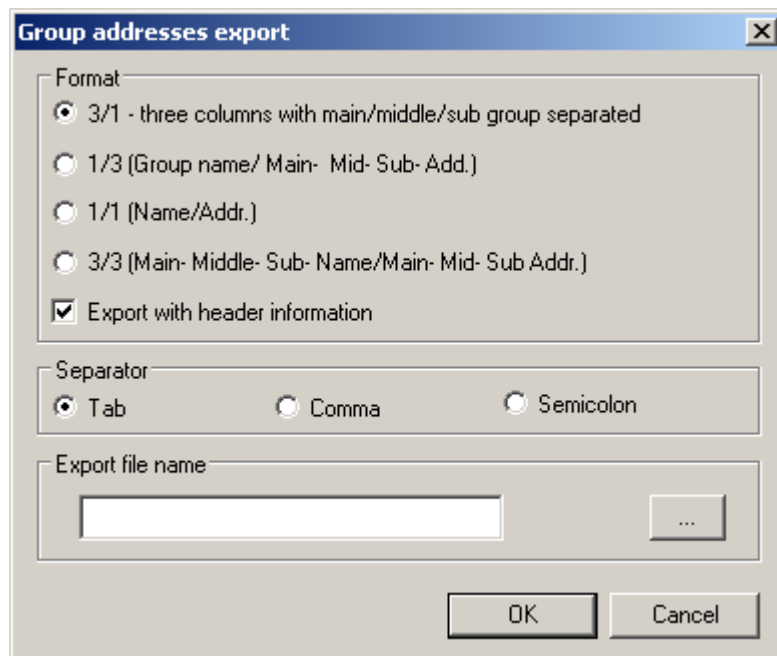


Fig 5.4-3: ETS3 group addresses export

The dialogue *Group addresses export* is opened in the ETS3 window *Group addresses* via the context menu of the node *Main groups*. In this dialogue the format *3/1 – three columns, Main/Middle/subgroup separate* is to be set.

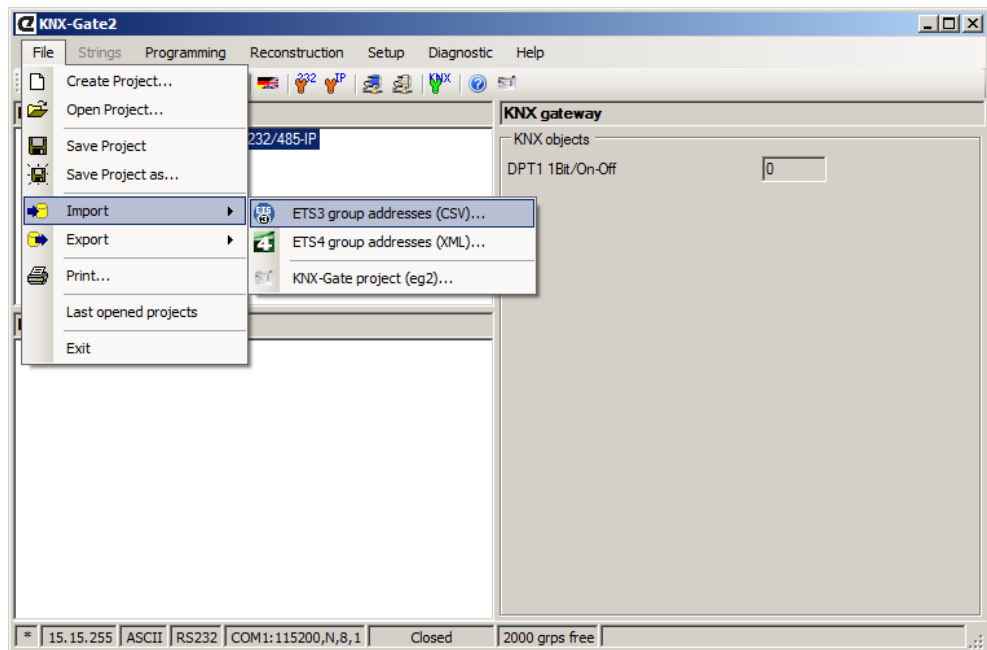


Fig. 5.4-4: Import group addresses

In *KNX-Gate2* the group addresses out of ETS3 or ETS4 can be taken over via the menu item *File -> Import* or via the context menu on the main node in the group address tree.

### 5.4.3 Create and edit group addresses

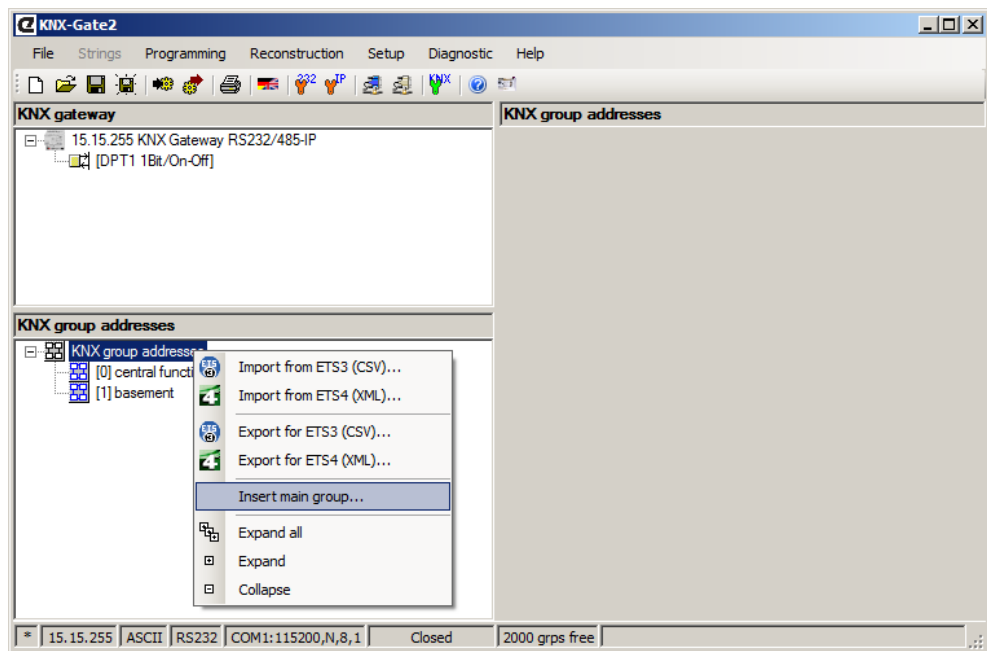
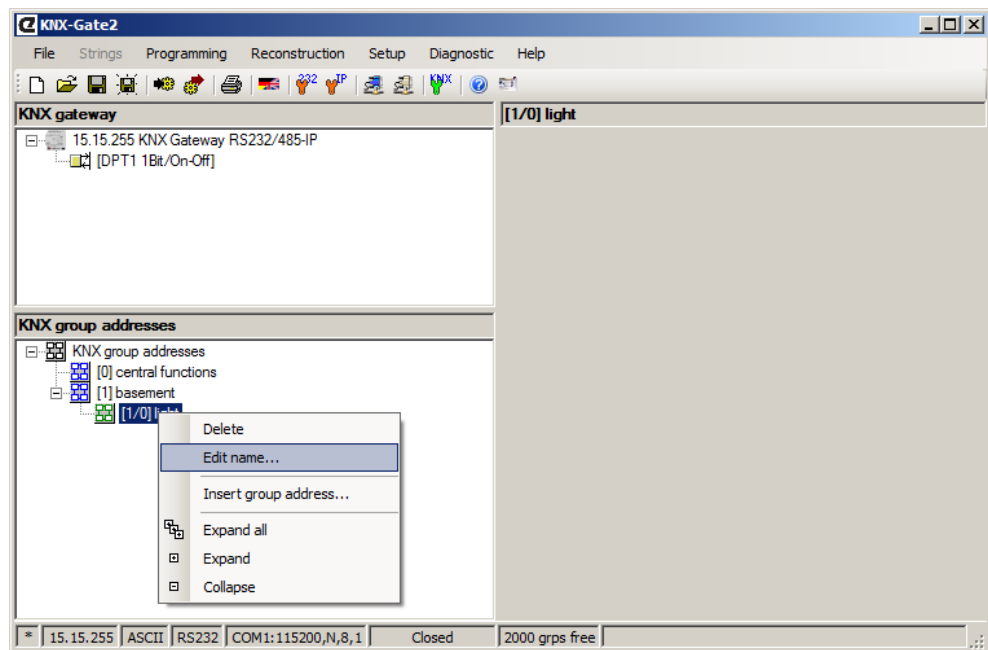


Fig. 5.4-5: Create group addresses

Within the tree topology, new group addresses can be created via the right mouse button commands *Add main group*, *Add middle group*, *Add group addresses*. When a free group address structure is used, instead of main groups or middle groups address areas are added.



**Fig. 5.4-6: Edit group addresses**

Within the group address tree, subsequently the name and *Read at Restart* can be changed via the context menu.

Group addresses, for which the reading function is activated, are displayed in the tree topology with a filled symbol.

If group addresses are to be read, within the ETS project the read flags and the sending group addresses of the relevant devices have to be observed.

#### 5.4.4 Connect group addresses with objects

To connect the KNX objects with the group addresses, the subgroups are pulled onto the corresponding KNX objects. Here, each group can be connected with one object only.

## 5.5 Operation mode Strings

When in creation of a new project the *Protocol for String-processing* has been activated, the menu *Strings* is released with further commands.

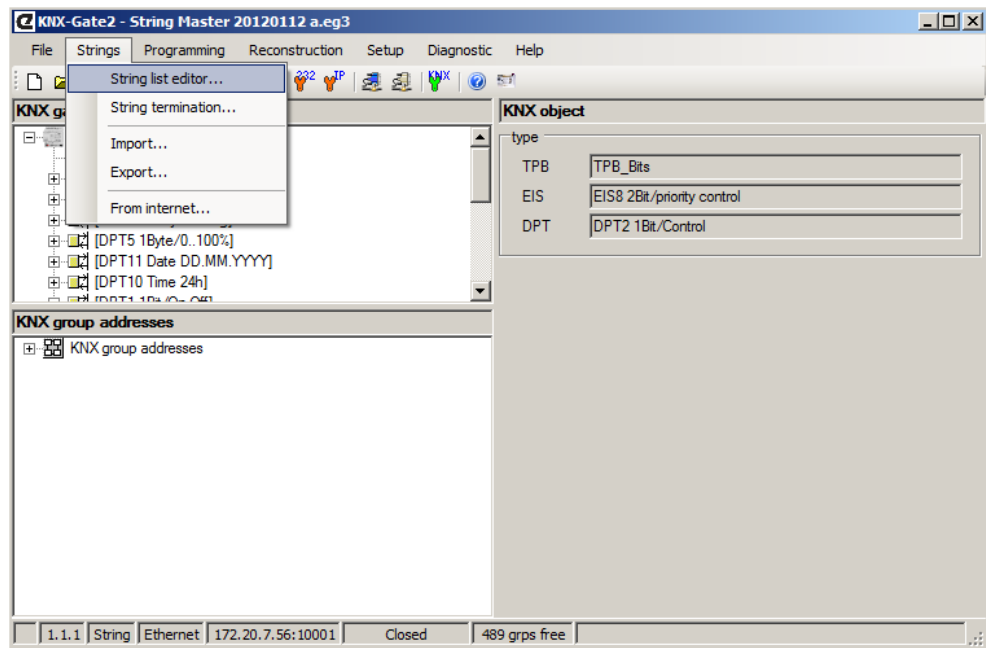


Fig. 5.5-1: Menu strings

Matching the connected subsystem, in the dialogue *String-End* the criterion indicating the end of a transmitted string can be set. Often the ASCII-sign <CR> (carriage return) is used with the hexadecimal value 0D (decimal 13). Some systems expect the combination of both signs instead <CR> <LF> (carriage return, line feed) with the hexadecimal values 0D 0A (decimal 13, 10). If these values can also appear in the transmitted strings, it is not possible to use them to mark the end of a string. In this case you have the possibility to use a *Timeout*.

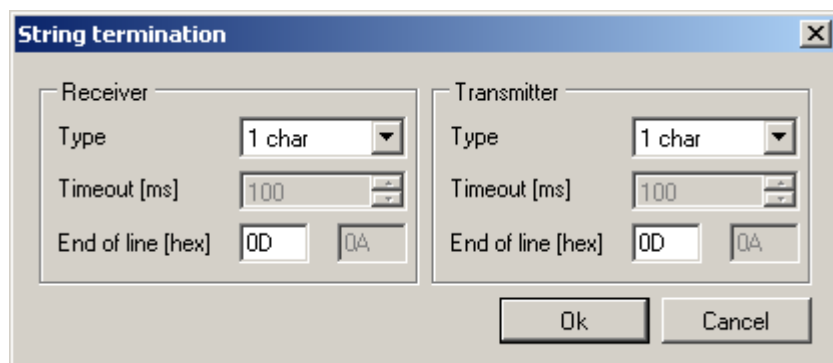


Fig. 5.5-2: Set string end

You can look up in the documentation of the subsystem which criterion is used.

### 5.5.1 Creating Strings

In the dialogue *Gateway string list* all strings of the project are administered. The *Name* helps identifying hexadecimal strings. With the buttons *Text* and *Hex* the corresponding dialogues for the creation of new strings are opened. By means of the button *Edit* existing strings can be edited. Strings that are no longer needed can be deleted from the string list with the *Delete* button.

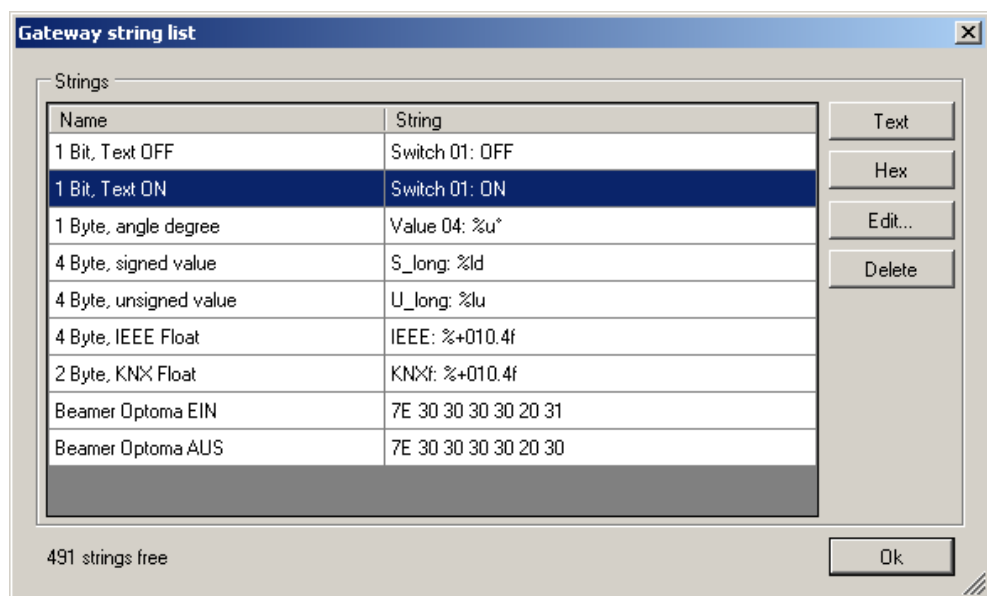


Fig. 5.5-3: String list

#### 5.5.1.1 Text strings

Text strings can be used for two fields of functions. In the first case, on receipt of a defined KNX telegram a constant text is sent or on receipt of a constant text a defined KNX telegram is sent. In the second case the text consists of a constant and a variable part and the value of the KNX telegram can be entered into this variable part.

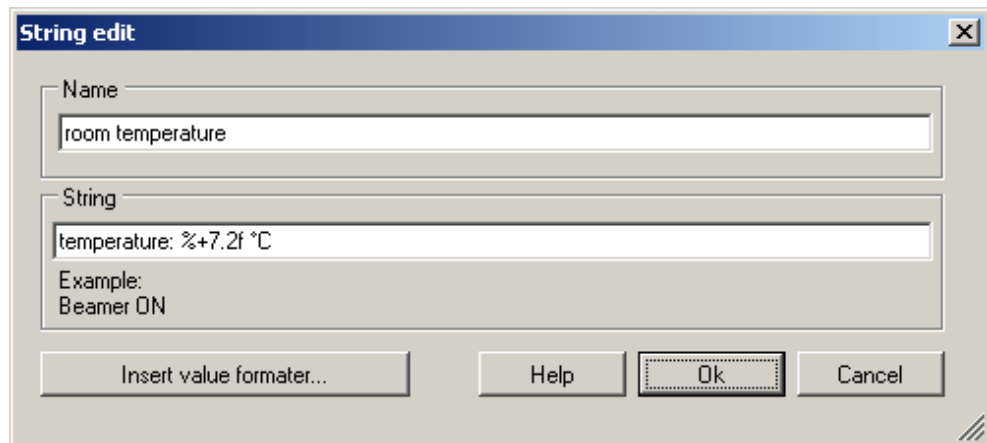


Fig. 5.5-4: Create string

If the value of the KNX telegram is to be transmitted in the string or a value received by the subsystem is to be transmitted to the KNX, a variable is entered in the string. (e.g.%4d). For conversion, the formats of the printf()-command of the programming language C are used. Typical formats with examples are described in the appendix.

Entering the format of figures with floating decimal point is simplified via the dialogue *Create format*, which is opened via the button *Add value format*.

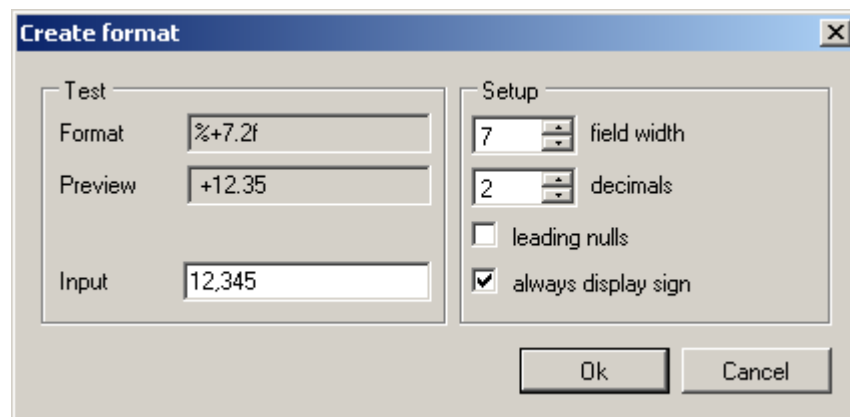


Fig. 5.5-5: Format setting for figures with floating decimal point

### 5.5.1.2 Hexadecimal Strings

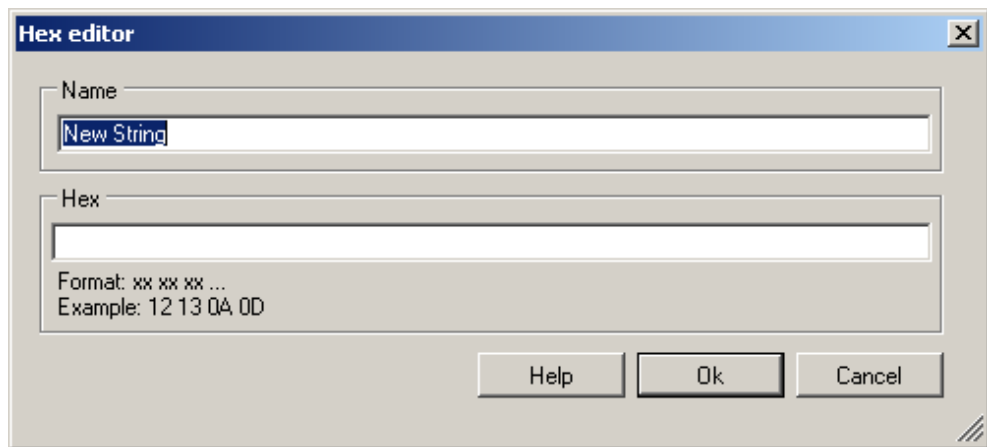


Fig. 5.5-6: Create hexadecimal String

Hexadecimal strings are used, for instance, in media technology to control beamers or similar devices. Corresponding to the documentation of these devices, the hexadecimal values are separated by blanks (e. g.: 12 13 0D 0A) when entered.

### 5.5.2 Send and receive Strings

When a group address has been assigned to an object, this group address can be selected in the object tree. By means of the context menu in the lists *Send Strings* or *Receive Strings* a string out of the *String list* can be added.

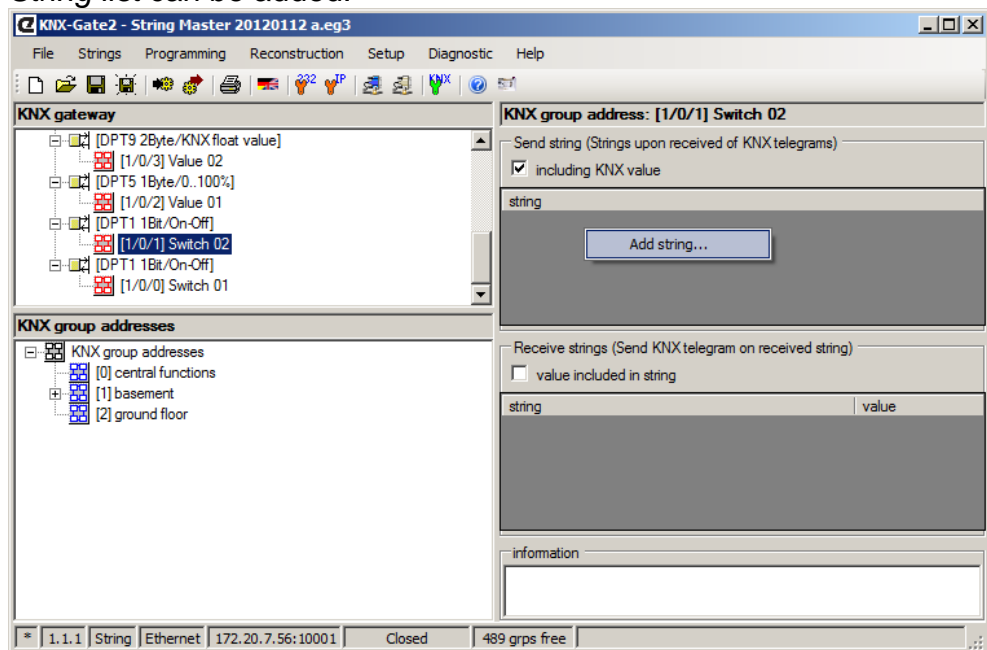


Fig. 5.5-7: Add String

When the checkboxes *Insert KNX value into string* or *Take value out of string* are not marked, you can indicate in the dialogue *change value*, with which object value the string is to be sent or which object

value is to be sent on receipt of this string. In this dialogue only values can be entered which match the type of the KNX object.

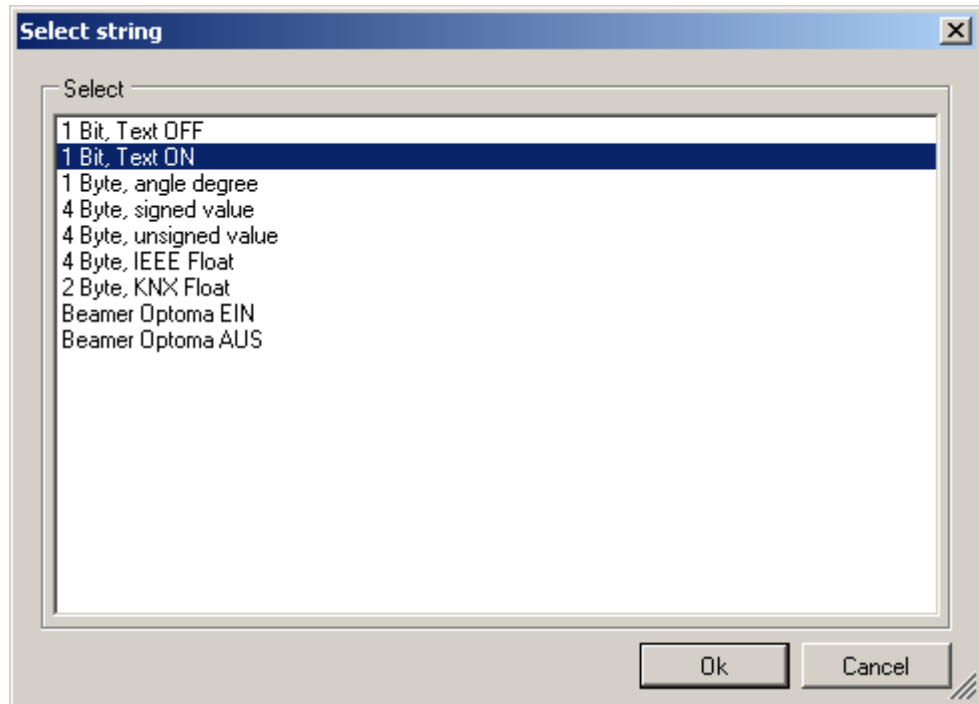


Fig. 5.5-8: Select Send or receive string

### 5.5.3 Export and import Strings

For easier projecting of similar string lists, they can be exported from an existing project and imported into other projects.

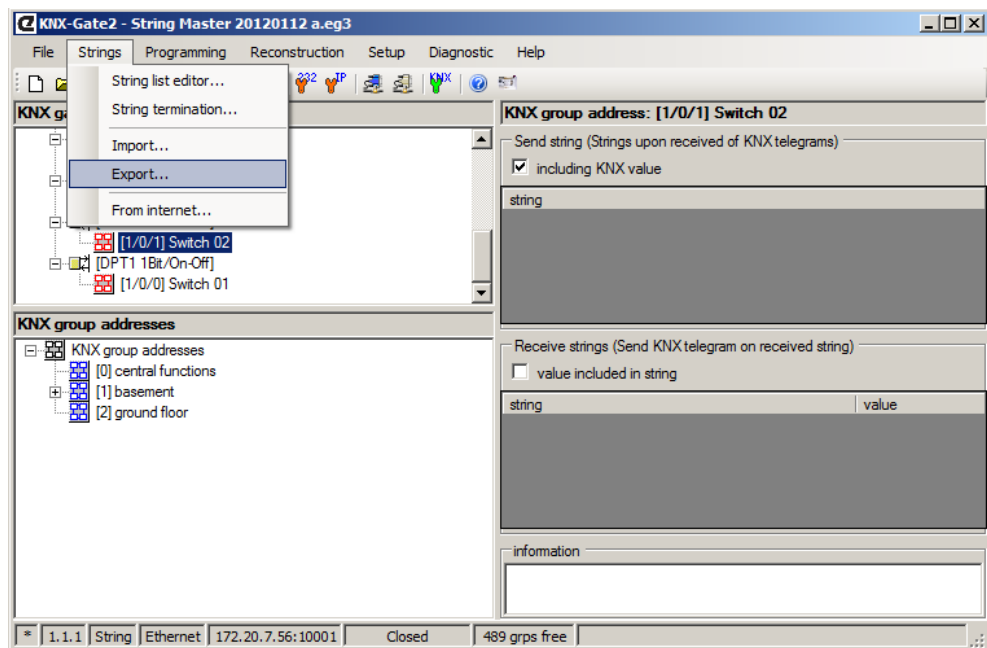


Fig. 5.5-9: Export String list

## 5.6 Documentation

To finish projecting, via *File-> Print...* the current state can be printed.

## 5.7 Commissioning

Independent of the projected operation interface, commissioning of the gateway can be effected via the RS232-or via the Ethernet/IP interface.

The currently set commissioning interface of the PC is displayed in the status bar of the main window.

### 5.7.1 Choice of interface

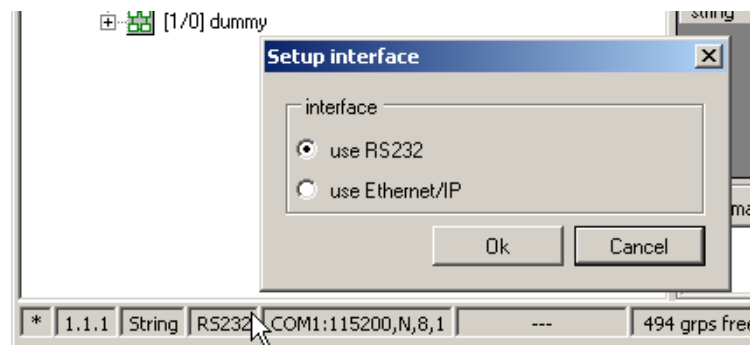


Fig. 5.7-1: Choice of interface

The dialogue for choice of the commissioning interface can be opened via the button in the tool bar or via the display in the status bar.

### 5.7.2 IP settings

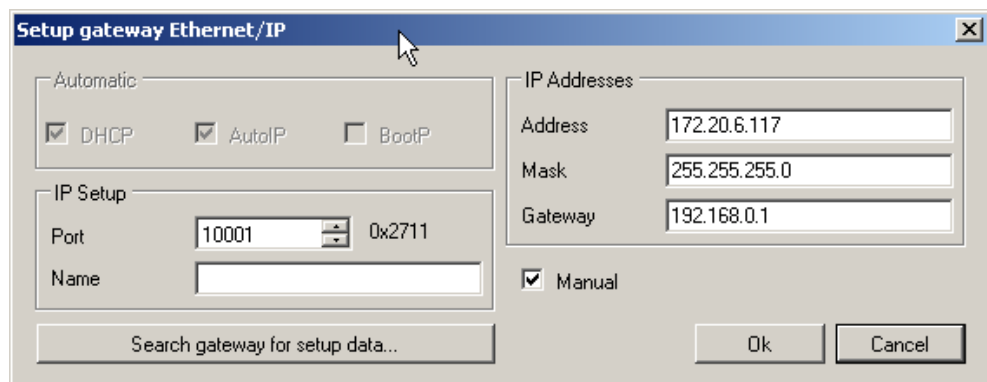


Fig. 5.7-2: IP parameters of the gateway

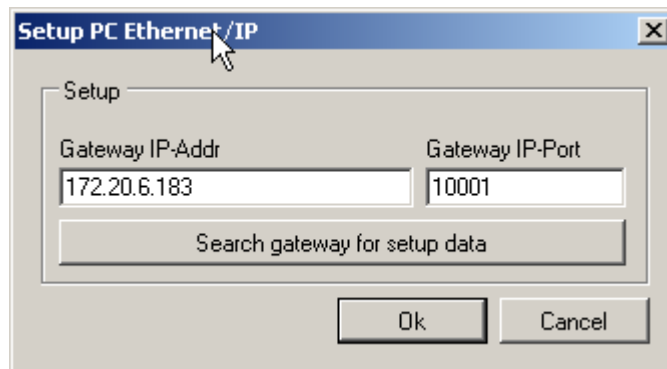


Fig. 5.7-3: IP parameters of the PC

If commissioning of the gateway is to be effected via the Ethernet/IP interface, it can either be connected as part of the whole network or directly to the PC. For direct connection to the PC it might be necessary to use a crosslink network cable.

Unless there are any restrictions made by the network administrator, in most cases the gateway can automatically obtain the IP address. The IP address is assigned automatically after the supply voltage of the gateway has been switched on. If possible, the gateway tries to get its IP-address from a DHCP-Server or a BootP-Server. If this is not possible, because the gateway is directly connected to a PC, it falls back to the Auto-IP-process. Here it gets an IP-address between 169.254.1.0 and 169.254.254.255 by means of a random number generator. If necessary, the network interface of the PC must be configured accordingly.

Number 10001 is preset as port address of the gateway. If access via internet is intended, it is recommended to check the firewall settings.

Communication on the Ethernet is effected via the unambiguous MAC address of the devices. For actual commissioning, the gateway can be identified unambiguously via its MAC address. Please see the label with the MAC address on the side of the device. In the dialogues of the settings of the IP interfaces of the PC and the gateway you can set the option that the PC looks for suitable gateways. All IP addresses found are then listed together with the MAC addresses.

Depending on the operating system it is possible that the firmware version cannot be displayed in the dialogue KNX Gateway RS232/485-IP, if network access is effected via a proxy server. In this case it might be necessary to make a note in the network settings of the PC that access to the IP address of the gateway is not to be effected via the proxy server.

### 5.7.3 RS232-settings

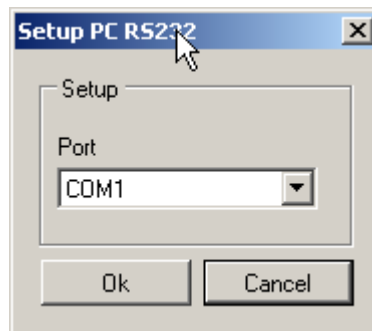


Fig. 5.7-4: RS232 parameters of the PC

The parameters of the RS232 interface in the gateway are firmly set. In the dialogue *settings PC RS232* the PC interface used can be chosen. No further settings are required.

### 5.7.4 Download

The download dialogue can be opened via the menu *Commissioning* or via the symbol in the tool bar.

## 5.8 Reconstruction

For reconstruction the connection PC-gateway is set in the same way as in commissioning.

In order to be able to identify the gateway unanimously via the Ethernet/IP interface during reconstruction, it is recommended to connect the gateway with the PC direct (if required via a crosslink lead) or to note down the MAC address on the label at the side of the gateway.

By means of a reconstruction, all functionally relevant data can be read out of the gateway.

## 5.9 Diagnosis

For diagnosis of the gateway two commands are available.

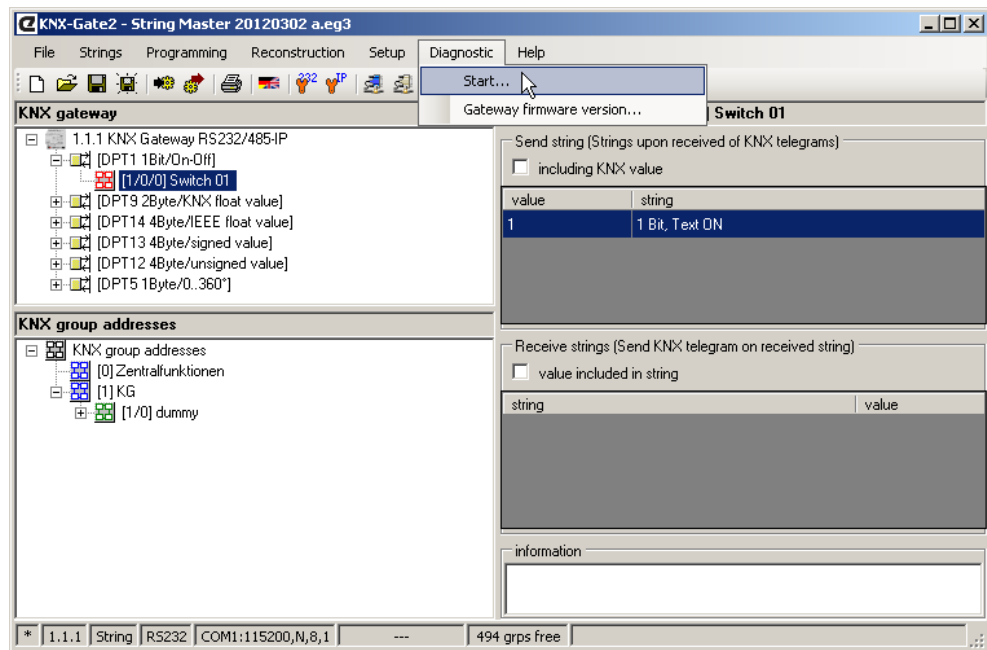


Fig. 5.9-1: Diagnosis functions

The command *Diagnosis -> Start...* opens the diagnosis interface set in project planning of the gateway and in the dialogue *Diagnosis* it then shows the information given out by the gateway.

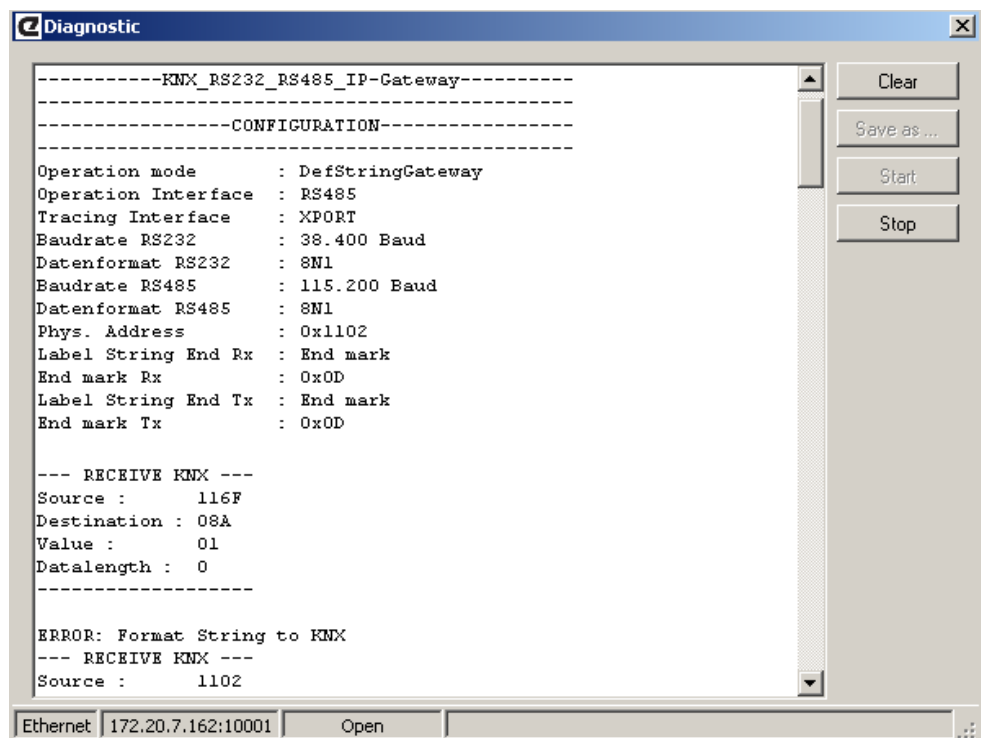
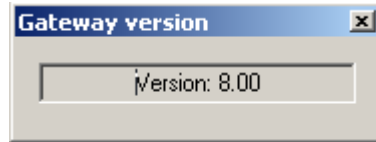


Fig 5.9-2: Diagnostic information

This information includes the basic configuration of the interfaces and a record of received KNX Telegrams and communication problems.

The command *Diagnose* -> Gateway firmware version inquires the version of the internal device software via the commissioning interface.



**Fig 5.9-3: Gateway firmware version**

If this connection to the gateway is terminated, the gateway runs a restart.

## 6 Appendix

### 6.1 ASCII protocol

Generally, data exchange between external device and gateway is based on a firm telegram structure. The telegrams only contain data in ASCII format, in order to avoid errors in synchronisation and data transmission.

A telegram must not exceed a maximum length of 40 bytes.

#### 6.1.1 Spontaneous sending / polling

Depending on the parameterisation, the gateway can send telegrams to the subsystem automatically or only on demand of the subsystem.

Each telegram received must be acknowledged by the receiver within a timeout-time defined in the dialogue *Setup Gateway KNX* either by an acknowledgement signal (ACK or NAK) or a response telegram.

#### 6.1.2 Telegram structure

A telegram has got the following structure:

<STX>	ASCII sign "Start of Text" (\$02)
Function byte	Indicates number and meaning of the following data. A list of possible function bytes and their meaning is given later on.
Data bytes	Data contents of the telegram.
Checksum	To check the telegram for faultlessness.
<CR>	ASCII sign "Carriage Return" (\$0D)

The function byte, the data bytes and the checksum are transmitted in ASCII format, i.e. one data byte is displayed with two ASCII signs (characters). Example: function byte 04 (inquiry gateway): the signs '0' (Hex. \$30) and '4' (Hex.: \$34) are transmitted.

The checksum is calculated acc. to the following code:

8-bit addition of raw data without carried forward, starting with the function byte and all subsequent data bytes with final bit by bit negation.

Example: <STX>0341230395<CR>

$\$03 + \$41 + \$23 + \$03 = \$6A$

Negation of the byte \$6A equals checksum \$95

### 6.1.3 Version inquiry of the gateway firmware

With this function, the external device can inquire the software version of the gateway firmware.

Telegram contents  
Function byte: \$01  
Data contents: ---

Response  
Function byte: \$81  
Data contents: 2 byte version number

Example  
ext. device: <STX>01FE<CR>  
Gateway: <STX>81080076<CR>  
ext. device: <ACK>

The first byte of the version number indicates the figure in front of the point, the second byte indicates the figure behind the point in hexadecimal form (in the example: Version 8.00).

### 6.1.4 Inquiry on incoming bus telegrams

This function inquires with the gateway whether a bus telegram has been received. If data exist, the gateway sends back the relevant data of a bus telegram. Otherwise ACK (\$06) is sent as answer. An error message can also be sent as response by the gateway. This function should be called up cyclically by the external device to get the current state of the projected group addresses.

Telegram contents  
Function byte: \$04  
Data contents: ---

Response  
Function byte: \$FC  
Data contents: 2 byte group address  
n byte data  
or acknowledgement: <ACK>  
or error message

Example  
ext. device: <STX>04FB<CR>  
Gateway: <STX>FC080101F9<CR>  
ext. device: <ACK>  
Gateway: <ACK>

### 6.1.5 Reset Gateway

A telegram with this function byte releases a reset of the gateway which thereupon, depending on its parameterisation, makes a status inquiry on the KNX.

Telegram contents

Function byte: \$08  
Data contents: ---

Response

<ACK>

Example

ext. device: <STX>08F7<CR>  
Gateway: <ACK>

### 6.1.6 Describe KNX group address

By means of this function, the external device sends data to a KNX object. The group address of the KNX telegram to be generated must be programmed in the gateway. Otherwise the gateway answers with "NAK", for the telegram could not be processed. The telegram data consist of the 16bit group address, the transmission priority and the data contents to be sent. The number of KNX data bytes must be identical with the data type programmed, otherwise the telegram is answered with "NAK". To enter the transmission priority either the value \$0C for low priority or \$04 for high priority must be entered.

Telegram contents

Function byte: \$0B  
Data contents: 2 byte group address  
1 byte transmission priority  
n bytes KNX data

Response

<ACK>

Example

ext. device: <STX>0B1C050C01C6<CR>  
Gateway: <ACK>

The sample telegram sends a KNX telegram with data contents \$01 and low priority to the group address 3/4/5.

With user data of a size of less than 1 byte, the highest value bits that are not used must be set to 0!

After corresponding inquiry, the acknowledgement (acknowledge) of the telegram from a KNX device with group address 3/4/5 has got a

response telegram from the gateway with the function "receive KNX telegram".

### 6.1.7 Read out KNX group address

With this function, the external device can inquire data from a KNX object. The group address of the KNX telegram to be generated must be programmed in the gateway. Otherwise the gateway answers with "NAK", for the telegram could not be processed. The data contents of this telegram consists of the group address to be read.

Telegram contents

Function byte:     \$0C  
Data contents:     2 byte group address

Response

<ACK>

Example

ext. device:       <STX>0C1C05D2<CR>  
Gateway:           <ACK>

The sample telegram makes the gateway send a „read“-command for the group address 3/4/5. Whether and how many members answer depends on the projecting of the KNX system.

Depending on the parameterisation, an answer from KNX devices to this inquiry is not passed on by the gateway automatically, but can only be called up by the function „inquiry on incoming bus telegrams“!

### 6.1.8 Error message

With this function the gateway indicates that an error has incurred. The data contents of the telegram consists of a 16-bit-value, of which each individual bit indicates a certain error.

Telegram contents

Function byte:     \$FE  
Data contents:     2 byte error value

Example

Gateway:           <STX>FE0009F8<CR>  
ext. device:       <ACK>

The sample telegram gives the error value \$0009. In binary output of the number (%0000 0000 0000 1001) the bits 0 and 3 are set. This means that two errors have incurred.

For the telegram function error message the following error bits are defined:

- Bit 0: KNX not acknowledge  
On an addressed group address no device has answered.
- Bit 3: BA-Busy  
The bus coupling unit is not ready-to-receive
- Bit 4: BA-Error  
Internal error of the bus coupling unit.
- Bit 7: Tx-Overflow  
The transmission memory of the gateway has overflowed
- Bit 8: Rx-Error  
Telegram was received whose length does not correspond to the projected length for the group address.
- Bit 11: FLASH CRC-Error  
The programmed data in the gateway have been deleted.
- Bit 13: BA-Layer  
The bus coupling unit could not be found on the layer suitable for gateway operation. This error is caused e.g. by a power failure on the KNX.

## 6.2 Formatting of strings

### 6.2.1 General description

In creation of the KNX objects different data point types can be selected. Each data point type defines a width (number of bits or bytes) and a certain coding.

For simple data point types with a limited number of possible values (e. g. a 1 bit data point type with only two values 0 and 1 representing switch on and switch off) it is easy to create a unique string for each value.

For other data point types with a large number of possible values it is better to insert the KNX value into a standard string or to extract the KNX value from a received string.

Depending on the data point type of a KNX object, different formats are suitable for the display of the values, which are called format strings. In large, the structure of the format strings corresponds to the information which is also used in programming language C.

The format string %f is used for numeric data point types. If necessary several options can be added between the % and the f.

%f	floating decimal figure
+	always display sign (+ or -)
0	display leading zeros
n.m	number of digits to be displayed and the decimal places

Other possible format strings are

%c	single ASCII character
%s	sign chain.
%%	per cent sign
.*f	ignore a number in a received string

## 6.2.2 KNX data point types

The KNX-Gate2 can use the following data point types.

KNX-DPT	EIS	Width	Type	Range
1	1, 7	1 bit	On/Off, Up/Down	0/1
2	8	2 bit	1bit control / priority	0 ... 3
3	2	4 bit	3bit control / dimming	0...15
4	13	1 byte	character	1 x ISO 8859-1
6	14	1 byte	signed	-128 .. 127
5	14	1 byte	unsigned	0...255
5	6	1 byte	unsigned	0...100%
5	6	1 byte	unsigned	0...360°
7	10	2 byte	unsigned Int	0 .. 65,535
8	10	2 byte	signed Int	-32,768 .. 32,767
9	5	2 byte	KNX float	1 bit sign 4 bit exponent 11 bit mantissa
10	3	3 byte	time	24h
10	3	3 byte	time	12h
11	4	3 byte	date	tt.mm.jjjj
11	4	3 byte	date	tt.mm.jj
11	4	3 byte	date	mm.tt.jjjj
11	4	3 byte	date	mm.tt.jj
12	11	4 byte	unsigned long	0 ... 4294967295
13	11	4 byte	signed long	-2147483648 ... 2147483647
14	9	4 byte	IEEE float	1 bit sign 8 bit exponent 23 bit mantissa
16	15	14 byte	KNX string	14 x ISO-8859-1

### 6.2.2.1 Simple data point types (1 bit, 2 bits, 4 bits)

For simple data point types unique strings can be defined for each value.

1 bit:

value	example
0	Beamer ON
1	Beamer OFF
0	window closed
1	window open

2 bits:

value	example
0	no priority, OFF
1	no priority, ON
2	priority, OFF
3	priority, ON

4 bits:

value	example
0	STOP
1	darker 1 step
2	darker 2 steps
3	darker 4 steps
4	darker 8 steps
5	darker 16 steps
6	darker 32 steps
7	darker 64 steps
8	STOP
9	brighter 1 step
10	brighter 2 steps
11	brighter 4 steps
12	brighter 8 steps
13	brighter 16 steps
14	brighter 32 steps
15	brighter 64 steps

### 6.2.2.2 Integer types

For integer types the format string depend on the possible range of values. Leading zeros can help to create table like columns

Format string	output
fan level %1.0f	fan level 0 fan level 3
scene %02.0f	scene 05 scene 17

### 6.2.2.3 Floating point types

Floating point data point types can have positive and negative values and varying resolution.

Format string	output
Temperature %5.1f °C	Temperature 19.8 °C
	Temperature 7.3 °C
Offset %+06.2f K	Offset +00.25 K
	Offset -09.00 K
Humidity %5.1 %%	Humidity 25.4 %

### 6.2.2.4 Date and time

Data point types for time and date combine different values. Therefore the appropriate number of format strings are used.

Format string	output
Date %02.0f.%02.0f.%04.0f	Date 29.02.2012
	Date 01.11.2012
Date %02.0f.%02.0f.%02.0f	Date 29.02.12

### 6.2.2.5 Special types

In addition to the simple enumeration data point types and the numeric types, special objects can also be used for single ASCII characters and KNX strings.

For these data point types the format strings %c (ASCII characters) and %s (KNX string) are available. They can only be used to send data from KNX to the subsystem.

Certain subsystems combine different values in one string. To extract one of these values from a received string and to transmit it as a single KNX value, the other values must be masked with the format string %\*f.

Received string	Formatstring	Output (KNX)
22.3_27.5_35.8	%*f_%f_%*f	27.5