Automations- und Steuerungstechnik GmbH

Product description

EIB AT

EIB-ASCII-Terminal

The plain text serial interface to the EIB

Order code: E001-H003011

IMPORTANT-READ CAREFULLY:

This b+b End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and b+b Automations- und Steuerungstechnik GmbH, for the software product identified above, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA.

SOFTWARE PRODUCT LICENSE

1. COPYRIGHT TREATIES

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

2. GRANT OF LICENSE.

This EULA grants you the following rights:
   a. Software Product. b+b grants to you as an individual,
      a personal, nonexclusive license to make and use copies of the
      SOFTWARE for the sole purposes of using the SOFTWARE´s
      functionality.
   b. Storage/Network Use. You may also store or install a copy of the
      SOFTWARE PRODUCT on a storage device, such as a network server, used
      only to install or run the SOFTWARE PRODUCT on your other computers
      over an internal network; however, you must acquire and dedicate a
      license for each separate computer on which the SOFTWARE PRODUCT is
      installed or run from the storage device. A license for the SOFTWARE
      PRODUCT may not be shared or used concurrently on different computers.
   c. Electronic Documents. Solely with respect to electronic documents
      included with the SOFTWARE PRODUCT, you may make an unlimited number
      of copies (either in hardcopy or electronic form), provided that such
      copies shall be used only for internal purposes and are not
      republished or distributed to any third party.

3. DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS.

   a. Limitations on Reverse Engineering, Decompilation, and Disassembly.
      You may not reverse engineer, decompile, or disassemble the SOFTWARE
      PRODUCT.
   b. Separation of Components. The SOFTWARE PRODUCT is licensed as a
      single product. Its component parts may not be separated for use on
      more than one computer.
   c. Changing documentations. You may not make changes to the documentation
      of the SOFTWARE PRODUCT.
   d. Termination. Without prejudice to any other rights, b+b
      may terminate this EULA if you fail to comply with the terms and
      conditions of this EULA. In such event, you must destroy all copies
      of the SOFTWARE PRODUCT and all of its component parts.

4. COPYRIGHT.

All title and copyrights in and to the SOFTWARE PRODUCT (including but not
limited to any images, photographs, animations, video, audio, music, text,
and "applets" incorporated into the SOFTWARE PRODUCT), the accompanying printed
materials, and any copies of the SOFTWARE PRODUCT are owned by b+b
or its suppliers. The SOFTWARE PRODUCT is protected by copyright laws and
international treaty provisions. Therefore, you must treat the SOFTWARE PRODUCT
like any other copyrighted material except that you may install the SOFTWARE
PRODUCT on a single computer provided you keep the original solely for backup
or archival purposes. You may not copy the printed materials accompanying the
SOFTWARE PRODUCT.

5. LIMITED WARRANTY

Except with respect to the REDISTRIBUTABLES, which are provided "as is,"
without warranty of any kind, b+b warrants that (a) the
SOFTWARE PRODUCT will perform substantially in accordance with the accompanying
written materials for a period of ninety (90) days from the date of receipt,
and (b) any hardware accompanying the SOFTWARE PRODUCT will be free from defects
in materials and workmanship under normal use and service for a period of
one (1) year from the date of receipt.

6. CUSTOMER REMEDIES.

b+b´s entire liability and your exclusive remedy shall be, either
(a) return of the price paid, or (b) repair or replacement of the SOFTWARE
PRODUCT or hardware that does not meet b+b Limited Warranty.
This Limited Warranty is void if failure of the SOFTWARE PRODUCT or hardware
has resulted from accident, abuse, or misapplication.

NO OTHER WARRANTIES: TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE
LAW, b+b DISCLAIMS ALL OTHER WARRANTIES

NO LIABILITY FOR CONSEQUENTIAL DAMAGES: TO THE MAXIMUM EXTENT
PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL b+b
OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL,
INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING,
WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS PROFITS,
BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY
OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY
TO USE THE SOFTWARE OR HARDWARE PRODUCT, EVEN IF b+b
HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Life support:
These products are not designed for use in life support appliances, devices or
systems where malfunction of these products can reasonably be expected to result
in personal injury. B+b customers using or selling these products
for use in such applications do so at their own risk and agree to fully
indemnify b+b for any damages resulting from such application.

Microsoft and Windows are trademarks of Microsoft Corporation.

This handbook describes also functions, which are options.
Only qualified persons are allowed to install our units.

Softwaredesign und Coding: Volker Knapp, Hartmut Zander, Peter Bernert, Michael Weis
Documentation: Frank Schlaps, Volker Knapp, Michael Weis
Editor: Frank Schlaps

b+b Automations- und Steuerungstechnik GmbH
Klingenweg 17
64385 Reichelsheim
Tel: +496164 / 912057
e-mail: support@bb-steuerungstechnik.de
internet: http://bb-steuerungstechnik.de

Contents

Product Highlights EIB-ASCII-Terminal (EIB-AT)

- Easiest access to the EIB
- Standard-ASCII-Code
- Supports all 32767 group addresses simultanously
- Supports all non-structured EIS-types
- Simple setup using a terminal program
- built in german and english help texts
- Fully transparent even at 100% bus load
- Built in filter- and EIS-type tables
- All values read and writeable using plain text
- Simple integration of the EIB into own systems (PC,uC…)
- Built in bus coupling unit
- Standard-RS232-Interface, no system drivers required
- Small housing (67*92*26mm)

Fields of application

- Coupling to any EIB-world over ASCII-protocol without control sign
- Access to EIB from own programs (no PC necessary)
- Connection of own systems to the EIB

Contents of delivery

- EIB-ASCII-Terminal
- Cable set
- Parametrization software
- Documentation

Introduction

The EIB-ASCII-Terminal (EIB-AT) is a serial interface to the EIB with an integrated twisted-pair bus coupling unit. The EIB-AT allows an operating system independant connection of the EIB to your own system. The communication is done in human readable text without time critical handshake signals.

Functional description (Rev 4.03)

EIB-Interface:

2-Wire twisted pair with integrated bus coupling unit
supported group addresses: 32765

Serial interface:

Host interface RS232
1 Startbit,8 Databits, 1 Stopbit, no Parity
Handshake RTS/CTS (user selectable option)
Baudrates: 9600, 19200, 38400 Baud, selectable by pushbutton, stored nonvolatile

Inportant Hint::
If you are using 9600 or. 19200Baud you may expiriencedata loss on high busload, since there are more characters to transfer to the Host as are received from the EIB. Therefore we strongly recommend to use 38400 Baud transfer speed. At this baudrate the transfer time to the host is shorter as the correlated EIB telegram.
Please pay also attention not to block the device by deasserting the RTS signal

ASCII-Protocol description:

The communication between host and EIB-AT uses readable text only. The device neither sends nor accepts characters below 20Hex (ASCII Blank).
The only exception to this rule is the carriage return char(0d Hexadecimal), shortened called "cr" in the following text.
The cr is used to signal the end of a transmission. Characters sent from the host are not echoed by the device.
All command related characters are treated as uppercase, the means the device is "case insensitive"

ASCII-Commands:

ASCII-Command: request help text

Command:	"?"
Purpose:	Request of the help text
Description:	The device sends a short version of this document to the host
Remark:	During this command the EIB Communication is stopped!
Reply from the Device:	Helptext

ASCII-Command: Version request

Command:	"?V"
Purpose:	Query the firmware version
Description:	The EIB-AT sends its firmware revision and serialnumber
Reply from the device:	"EIB_Terminal Vn.nn SN:xxxxxxxx" n.nn = Firmwareversion, xxxxxxxx = Serialnumber
Example reply:	"EIB_Terminal V4.05 SN:3709651"

ASCII-Command: Physical address request

Command:	"?P"
Purpose:	Query the physical address of the device
Description:	The EIB-AT sends ist physical address
Reply from the device:	"Phys. Addr=nn.nn.nnn" or "Phys. Addr=xxxx" if the hexadecimal option is activated.
Example reply:	Phys. Addr=01.01.254

ASCII-Command: Setup the physical address

Command:	"P:PA"
Purpose:	Setup the physical address of the device and store it non-volatile
Description:	PA may be given as AA.LL.DDD or Hexadecimal as "xALDD"

Telegram contents:	"P:AA.LL.DDD" or "P:xALDD", where: A=Area, L=Line, D=device
Reply from the device:	"Phys. Addr=nn.nn.nnn" or "Phys. Addr=xxxx" if the hexadecimal option is activated.
Remark:	Device must be nonzero, address 15.15.255 is invalid
Error messages:	"!Bad Command Format" if the command is badly formatted. "!Bad value" if an illegal address given.
Example:	P:1.1.250 sets the physical address of the EIB-AT to 1.1.250

ASCII-Command: Get the current options

Command:	"?O"
Purpose:	Request the status of the various options
Reply:	"OPTIONS: Dw Ew Gw Hw Nw Qw Rw Sw Vw Ww Xw"
Remark:	w="+" option activated, w="-" option deactivated During this command the EIB Communication is stopped!
Possible Options:	

     "OVw"   Verbose mode
            If Verbose mode is activated the device replies to commands with an answer,either „OK" or an error description.

     "OEw"   Echo of own write telegrams
            If this option is activated own write telegrams are sent back to the host after received „back" from the bus. Since this filtering is done by comparing physical source address please pay attention to assign physical addresses only once. The echo telegram will be processed by the normal receive filter(described later).

     "OQw"   Report read telegrams to the host
            If this option is activated received read telegrams of enabled group addresses are sent to the host. The format is similar to an value telegram, instead of „=value" read telegrams are signaled with „*".

"OXw"   Hexadecimal mode
If this option is activated the addresses are transmitted as „XXXX",
that means as 4 hexadecimal digits.
The components are divided as follows:
Source address(physical address): "ALDD"
4 Bit Area address, 4 Bit Line address, 8 Bit Device address
Target address(group address):"0HHHHMMMUUUUUUUU"
To explain this we need to use the binary description:
The most significant Bit is always zero, followed by 4 bits for the
maingroup, 3 bits for the middlegroup and 8 bits for the subgroup.
If this option is deactivated the source addresses are formatted as
AA.LL.DDD and target addresses as HH/M/SSS
A=Area,L=Line,D=Device,H=Maingroup,M=Midgroup,S=Subgroup.

"ORw"   Allow read telegrams to disabled group addresses
If activated the device allows to send read telegrams to disabled
group addresses, else the read request is dropped.

"OSw"   Send source address to the host
If enabled the source address of each telegram is inserted in front of
the destination address. This causes more data transfer and requires
38400Baud!

"OHw"   RTS/CTS Handshake
If activated the device sends data to the host only if the RTS line is
activated. The CTS line signals that data may be transmitted by the
host.

"ONw"   Telegram numbering
If activated the telegrams to the host are prefixed with an 3 digit
decimal or
2 digit hexadecimal number. This number is incremented by 1 after
each transfer and set to zero again when reached 255/0xff   .

"OGw"   Language selection
OG+ selects german, OG- englisch

"ODw"   Decimal sign selection
OD+ selects the Dot as decimal delimiter,
OD- selects comma.

"OWw"   Wait message
If activated the device sends an "please wait" message to the
host on lengthy operations, like setting up attribute ranges of the
ERASE! Command.

ASCII-Command: Converting of group addresses

Command:	"Cga"
Purpose:	Useful utility function to convert group address formats
Remark:	Causes no read or write action to the bus
Example1:	An hexadecimal groupaddress is given, the decimal format is wanted: Host sends: CX5F03 The device answers with X5F03=11/7/003
Example2:	The other way round: Host sends: C11/7/3 The device answers with 11/7/003=X5F03

ASCII-Command: Get group address attributes

Command:	"Gga"
Purpose:	Get the remanent attributes of group address „ga".
Remarks:	Causes no read or write action to the bus "ga" may be specified as "HH/M/SSS" or "xNNNN". While processing this command the EIB communication is stopped!
Reply:	"Gga:n,f" n = dataformat used for this ga, 0..21 (see data formats!) f=receive enable, "E"/"D" E=Enabled, D=Disabled
Example:	G0/0/1 requests the attributes for group address 0/0/1.
Example Reply:	00/0/001:1,E that is: Format 1, receive enabled

ASCII-Command: Get group address range attributes

Command:	"Gga1-ga2"
Purpose:	Get the remanent attributes of group address „ga1" upto „ga2".
Remarks:	Causes no read or write action to the bus "ga" may be specified as "HH/M/SSS" or "xNNNN". While processing this command the EIB communication is stopped!
Reply:	"Gga:n,f" n = dataformat used for this ga, 0..21 (see data formats!) f=receive enable, "E"/"D" E=Enabled, D=Disabled The whole specified range is sent to the host, one line for each ga.

Possible error messages:
If option "V-" is set: none

IF "V+" is set:
"!Bad Command Format"
"!Bad Group Address"

Example:	G0/0/1-0/0/4 requests the attributes of the group addresses 0/0/1 to 0/0/4.
Example reply:	00/0/001:1,E that is format 1, receive enabled
	00/0/002:1,E that is format 1, receive enabled
	00/0/003:1,E that is format 1, receive enabled
	00/0/004:1,E that is format 1, receive enabled

ASCII-Command: Setup group address attributes

Commands:	"Sga:n" "Sga,f" "Sga:n,f"
Purpose:	Setup attributes for group address „ga" and store settings nonvolatile.
Remarks:	"ga" may be specified as "HH/M/SSS" or "xNNNN". n = dataformat used for this ga, 0..21 (see data formats!) f=receive enable, "E"/"D" E=Enabled, D=Disabled While processing this command the EIB communication is stopped!
Reply:	If option "V-" is set: none If "V+" is set: "Ok".
Possible error messages:	
	If option "V-" is set: none IF "V+" is set: "!Bad Command Format" "!Bad Group Address"
Example:	S1/0/1:1,E defines the group address 1/0/1 as 1-6 Bit value and enables the receive This attributes are used i.e. for ON/OFF telegrams.

ASCII-Command: Setup group address range attributes

Commands:	"Sga1-ga2:n" "Sga1-ga2,f" "Sga1-ga2:n,f"
Purpose:	Setup attributes for group address „ga1" upto "ga2" and store settings nonvolatile.
Remarks:	"ga" may be specified as "HH/M/SSS" or "xNNNN". n = dataformat used for this ga, 0..21 (see data formats!) f=receive enable, "E"/"D" E=Enabled, D=Disabled While processing this command the EIB communication is stopped!

| Reply: | If option "V-" is set: none |
| | If "V+" is set: "Busy...", after processing "Ok". |

Possible error messages:	
	If option "V-" is set: none
	IF "V+" is set:
	"!Bad Command Format"
	"!Bad Group Address"

| Example: | S1/0/1-1/0/255:1,E defines the group addresses 1/0/1 to 1/0/255 as 1-6 Bit value and enables the receive. This attributes are used i.e. for ON/OFF telegrams. |

| additional hint: | The device needs about 10 seconds for setting up the whole address range. |

ASCII-Command: Erase all group address attributes

| Command: | "ERASE!" |

| Purpose: | Remove all prvious settings to group address attributes, sets all group addresses to dataformat 0, receiving disabled. |

| additional hint: | The device needs about 10 seconds for setting up the whole address range, while processing this command the EIB communication is stopped! |

ASCII-Command: Read request to an group address

| Command: | "Rga" |

| Purpose: | Send a value read telegram to the bus for the specified address "ga". |

| Remarks: | "ga" may be specified as "HH/M/SSS" or "xNNNN". |
| | The read telegram is only sent to the bus if the group address is activated or the global option "R+" has been set. |

| Reply: | If option "V-" is set: none |
| | If option "V-" is set: "OK" on success |

Possible error messages:	
	If option "V-" is set: none
	IF "V+" is set:
	"!Bad Command Format"
	"!Bad Group Address"
	"!Group Address Disabled"
	"!EIB not connected"

| Example: | R0/0/1 creates a value read telegram for GA 0/0/1. |

ASCII-Command: Send value to a group address

| Command: | "Wga=v" |

Purpose:	Send a value write telegram to the bus for the specified address "ga" with value "v"
Remarks:	"ga" may be specified as "HH/M/SSS" or "xNNNN". "v"=Data value, dependant on the selected data format (see section data formats)
Reply:	If option "V-" is set: none If option "V-" is set: "OK" on success
Possible error messages:	
	If option "V-" is set: none IF "V+" is set: "!Bad Command Format" "!Bad Group Address" "!Group Address Disabled" "!EIB not connected" "!Bad value"
Examples:	W1/0/1=1 sends a „ON" telegram on the group address 1/0/1 if this one is setup with data format 1. W1/0/1=0 sends a „OFF" telegram on the group address 1/0/1 if this one is setup with data format 1.

Data formats:

The supported data formats are supporting all currently defined non structured "DPT" types of the EIB Specification. Those have been extended to support special host requirements.

Data formats: Format1

Data type:	Binary data, 1-6 Bit raw data length
Value range:	0..63 Decimal, x0..x3F Hexadecimal
Remarks:	Supports the DPT 1.x,2.x and 3.x
	On sending either a decimal or an hexadecimal value may be specified.
	Hexadecimal values must be prefixed with an "x".
	On receiving the data format is decimal.

Data formats: Format2

Data type:	1 ASCII-char
Value range:	20H..0FFH
Remarks:	DPT 4.001 8 Bit ," character ASCII"
	On sending the char following the „=" is sent to the bus.
	On receiving exactly one char is transmitted.

Data formats: Format3

Data type:	Percentage 0-100%
Value range:	0..100 Decimal, x0..x64 Hexadecimal
Remarks:	DPT 5.001 8 Bit ," percentage"
	On sending either a decimal or an hexadecimal value may be specified.
	Hexadecimal values must be prefixed with an "x".
	On receiving the data format is decimal with 2 fractional digits.

Data formats: Format4

Data type:	Angle 0-360 Degrees
Value range:	0..360 Decimal, x0..x168 Hexadecimal
Remarks:	DPT 5.003 8 Bit ," angle degrees "
	On sending either a decimal or an hexadecimal value may be specified.
	Hexadecimal values must be prefixed with an "x".
	On receiving the data format is decimal with 2 fractional digits.

Data formats: Format5

Data type:	1 Byte unsigned
Value range:	0..255 Decimal, x0..xff Hexadecimal
Remarks:	DPT 5.010 8 Bit ," 1 Byte unsigned value" On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal.

Data formats: Format6

Data type:	1 Byte signed
Value range:	-128..127 Decimal, x0..xff Hexadecimal
Remarks:	DPT 6.x 8 Bit ," 1 Byte signed value " On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal, negativ values are prefixed with "-".

Data formats: Format7

Data type:	2 Byte unsigned
Value range:	0..65535 Decimal, x0..xffff Hexadecimal
Remarks:	DPT 7.x 2 Byte," 2 Byte unsigned value" On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal.

Data formats: Format8

Data type:	2 Byte signed
Value range:	-32768..32767 Decimal, x0..xffff Hexadecimal
Remarks:	DPT 8.x 2 Byte," 2 Byte signed value " On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal, negativ values are prefixed with "-".

Data formats: Format9

Data type:	2 Byte EIB floating point
Value range:	-671088.64 bis 670760.96, exponents 0..15.
Auflösung:	0.01.
Data format:	"SEEEEMMMMMMMMMMM" S= sign of mantissa E= Exponent M= Mantissa
Remarks:	DPT 9.x 2 Byte," 2 Byte float value " The EIB floating point format consists of the sign bit of the mantissa, 4 bits exponent and 11 bits for the mantissa itself. Per definition the mantissa has to be multiplied with 0.01. On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". If given as hexadecimal the value must contain the floating point value already! On receiving the data format is decimal with 2 fractional digits, negativ values are prefixed with "-".

Data formats: Format10

Data type:	3 Byte Time
Data format:	"hh:mm:ss", hh= hour 0..23, mm= minute 0..59, ss= second 0..59
Remarks:	DPT 10.x 3 Byte," Time" On sending either the format "hh:mm:ss" or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". If given as hexadecimal the value must contain the hh value in the 1st byte, the mm value in the 2nd byte and the ss value in the third byte. "xhhmmss", hh= 00..17h, mm= 00..3Bh, ss= 00..3Bh On receiving the data is formatted as "hh:mm:ss".

Data formats: Format11

Data type:	3 Byte Date
Data format:	"dd.mm.yy", dd= day 1..31, mm= month 1..12, yy= year 00..99
Remarks:	DPT 11.x 3 Byte," Date" On sending either the format "dd.mm.yy" or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". If given as hexadecimal the value must contain the dd value in the 1st byte, the mm value in the 2nd byte and the yy value in the third byte. "xddmmyy", dd= 01..1fh, mm= 01..0Ch, yy= 00..63h On receiving the data is formatted as "dd:mm:yy". The data is not completely validated, that means the date 31.02.02 will be accepted by the device.

Data formats: Format12

Data type:	4 Byte unsigned
Value range:	0...4294967295 Decimal, x0..xffffffff Hexadecimal
Remarks:	DPT 12.x 4 Byte," 4 Byte unsigned value" On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal.

Data formats: Format13

Data type:	4 Byte signed
Value range:	-2147483648..2147483647 Decimal, x0..xffffffff Hexadecimal
Remarks:	DPT 13.x 4 Byte," 4 Byte signed value " On sending either a decimal or an hexadecimal value may be specified. Hexadecimal values must be prefixed with an "x". On receiving the data format is decimal, negativ values are prefixed with "-".

Data formats: Format14

Data type: 4 Byte IEEE floating point according P754

Value range: 1.8446762e19, x0..xffffffff Hexadecimal

Remarks: DPT 14.x   4 Byte," 4 Byte float value   "
                          Concerning the value range and precision please read the related literature.
                          On sending either a decimal or an hexadecimal value may be specified.
                          Hexadecimal values must be prefixed with an "x".
                          If given as hexadecimal the value must contain the floating point value
                          already!
                          On receiving the data format is decimal with up to 8 fractional digits, negativ
                          values are prefixed with "-".
                          On very small or big values an exponential format may be generated.
                          There is no rounding mechanism on received values.

Data formats: Format15

Data type: 4 Byte access control

Value range: 0..4294967295 Dezimal, x0..xffffffff Hexadezimal

Remarks: DPT 12.x   4 Byte," 4 Byte unsigned value"
                          On sending either a decimal or an hexadecimal value may be specified.
                          Hexadecimal values must be prefixed with an "x".
                          On receiving the values are sent hexadecimal with 8 chars.

Data formats: Format16

Data type: Textstring

Value range: 1 to 14 Ascii chars from 20h..0ffh.

Remarks: DPT 16.x 14 Byte," Character String   "
                          Deviating from the DPT16.x specification exactly the given count of
                          chars are transmitted. Therefore the string must be extended with blanks to
                          get 14 chars transmitted. If the given text exceeds 14 chars the extra chars
                          are ignored.

Data formats: Format17

Data type:	Textstring, zero terminated
Value range:	1 to 13 Ascii chars from 20h..0ffh.
Remarks:	DPT 16.x 14 Byte," Character String "
	Internally there is an zero byte appended to the given text.
	Deviating from the DPT16.x specification exactly the given count of
	chars are transmitted. Therefore the string must be extended with blanks to
	get 13 chars and the terminating zero transmitted. If the given text exceeds 13
	chars the extra chars are ignored.

Data formats: Format18

Data type:	1 to 14 Byte Decimal format, Semicolon separated
Value range:	Range per Byte 0..255
Remarks:	The single byte values have to be separated by a semicolon char(";")
	Die einzelnen Bytewerte sind durch Semikolon zu separieren.
	If the source address option is activated more than 64 chars may result on
	receiving. These will be limited to 64 chars.
Example:	Send a 1 Byte Telegram: "Wga=127"
	Send a 2 Byte Telegram: "Wga=127;0"
	Send a 3 Byte Telegram: "Wga=127;0;100" …

On receiving the data is formatted decimal, the values are semicolon separated.

Data formats: Format19

Data type:	1 to 14 Byte Hexadecimalformat, Semicolon separated
Value range:	range per Byte 0..ffh
Remarks:	The single byte values have to be separated by a semicolon char (";").
	Hexadecimal prefix neither needed nor allowed.
	If the source address option is activated more than 64 chars may result on
	receiving. These will be limited to 64 chars.
Example:	Send a 1 Byte Telegram: "Wga=7f"
	Send a 2 Byte Telegram: "Wga=7f;0"
	Send a 3 Byte Telegram: "Wga=7f;0;64" …

On receiving the data is formatted hexadecimal, the values are semicolon separated.

Data formats: Format20

Data type: 1 to 14 Byte Hexadecimalformat, unseparated

Value range: range per byte 0..ffh

Remarks: The different byte values must not be separated.
Hexadecimal prefix neither needed nor allowed.
For each byte value 2 chars must be transmitted.

Example: Send a 1 Byte Telegram: "Wga=7f"
Send a 2 Byte Telegram: "Wga=7f00"
Send a 3 Byte Telegram: "Wga=7f0064" …

On receiving the data is formatted hexadecimal, the values are not separated, for each byte 2 chars are generated.

Data formats: Format21

Data type: 1 to 6 Bit Hexadecimalformat

Value range: 0..3fh.

Remarks: Hexadecimal prefix neither needed nor allowed.

Example: Send a 1-6 Bit Telegram: "Wga=01"

On receiving the data is formatted hexadecimal, 2 chars are generated.

Receive telegrams

Received telegrams containing group address data are first checked for „Echo" (see option „OE").
Then the device checks whether the ratget group address is enabled (see "Sga" command).
If the address is enabled the received datalength is compared with the setup data len.
If the length is different the Value field gets "?" assigned as content. Now the data is formatted as follows:

 "N PH>GA=Value"

N= Telegram number, only if "ON+"
PH= source address, only if "OS+"
GA= target address

Dependant on the Option „X" those datas are formatted as

"017 12.03.127>0/3/49=..." or as "11 B37F>0331=..."

The data value itself is formatted depending on the selected data format.

Example telegram hexadecimal: 13 000A>0003=0

Example telegram decimal: 026 00.00.010>00/0/003=0

Basic parameterization with the b+b-Terminal-Program

The basic parameterization of the EIB-ASCII-Terminal can be easily done with the b+b-Terminal-Program.

This terminal-program is located in the installation folder of EIB.VB for the EIB-ASCII-Terminal or on the EIBTools CD in folder Setup\Bin\EIBAT_EIBVB, if EIB.VB ist not installed. The file b+bTerminal.exe can be copied to any location on your local hard disk and is started with a double click.

First you have to configure the settings for the serial interface in the program by clicking in the menu on „Settings" and then on the item „Schnittstelle / Allgemeine Einstellungen". In the appearing dialog you have to apply the following settings:

Interface: COM port, where the EIB-ASCII-Terminal is connected to
Baudrate: baudrate like selected at the EIB-ASCII-Terminal
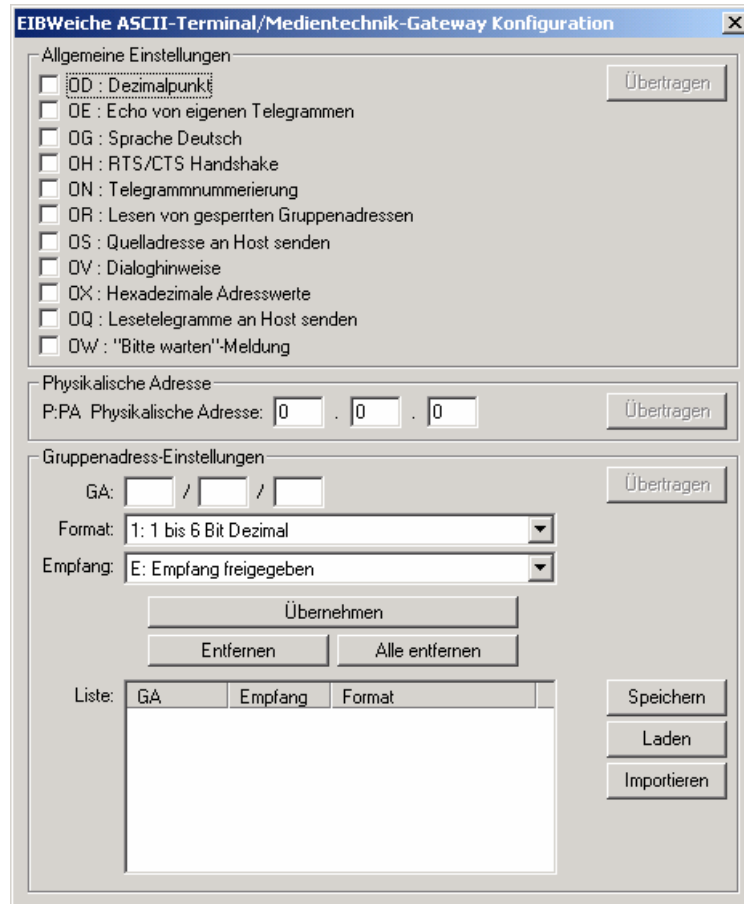Databits: 8
Stopbits: 1
Parity: N (keine)
Protocol: Without or Hardware (RTS/CTS)

The dialog must be colsed with the button „OK".

Now the connection to the EIB-ASCII-Terminal can be opened, while clicking in the toolbar on Start (green arrow). To check the connection you now should request the version of the EIB-ASCII-Terminal with the command „?V". If the terminal doesn't answer, you should check if your connection settings are correct.
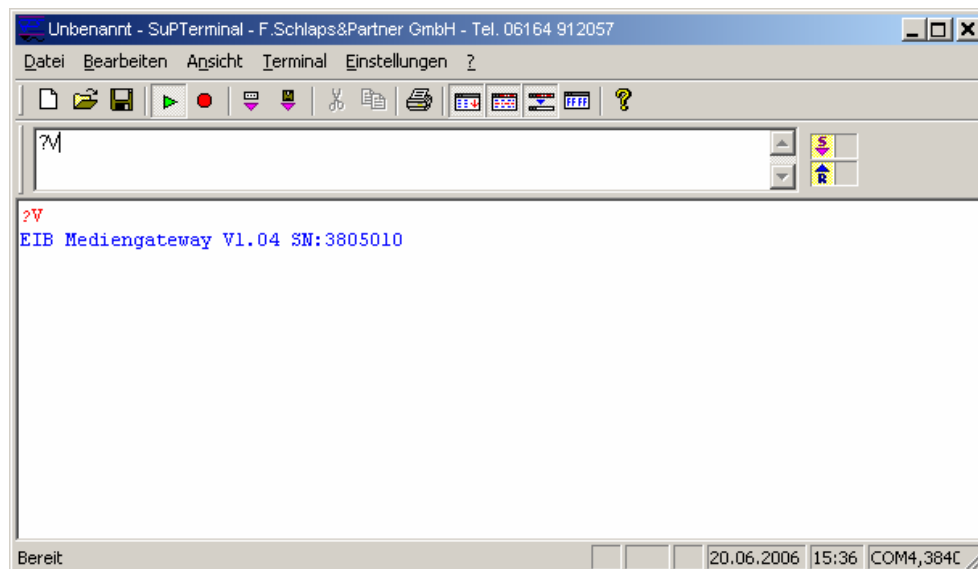
The dialog to parameterize the EIB-ASCII-Terminal is opened, when you click in the menu „Settings" on "EIBWeiche ASCII-Terminal/Medientechnik-Gateway". The following dialog appears:

In chapter „Allgemeine Einstellungen" the options of the EIB-ASCII-Terminal can be parameterized, like described above.
After you have applied the settings, it will be transferred to the EIB-ASCII-Terminal if you click on the button „Übertragen".

In chapter „Physical Address" you can setup the physical address of the EIB-ASCII-Terminal.

In chapter „Gruppenadress-Einstellungen" you can configure the goup addresses for the EIB-ASCII-Terminal. Over the button „Speichern" the actual group addresses are saved in a text file. The content of this text file can be used to configure the EIB-ASCII-Terminal with the address definitions. The configuration is done directly when you click on the button „Übertragen".
The saved settings can be loaded again with the button „Laden". With the button „Importieren" you can load the complete data of ein EIB project, that has been imported with EIB Explorer or the FIAVisManager. To do so you have to create a EIB Explorer or FIAVis Manager project in the according software and to import the EIB-addresses from an existing ETS-project.
Then you have to transfer the data into a directory. In the EIB Explorer software you have to select the menu item „Transfer data" or in the FIAVis Manager the menu item „Transfer data to application". In the following window have have to select the option „EIB.VB directory", configure the destination directory and execute the transfer with the button „Start...". The created GA-file contains all address informations and can be imported in the b+b-Terminal-Program.