

**eibNode**

## INSTRUCTION SHEET and OPERATING MANUAL

Art.-Nr.: 10300



# Content

General	3
Functions and Services	3
Applications	5
Applications	
Software components	6
Hardware survey	7
Data links/Ports	7
Power supply	7
Important notes regarding mounting	7
Setting up operation	9
General procedure	9
Physical Address in EIB	9
Connecting LAN	12
IP-address change	13
LineMask	17
NetID	21
Appendix	23
Status LED's	23
Technical Data	24

Copyright 11/2005

b.a.b-technologie-gmbh

Technische Änderungen vorbehalten

2

# General

## Functions and Services

*eibNode* is a coupler between an EIB installation and an EtherNet using the TCP/IP protocol. For this interconnection different services are provided:

- BMX-Server
- Table of actual status
- Recordings
- Line- and area coupling
- Logical funktions
- FTP-Server
- HTTP-Server

Using the BMX-Server-service the *eibNode* provides actual state and current switchings within the EIB installation as TCP/IP Server with Port address 1735. The server can provide all necessary data for visualization- maintainance- and other applications by using miscellaneous Software resp. Software-components.

As well the BMX-Server is able to receive switching and/or monitoring tasks from EIB and also carries them out on EIB.

Using this interface the data from the status table is provided. In the status table all existing EIB addresses (main group 0-15) and all virtual addresses (main group 16-31) are buffered. This means the last activity and the respective timestamp is buffered.

Querying these addresses all actual states may be assigned to all EIB objects. So the status table describes the complete figure of all objects within the installation.

Likewise *eibNode* has a recording buffer which may be read out by the BMX-Server.

The last 20.000 telegrams are saved within this buffer. In case of a buffer overflow the eldest dataset will be overwritten when the next telegram is received.

Using the functionality of line- resp area- coupling *eibNode* is communicating via UDP-service. (=> Setting up ETS-application).

The device communicates with all other *eibNodes* within the same network-segment, which the same UDP- port address was assigned to by the ETS-application.

The UDP-Message- handler allows the exchange of all EIB-telegrams. Beneath the Group Message Telegrams (switching telegrams) also the Single Device Telegrams (Programming telegrams) belong to it.

This allows a linespanning programming of EIB devices. Therefore the EIB typical physical addressing of the *eibNodes* and the subordinated EIB devices is required.

Logical functions may be uploaded to the *eibNode* using an external maintenance program.

*eibNode* supports the following functionalities:

- Binary logical gates
- Staircase light functionality
- Delay
- Filter functions
- Receive of system time
- Receive of system date.

The detailed description of the respective logical function please see from the documentation of the maintenance program.

Using the service FTP-server (Port 21) the firmware, the logical functions and the HTTP-Server within the *eibNode* may be maintained. The FTP-server provides rudimental FTP-functions for exchanging programs, files and documents.

**Note:**

*The FTP-function should only be used by a trained expert because the primary software kernel is affected too and may be destroyed in case of incorrect handling. If the kernel is destroyed the device cannot be repaired by external procedures.*

## Applications

Acting as a BMX-Server for EIB one of the typical applications for *eibNode* is the connection between an EIB installation and a visualization.

For this purpose not just the actual data stream of the EIB is provided within a TCP/IP network in a simple way but also the actual states are displayed . So the visualization is relieved from any initialization runs on the EIB.

Moreover these services are provided up to 10 clients the same time.

Using its logical functions *eibNode* may be used as a memory programable controller for the EIB. *eibNode* is able to handle some hundreds of logical functions.

Recorded data may be used for different purposes:

The recorded data may be used f.i. by the installer as an important tool for fault diagnostics within an EIB installation which even considers events in the past.

Data figures showing f.i. power consumption data can be provided to the user/customer. Therefore analysis programmes like Microsoft Excel may be used.

If the HTTP-Server is used small operating possibilities for the installation can be created without any special software but just using a common internet browser (MIE, Netscape...)

Acting as a line-/ or area- coupler *eibNode* builds up a Fast Backbone.

So using this application conventional line-/ area couplers may be replaced by *eibNodes*.

The topology therefore is determined by the LineMask. (please refer to the respective paragraph in the setting up manual.)

All data traffic behind the line-/ area couplers is no longer transmitted with 9600 Baud, but with 10 Mbit/s. So an explicitly higher telegram rate is allowed.

Via line-/area coupling by using the *eibNode* EIB installations which are independant from each other may be merged to one transmission medium -EtherNet with IP-

even having physical and/or logical address overlapping.

Using the NetID the individual EIB installations are still logically separated. (please refer to the respective paragraph in the setting up manual.)

Nevertheless certain address ranges still may be used globally for central functions if parametrized correctly.

## Software components

For primary setting up only the ETS with the respective application is needed.

For advanced parametrization -for LineMask and NetID- a common Internetbrowser like

- Internet Explorer from Version 5.5
- Netscape from Version 6
- Konquerer from KDE 3.2

is needed. All Browsers must have activated Java (from Version 1.18).

For programming the logical functions the *eibVision* program is available.

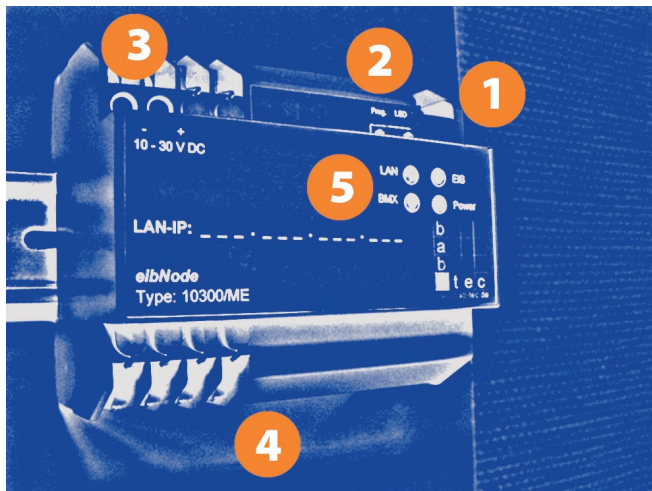
Within this programm bundle also the ActiveX component for embedding the BMX-Server functions into an external software can be found.

Moreover *eibVision* offers with the BMXClient a high-performance Browser for BMX- functions.

# Hardware survey

## Data links/Ports

- 1.) Bus clamp EIB
- 2.) Programm button-/LED for EIB
- 3.) Power supply 10-30V
- 4.) RJ 45 connector for LAN
- 5.) Signal-LED's (siehe appendix)



## Power supply

Operating voltage	10-30V DC
Power consumption	<= 5VA
Bus voltage	via EIB

## Important notes regarding mounting

Operations on low voltage net and on EIB may only be executed by trained qualified personnel.

Laying and connecting the busline, the 10-30V power supply line and the device to be mounted have to be

executed according to at the moment valid directives (DIN standard) and the EIB manual.

The device is designed to be mounted in distribution frames or electrical cabinets.

It may be used for fixed installations in

- interiors
- dry rooms
- low voltage distribution frames
- small cases

At this it has to be cared for that the environment conditions according to protection class of the device and the permitted operating temperature are kept.

Throttled bus voltage may not be used as operating voltage 10-30V DC.

### **Electric security**

DIN EN 55024	Einrichtungen der Informationstechnik
DIN EN 60950	Sicherheit von Einrichtungen der Informationstechnik
DIN EN 50090-2-2	Elektrische Systemtechnik für Heim und Gebäude

### **CE-labelling according to**

EMV-Richtlinie (Wohn- und Zweckbau)  
EN 50081-1  
EN 50082-2  
EN 50090-2-2

# Setting up operation

## General procedure

The setting up procedure is done in two steps:

- Connecting to the eibNode by the Configuration tool.
- Assigning an IP address, the physical address and other parameters.

## Physical Address in EIB

Within an EIB installation the used devices are segmented into areas, main lines and lines.

The physical addresses of the EIB components follow these structures and are also segmented into area, line and device (f.i. 4.7.11).

“Normal” EIB-devices generally may be arranged within this structure without any restrictions, except area- and line- couplers. Both devices a device address „0” has to be assigned to.

A specific attribute of these devices is that they have to recognize by means of their own physical address which other addresses are subordinated or superordinated.

So f.i. address 4.7.x (with  $x \geq 0$  and  $x \leq 255$ ) is subordinated to the line coupler with an address 4.7.0.

A device with address 4.8.x is superordinated to the mentioned line coupler.

Line- and area- couplers have to care for the correct transmission and/or routing of physically addressed telegrams (programming telegrams) to the according devices. *eibNode* is able to fulfill even these tasks.

Several eibNodes communicate via the UDP protocol. The UDP port can be set with the configuration tool (see chapter configuration tool). In order to function as a coupler also the Line Mask must be defined (see chapter Line Mask).

For the functionality for area-/ line coupling the physical address of the *eibNode* has to meet the above mentioned structural rules.

Here a special feature handling the physical addresses has to be considered:

A device address „0“ (f.i. 4.6.0) the ETS can assign just to a line coupler.

Because *eibNode* is no line coupler it has to get an physical address between 1 and 255.

Nevertheless it takes on the functionality of a line coupler with device address “0”.

To explain this exemplarily we assume a LineMask of 255. This means that the *eibNode* replaces a line coupler. More about this is explained in the chapter explaining the LineMask.

*eibNode* gets assigned a physical address from 4.7.x whereby  $x \geq 1$  and  $x \leq 255$ .

Like a “real” line coupler the *eibNode* recognises the devices in its line by means of their physical address. All devices having an address like 4.7.x (except the *eibNode* itself) must be subordinated to the *eibNode*. All other addresses are out of the range the *eibNode* manages.

If any device within the range of the *eibNodes* is addressed from exterior *eibNode* will route the telegram(s).

If the ETS addresses a device within the address range of the *eibNode* via serial interface the *eibNode* stays passive.

If the device is located outside this range *eibNode* sends out the according telegrams to LAN using the UDP-Message handler. The next *eibNode*, responsible for the address range receives the telegram and sends it into his range.

The address ranges have to be well defined (like it is necessary using line-/ area- couplers). In principle it doesn't matter which device address the *eibNode* is assigned to.

LineMask of 255 describes just the area and line subordinated to the *eibNode*.

Nevertheless in order not to lose the overview within an installation and the ETS project all *eibNodes* within this installation should get the device address "1" (f.i. 4.7.1).

A line coupler with address „0" may not exist. So *eibNode* is the first device in the list for this line.

## Connecting LAN

Connection between PC and LAN may be done by:

- LAN-cable (if operating in a network)
- crossover LAN-cable (if operating just with one PC)

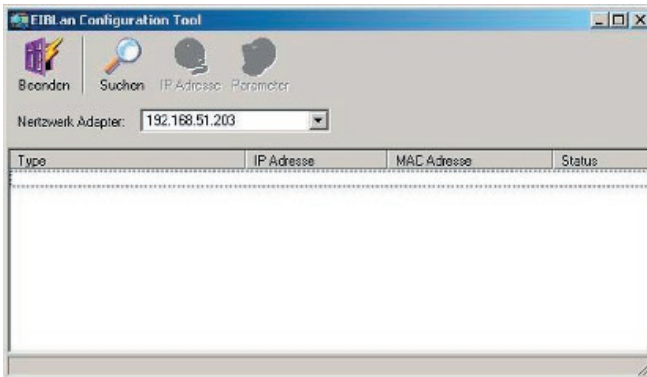
Connect the eibNode to the PC and start the configuration tool (see chapter Configuration Tool)

# Configuration Tool

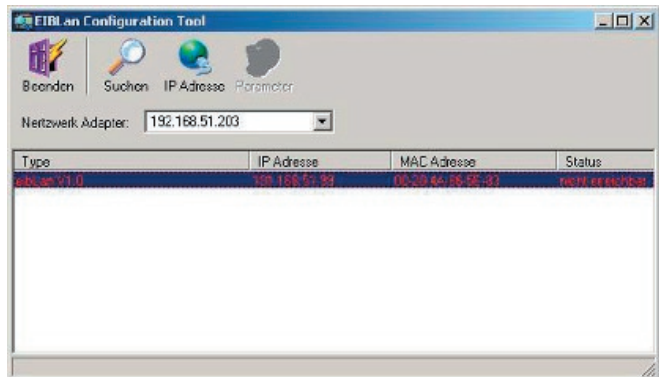
The configuration tool is used to set up an eibNode. With the software a connection to the eibNode is established and the parameters IP address, physical address and others are assigned to the eibNode.

## Start Configuration Tool

Start the configuration tool by clicking twice on the program icon. The program window appears:

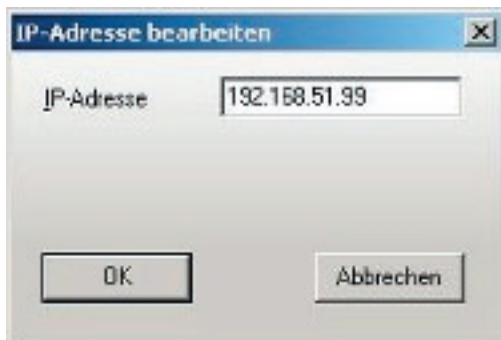


The software automatically detects the IP address(es) of the PC and shows them in the field Netzwerk Adapter (Network Adaptor). Click on the button Suchen (Search) to start searching for any eiNodes in the network. The search may take some seconds. All detected eibNodes will be listed. eibNodes that are in another IP address range than the PC will be displayed in red color.



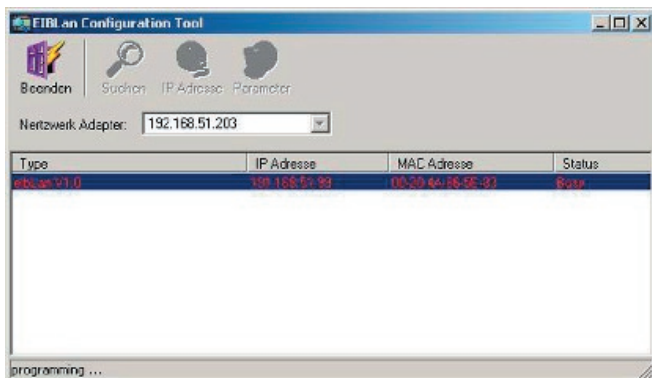
## IP-address change

To change the IP address click on the button IP Adresse:

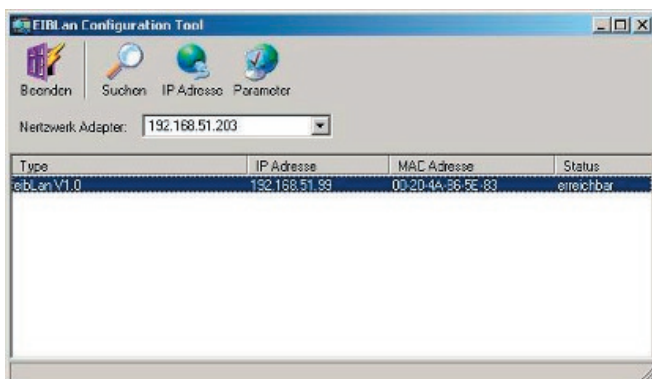


Enter a new IP address and confirm with OK. The changes now will be programmed to the eibNode. The progress is indicated in the status bar.

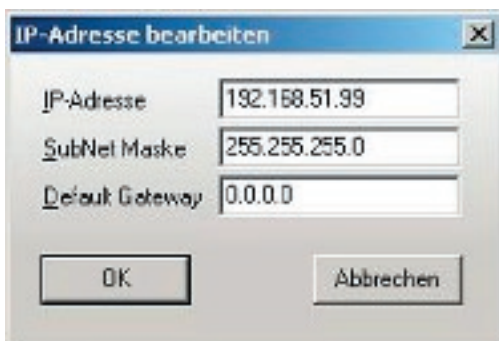
*The eibNode will reboot automatically.  
This may take some seconds.*



After programming the eibNode is in the same IP address range as the PC and will appear in black color in the list:



Click on the button IP Adresse to change the IP address, the subnet mask and the default gateway.



*Consideration: The IP address can only be changed within the current IP address range. To assign an IP address from another address range the IP address of the PC has to be changed first. Then restart the configuration tool.*

## Edit parameters

Click on the button Parameter.

The screenshot shows a dialog box titled "Parameter bearbeiten" with the following fields and controls:

- IP-Adresse: 192.168.51.99
- SubNet Maske: 255.255.255.0
- Default Gateway: 0.0.0.0
- UDP-Port: 1634 (dropdown menu) with a "Default" button.
- UDP Send IP-Adresse: 192.168.51.255 (text field) with a "Default" button.
- EIB Net-ID: 0 (spin box) with a "Default" button.
- Broadcast NetID: (spin box) with a "Default" button.
- Anlagenglobale Hauptgruppen unabhängig der NetID: A list box containing:
  - EIB Hauptgruppe 0
  - EIB Hauptgruppe 1
  - EIB Hauptgruppe 2
  - EIB Hauptgruppe 3
  - EIB Hauptgruppe 4
- phys. EIB Adresse: 1.2.21 (text field)
- Buttons: "OK" and "Abbrechen" at the bottom.

### UDP-Port

Choose the UDP Port from the menu. Click on the button Default for choosing the standard port 1634.

### UDP Send IP-Adresse

Enter the sending IP address for UDP connections. Click on the button Default for choosing the standard address 192.168.51.255.

### **EIB Net-ID**

Enter the EIB Net ID. Click on the button Default to choose the ID 0.

### **Broadcast NetID**

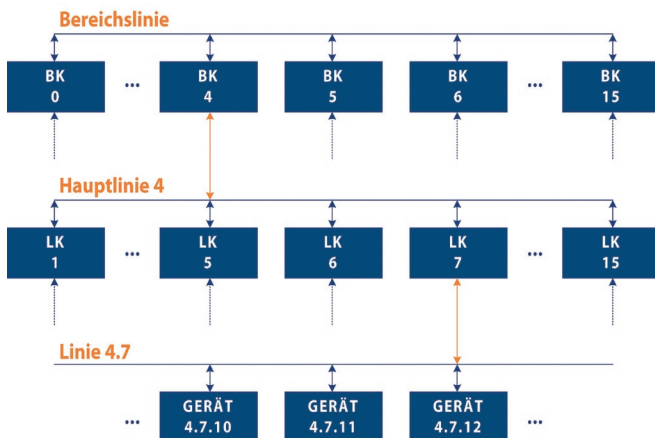
Enter the Broadcast NetID. Click on the button Default to choose the standard ID 0.

### **Anlagenglobale Hauptgruppen (global main groups)**

Set the EIB main groups that shall be used as global main groups for the complete facility.

## LineMask

Usually an EIB-installation is divided into areas, lines and devices. In this tree- structure the physical addresses are reflected. A physical address is also divided in area, line and device. (f.i. 4.7.11 means area 4, line 7 and device 11). The according lines and areas are connected using line-/ area couplers.



Beneath Group Message Telegrams (Switching telegrams), whose dataflow is decided by filter tables the line-/ area coupler route Single Device Telegrams (Programming telegrams) according to their own physical address.

A line coupler always has a device address „0“. All devices with the same area-/line address but different device address are subordinated to this line coupler.

In this example device 4.7.11 is subordinated to line coupler 4.7.0. All other devices don't belong to this line.

An area coupler always has a line address „0“ and device address „0“.

All devices with the same area address are subordinated to this area coupler.

In the example device 4.7.11 is subordinated to line coupler 4.7.0 and area coupler 4.0.0.

For this mode of operation beneath the syntax for the physical address the technical syntax has to be used. The address is depicted as a 2 Byte digit. The following split-up

applies:

2. Byte								1. Byte							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bereich				Linie				Geräteadresse							

In hexadecimal syntax the physical address 4.7.11 is depicted as 2 Byte digit 470Bh.

A line coupler having the address 4.7.0 (or 4700h) manages all addresses from 4.7.0 up to 4.7.255 (resp from 4700h up to 47FFh). A line coupler computes its „responsibility“ by means of the low-order Byte or the device address

Therefore this formula is used:

$$(\text{own address or } 255) = (\text{target address or } 255)$$

or

$$(\text{own address or } 00FFh) = (\text{target address or } 00FFh)$$

The line coupler has a LineMask of 255 (in Hex 00FFh).

An area coupler determines its „responsibility“ by means of line- and device address.

It uses the LineMask of 4095 (in Hex 0FFFh). So area coupler 4.0.0 manages all addresses from 4.0.0 up to 4.7.255.

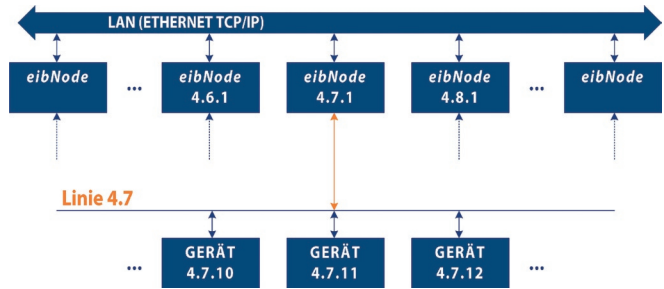
It uses the same formula as the line coupler:

$$(\text{own address or } 0FFFh) = (\text{target address or } 0FFFh)$$

It is able to manage 16 lines underneath (15 line coupler and one area line).

Through this the typical tree structure results.

The *eibNode* in consequence of the different mediums used has a different physical structure:



All *eibNodes* equally are connected in parallel to LAN Independent of the particular IP-address each *eibNode* has an physical EIB address of its own. This EIB address is defined by the ETS programming.

The address has to be unique as known from area- and line- couplers. Different from the tree structure of the EIB there don't exist area lines resp. main lines.

In principle *eibNodes* act the same way as area-/ line couplers when defining their „responsibility“.

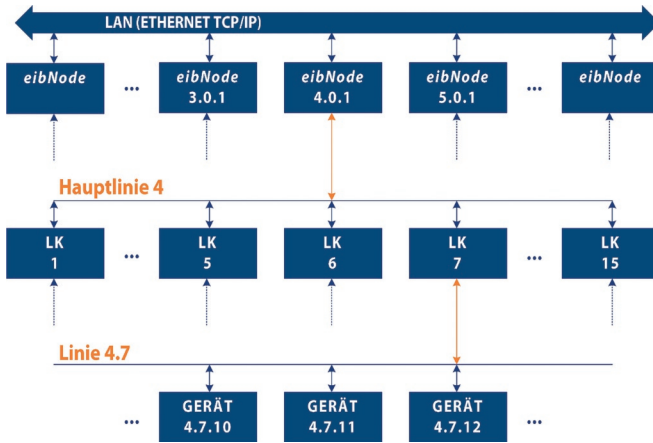
The formula:

$(\text{own address or LineMask}) = (\text{target address or LineMask})$   
is used.

The LineMask is adjustable in the *eibNode*. According to the setting of the LineMask the number of addressable area couplers, line couplers and devices underneath the *eibNode* results.

A LineMask of 255 (00FFh) emulates the line structure of the EIB. In this case 256 *eibNodes* may be used each managing an EIB-Line with up to 254 devices (256 addresses minus device address 0 minus address of the *eibNodes*). (see picture above)

A LineMask of 4095 (0FFFh) emulates the area structure of the EIB. In this case 16 *eibNodes* may be used. Each *eibNode* is able to manage 15 line couplers with each up to 255 devices as well as another 239 devices (256 addresses minus device address 0 minus address of the *eibNodes* minus 15 line coupler addresses) in the main line.



Because of the free choice of the Linemask other partitioning is possible (theoretical values):

LineMask	Hex	eibNode	Area couplers	Line couplers	Devices
15	000Fh	4096	0	0	14
31	001Fh	2048	0	0	30
63	003Fh	1024	0	0	62
127	007Fh	512	0	0	126
255	00FFh	256	0	0	254
511	01FFh	128	0	1	509
1023	03FFh	64	0	3	1019
2047	07FFh	32	0	7	2039
4095	0FFFh	16	0	15	4079
8191	1FFFh	8	1	30	8158
16383	3FFFh	4	3	60	16318

*eibNode* = number of applicable *eibNodes*

area coupler = number of area couplers per *eibNode*

line coupler = number of line couplers per *eibNode*

devices = number of other EIB devices per *eibNode*

Because the *eibNodes* must not have the same LineMask also combinations within this table are possible.

Within installation segments with a high bus load the

*eibNode* f.i. replaces the line coupler, in other segments the area coupler or it is positioned even above the area coupler.

The *eibNode*'s advantage over the classic coupler is up to its higher velocity in the backbone.

According to this installation segments with a high bus load (cyclic transmitter, heating control etc.) should have less devices underneath the *eibNode* than installation segments with a lower bus load.

A reasonable segmentation results from a analysis of the respective bus loads.

## NetID

If more than one EIB installations are planned within the same locality these separate installations may be merged via LAN.

Each *eibNode* has an NetID of its own. Beneath the network parameters (IP-address, SubNet and UDP-Message handler) the *eibNodes* having the same NetID generate an EIB installation.

Within this defined range no overlappings of group addresses is allowed- just as in an „usual“ EIB installation. Moreover the physical addressing has to be unique and adjusted to the net topology.

The NetID differentiates EIB installations now. The whole address range of the group addresses and the physical addresses is extended by a figure, the NetID. Anyhow all *eibNodes* communicate via the same network (same UDP-Port).

A central visualization or any other central controlling tool now is able to handle data from different EIB installations via LAN using the according NetID. As well to all installations data may be sent.

The NetID is a 1 Byte digit from 0 to 255. Each message sent via UDP-Message- handler is sent with the NetID of the sender. If an *eibNode* acts as receiver usually just manages messages having the same NetID as itself.

There is admittedly an exception according to certain Group Messages telegrams (Switching telegrams). This will be explained later.

Because the *eibNodes* have this unique assignment according to Single Device telegrams (programming telegrams), each EIB installation is generated and programmed with an ETS- project independent from the other EIB installations.

Beneath the advantage of parallel programming this features the independency of the several installations. Address overlapping is not possible

According to this the different programmers don't have to balance among each other.

So the setting up operation becomes more calculable. 2

Even when upgrading an existing EIB installation the old ETS project must not be extended but a new project is generated.

Nevertheless all data is available centrally via LAN.

In addition to the NetID for Group Messages telegrams (switching telegrams) main group ranges may be set transparent for the *eibNodes*.

These special central addresses are managed by the receiver independent from their NetID.

# Appendix

## Status LED´s

The actual operational status of the *eibNode* is displayed by the LEDs seen on the front of the device. Possible states are:

### Boot phase:

During the boot phase the POWER-LED flashes for about 30 sec "orange" and programming LED for EIB is active. Expiration of the boot phase is signalised by a check of all LEDs. After this the POWER-LED flashes "green".

If all LEDs blink in a common mode (timed in second) a hardware failure exists.

### POWER-LED

OFF	Device is not ready. Operating voltage is „off“.
RED	Hardware error
GREEN	Device is ready and configured.
ORANGE	<i>eibNode</i> is booting (about 30 sec)
blink	Timed in second: BCU is not configured with ETS
Blink	Every 2 seconds BCU is programmed with wrong data (nonsensical IP-Parameter)

### BMX-LED

OFF	No data exchange with <i>eibNode</i> programm kernel
RED	Telegram sent via UDP-Message handler
GREEN	An IP-connection to <i>eibNode</i> is built up
ORANGE	Data exchange via IP-connection is displayed

**EIB-LED**

OFF	<i>eibNode</i> is not connected to EIB
RED	Blinking; <i>eibNode</i> tries to connect to EIB
GREEN	<i>eibNode</i> is connected to EIB (controlled cyclically)
ORANGE	<i>eibNode</i> exchanges data with EIB

**LAN-LED**

OFF	<i>eibNode</i> is not connected to LAN
RED	Not used
GREEN	<i>eibNode</i> has a physical connection to LAN
ORANGE	<i>eibNode</i> exchanges data with LAN

## Technical Data

Operating voltage:	10-30V DC
Power consumption:	<= 5VA
Bus voltage:	über den EIB

### Important:

*The throttled EIB bus voltage may not be used as operating voltage!*

eibNode has an gold-cap capacitor for data buffering of the internal clock. This serves a buffering of 30 minutes in case of operating voltage blackout.

Climate durability acc. to:	EN50090-2-2
Environmental temperature:	0-30 °C
Storage temperature:	-20 - +70 °C
Rel. Humidity(not conensing):	5% - 80%

### Mechanical Data:

Measures:	Row installed device; (With 6TE á 18mm)
Casing:	Plastics
Weight:	ca. 0,4 kg
Mounting:	DIN Rail acc. to EN 50022-35x7,5
Protection class:	IP20 (acc. to EN60529)

b.a.b-technologie gmbh  
Heinrich-Hertz-Straße 4

59423 Unna



Support: 0180-5669900

Tel. 0 23 03 96 24 96

Fax. 0 23 03 96 24 97

E-Mail. [info@bab-tec.de](mailto:info@bab-tec.de)

Internet: [www.bab-tec.de](http://www.bab-tec.de)