



b.a.b-technologie gmbh

# **eibNode**

## Dokumentation

Art.Nr.: 10300

Stand 11/2005  
Datum: 6. März 2012

DE



b.a.b – technologie gmbh

im INHOUSE Dortmund  
Rosemeyerstr. 14  
44139 Dortmund

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

Tel.: +49 (0) 231 – 476 425 - 30  
Fax.: +49 (0) 231 – 476 425 - 59  
[www.bab-tec.de](http://www.bab-tec.de)



# INHALTSVERZEICHNIS

<b>1</b>	<b>Allgemeine Erläuterungen .....</b>	<b>6</b>
	Funktionen und Dienste .....	6
	Anwendungen.....	7
	Softwarekomponenten.....	7
<b>2</b>	<b>Hardwareübersicht .....</b>	<b>8</b>
	Anschlüsse.....	8
	Spannungsversorgung .....	8
	Wichtige Hinweise zum Einbau.....	8
<b>3</b>	<b>Inbetriebnahme.....</b>	<b>10</b>
	Generelle Vorgehensweise .....	10
	Physikalische Adresse im EIB.....	10
<b>4</b>	<b>Verbindung LAN.....</b>	<b>11</b>
<b>5</b>	<b>Konfigurationstool .....</b>	<b>12</b>
5.1	Konfigurationstool starten .....	12
5.2	IP-Adresse ändern.....	13
5.3	Parameter bearbeiten.....	14
	UDP-Port.....	14
	UDP Send IP-Adresse .....	14
	EIB Net-ID .....	14
	Broadcast NetID .....	14
	Anlagenglobale Hauptgruppen.....	14
	Physikalische EIB Adresse .....	14
5.4	LineMask.....	15
5.5	NetID .....	18
<b>6</b>	<b>eibNode Parametrierung mit eibvision .....</b>	<b>19</b>
6.1	Das eibNode Parametrierungs Tool.....	19
6.1.1	Ein eibNode Projekt anlegen .....	19
6.1.2	Funktionen parametrieren .....	20
6.1.3	Projekt übersetzen und übertragen.....	21
6.1.4	Projekt speichern.....	21
6.2	Die Befehlszentrale .....	21
6.3	Die Elemente .....	23
6.3.1	Binäre Logik.....	23
6.3.2	Szene .....	23
6.3.3	Treppenhaus.....	24
6.3.4	Verzögerer .....	25
6.3.5	Transformer .....	26
6.3.6	Tor Gatter .....	26
6.3.7	Vergleicher .....	27
6.3.8	Hysterese .....	28
6.3.9	Zyklischer Repeater .....	29
6.3.10	Binärer Filter.....	29
6.3.11	Mathematik.....	30
6.3.12	Wertespeicher.....	31
6.3.13	Wochenuhr .....	32
6.3.14	Jahresuhr .....	33
6.3.15	Zeitletogramm senden .....	34
6.3.16	Datumsteletogramm senden.....	35
6.3.17	Zyklischer Sender .....	35
6.3.18	Zeitletogramm empfangen .....	36
6.3.19	Datumsteletogramm empfangen.....	36
6.3.20	Adressen - Initialisierung.....	37



---

6.3.21	Filter EIB .....	37
<b>7</b>	<b>Anhang .....</b>	<b>39</b>
	Status LED's .....	39
	Technische Daten .....	40



## Abbildungsverzeichnis

Abbildung 1: eibNode Anschlüsse .....	8
Abbildung 2: eibNode Konfigurationstool.....	12
Abbildung 3: eibNode erkannt.....	12
Abbildung 4: IP Adresse ändern .....	13
Abbildung 5: eibNode neugestartet.....	13
Abbildung 6: IP-Adresse bearbeiten .....	13
Abbildung 7: Parameter bearbeiten.....	14
Abbildung 8: Schema phys. Adresse .....	15
Abbildung 9: Bytedarstellung phys. Adresse .....	15
Abbildung 10: Struktur KNX/EIB und LAN - Linie.....	16
Abbildung 11: Struktur KNX/EIB und LAN - Hauptlinie.....	17
Abbildung 12: Aufteilungen .....	17
Abbildung 13: eibNode Projektverwaltung.....	19
Abbildung 14: eibNode Option .....	20
Abbildung 15: Parameter eingeben .....	20
Abbildung 16: BMX Compiler auswählen .....	21
Abbildung 17: Befehlszentrale.....	22
Abbildung 18: BMX Server Verbindungsdaten.....	22
Abbildung 19: Schaltzeiten .....	32
Abbildung 20: Jahresuhr .....	34
Abbildung 21: Filter EIB .....	38

# 1 ALLGEMEINE ERLÄUTERUNGEN

## FUNKTIONEN UND DIENSTE

Der eibNode ist ein Koppler zwischen einem EIB-Netz und dem Ethernet mit TCP/IP Protokoll. Für die Kopplung stellt das Gerät unterschiedliche Dienste zur Verfügung:

- BMX-Server
- Zustandstabelle
- Aufzeichnungen
- Linien- und Bereichskopplung
- Logische Funktionen
- FTP-Server
- HTTP-Server
- 

Über den BMX-Server stellt der eibNode die aktuellen Zustände und die laufenden Schalthandlungen in der EIB-Anlage als TCP/IP Server mit der Portadresse 1735 zur Verfügung. Über diverse verfügbare Software und Softwarekomponenten kann der Server für Visualisierungen, Wartungsanwendungen und weitere Anwendungen alle erforderlichen Daten liefern. Ebenso kann der BMX-Server Schalt- und Abfrageaufgaben entgegen nehmen und am EIB ausführen.

Über diese Schnittstelle werden die Daten aus der Zustandstabelle zur Verfügung gestellt. In dieser Tabelle werden für alle existierenden EIB-Adressen (Hauptgruppe 0 bis Hauptgruppe 15) und alle virtuellen Adressen (Hauptgruppe 16 bis Hauptgruppe 31) jeweils die letzten Schalthandlungen mit Zeitstempel zwischengespeichert. Über die Abfrage dieser Adressen können die Zustände für alle EIB-Objekte bestimmt werden, so dass diese Zustandstabelle ein vollständiges Bild aller in der EIB-Anlage befindlichen Objekte darstellt.

Ebenfalls verfügt der eibNode über einen Aufzeichnungsspeicher, der über den BMX-Server ausgelesen werden kann. Die jeweils letzten 20.000 Schalttelegramme werden in diesem Speicher abgelegt. Sollte der Speicher voll sein, so wird bei der nächsten Schalthandlung der jeweils älteste Eintrag überschrieben.

Die Funktion der Linien- und Bereichskopplung kommuniziert über den UDP-Dienst. (siehe Inbetriebnahme ETS-Applikation; hier wird auch die Portadresse gewählt). Er kommuniziert mit allen anderen eibNodes im Netzwersegment, die über die ETS-Applikation die gleiche Portadresse für die UDP-Messages bekommen haben.

Es werden über den UDP-Message-Händler alle EIB-Telegramme ausgetauscht. Hierzu gehören neben den Group Message Telegrammen (Schalttelegramme) auch die Single Device Telegramme (Programmiertelegramme). Dies ermöglicht ein linienübergreifendes Programmieren von EIB-Geräten. Voraussetzung hierfür ist die EIB-typische physikalische Adressierung der eibNodes und der darunter liegendes EIB-Gerät.

Die logischen Funktionen können über ein externes Verwaltungsprogramm in den eibNode geladen werden. Der eibNode unterstützt die folgenden Funktionen

- Binäre Logikgatter
- Treppenhausfunktion
- Verzögerungsgatter
- Filterfunktionen
- Systemzeit Empfang
- Systemdatum Empfang:

Die Beschreibung der jeweiligen logischen Funktionen entnehmen Sie bitte der Dokumentation des Verwaltungsprogrammes.

Über den FTP-Server (Port 21) kann die Firmware, die logischen Funktionen und der HTTP-Server im eibNode gewartet werden. Er stellt rudimentäre FTP-Funktionen zur Verfügung, damit Programme, Dateien und Dokumente ausgetauscht werden können.

**Achtung:**

Die FTP-Funktion sollte nur von einem eingewiesenen Fachmann benutzt werden, da auch der primäre Software-Kernel behandelt und auch zerstört werden kann. Sollte der Kernel zerstört werden, kann das Gerät durch externe Maßnahmen nicht mehr repariert werden.

## ANWENDUNGEN

---

Als BMX-Server für den EIB, ist eine der typischen Anwendungen die Verbindung zwischen einer EIB-Anlage und einer Visualisierung. Hierfür wird nicht nur der laufende Datenverkehr des EIB auf einfacher Weise in einem TCP/IP Netz zur Verfügung gestellt, auch die aktuellen Zustände werden geliefert, so dass Visualisierungen keine zeitaufwendigen Initialisierungsläufe im EIB-Netz machen müssen. Weiterhin steht dieser Dienst bis zu 10 Clients gleichzeitig zur Verfügung.

Mit seinen logischen Funktionen kann der eibNode als speicherprogrammierbare Steuerung für den EIB eingesetzt werden, die einige 100 logische Funktionen abarbeiten kann.

Die Aufzeichnungsdaten können für unterschiedliche Zwecke genutzt werden: Sie können für den Installateur als wichtiges Werkzeug für die Fehlersuche in einer EIB-Anlage eingesetzt werden, dass auch rückwirkende Ereignisse berücksichtigt. Für den Anwender können Datenkurven für z.B. elektrische Verbrauchsdaten über Auswerteprogramme wie z.B. Microsoft Excel bereitgestellt werden.

Über die Nutzung des HTTP-Servers können direkt vom eibNode ohne weitere Spezialsoftware mittels Internetbrowser einfache Bedienmöglichkeiten geschaffen werden.

Über die Linien- und Bereichskopplung bildet der eibNode einen Fast-Backbone. Mit dieser Anwendung können die klassischen Linien- und Bereichskoppler des EIB durch eibNodes ersetzt werden. Die Topologie wird hierzu mit der LineMask bestimmt. (Siehe entsprechenden Abschnitt in der Inbetriebnahme.) Die Daten hinter Linien- und Bereichskopplern werden dann nicht mehr mit 9600 Baud, sondern mit 10 Mbit/s übertragen, was eine deutlich höhere Telegrammlast zulässt.

Über die Linien- und Bereichskopplung können ebenfalls unabhängige EIB-Anlagen mit physikalischer und logischer Adressüberschneidung auf ein Übertragungsmedium, Ethernet mit IP, zusammengeschaltet werden. Über die NetID werden nach wie vor die einzelnen EIB-Anlagen logisch separat gehalten. (Siehe entsprechenden Abschnitt in der Inbetriebnahme.) Adressbereiche können aber dennoch global als Zentralfunktion in allen EIB-Anlagen über die Parametrierung genutzt werden.

## SOFTWAREKOMPONENTEN

---

Für die primäre Inbetriebnahme wird ausschließlich die ETS mit der entsprechenden Applikation benötigt. Für die weitergehende Parametrierung der LineMask und der NetID wird ein Internetbrowser wie

- Mozilla Firefox
- Google Chrome

benötigt. Auf allen Browsern Muss Java (ab Version 1.18) aktiviert sein.

Für die Parametrierung der logischen Funktionen steht das eibNode Inbetriebnahme Tool, als Teil der Visualisierungssoftware eibVision zur Verfügung. Mit diesem Tool können eibNode Projekte komfortabel erstellt, verwaltet und in eibNodes übertragen werden. In eibVision befinden sich auch die ActiveX Komponenten für die Einbindung der BMX-Server Funktionen in eine Fremdsoftware. Weiter wird mit dem BMX-Client ein leistungsstarker Browser für die BMX Funktionen zur Verfügung gestellt.

## 2 HARDWAREÜBERSICHT

### ANSCHLÜSSE

- 1.) Busklemme EIB
- 2.) Programmieraste-/LED für EIB
- 3.) Spannungsversorgung 10-30V
- 4.) RJ 45 Buchse für Ethernet LAN an der Unterseite des Gerätes
- 5.) Signal-LED's (siehe Anhang)

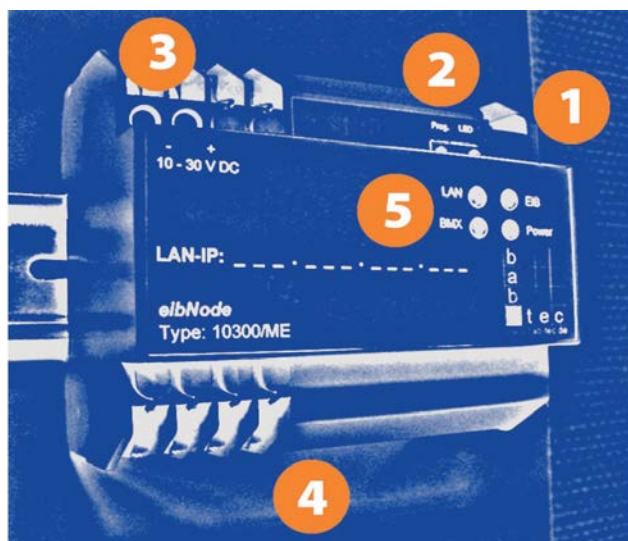


Abbildung 1: eibNode Anschlüsse

### SPANNUNGSVERSORGUNG

Betriebsspannung	10-30V DC
Leistungsaufnahme	<= 5VA
Busspannung	über EIB

### WICHTIGE HINWEISE ZUM EINBAU

Arbeiten am Niederspannungsnetz und am EIB-Bus dürfen nur von geschultem Elektrofachpersonal ausgeführt werden.

Verlegung und Anschluss der Busleitung, der 10-30V Spannungsleitung sowie der/des einzubauenden Gerätes müssen entsprechend den zurzeit gültigen Richtlinien nach DIN-VDE sowie des EIB-Handbuches durchgeführt werden.

Das Gerät ist für den Betrieb im Verteiler bzw. Schaltschrank vorgesehen. Das Gerät kann für feste Installationen in

- Innenräumen
- Trockenen Räumen
- Niederspannungsverteilern
- Kleingehäuse

verwendet werden.





Hierbei ist zu beachten, dass die Umgebungsbedingungen gemäß der Schutzklasse des Gerätes und der zulässigen Betriebstemperatur eingehalten werden.

Die gedrosselte Busspannung darf nicht zur Betriebsspannung 10-30V DC verwendet werden.

**Elektrische Sicherheit**

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-Kennzeichnung gemäß**

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

## 3 INBETRIEBNAHME

### GENERELLE VORGEHENSWEISE

Die Inbetriebnahme des eibNodes wird in zwei Schritten durchgeführt:

- Kontaktaufnahme zum eibNode mit dem Konfigurationstool und eventuell Änderung der IP-Adresse.
- Vergabe der physikalischen Adresse und weiterer Parameter.

### PHYSIKALISCHE ADRESSE IM EIB

In einer EIB-Anlage werden die Geräte in Bereiche, Hauptlinien und Linien aufgeteilt. Die physikalische Adresse der EIB-Komponenten folgt dieser Struktur und ist in Bereich, Linie und Gerät (z.B. 4.7.11) aufgeteilt. Normale EIB-Geräte können in der Regel frei in dieser Struktur angeordnet werden, bis auf Linien- und Bereichskoppler. Diese Geräte müssen mit der Geräteadresse „0“ (z.B. 4.7.0) versehen werden.

Das Besondere an diesen Geräten ist, dass sie anhand ihrer eigenen phys. Adresse erkennen müssen, welche phys. Adressen über, und welche unter ihnen angeordnet sind. So liegt z.B. die Adresse 4.7.x (mit  $x \geq 0$  und  $x \leq 255$ ) unter dem Linienkoppler der Adresse 4.7.0. Ein Gerät mit der Adresse 4.8.x liegt über dem genannten Linienkoppler.

Linien- und Bereichskoppler müssen dafür sorgen, dass physikalisch adressierte Telegramme (Programmiertelegamente) zu den richtigen Geräten weitergereicht oder geroutet werden. Der eibNode kann auch diese Aufgabe übernehmen. Mehrere eibNodes kommunizieren untereinander über das UDP-Protokoll. Im Konfigurationstool kann der UDP Port eingestellt werden (siehe Abschnitt Konfigurationstool). Über die LineMask wird definiert, welche Funktion ein eibNode im Netzwerk erfüllen soll (siehe Abschnitt LineMask).

Bei der Funktion als Linienkoppler ist eine Besonderheit bei der phys. Adresse zu berücksichtigen: Die Geräteadresse 0 (z.B. in 4.6.0) kann die ETS nur einem Linienkoppler zuordnen. Der eibNode ist aber kein Linienkoppler und muss somit eine physikalische Geräteadresse zwischen 1 und 255 haben.

Dennoch übernimmt er die Funktion eines Linienkopplers mit der Geräteadresse 0. Um es beispielhaft zu erklären gehen wir hier von einer LineMask von 255 aus. Dies bedeutet dass der eibNode einen Linienkoppler ersetzt. Näheres hierzu im Kapitel über die LineMask.

Der eibNode bekommt die phys. Adresse von 4.7.x wobei  $x \geq 1$  und  $x \leq 255$  ist. Wie ein echter Linienkoppler erkennt der eibNode die Geräte in seiner Linie anhand der phys. Adresse. Alle Geräte mit einer Adresse 4.7.x außer der Adresse des eibNodes selbst müssen unter dem eibNode angeordnet sein. Alle anderen Adressen liegen außerhalb des vom eibNode verwalteten Adressbereichs.

Wird nun von aussen ein Gerät angesprochen, dass innerhalb seiner Linie liegt, so routet der eibNode diese Telegramme. Wird mit der ETS über eine serielle Schnittstelle (innerhalb des Adressbereichs/Linie montiert) ein Gerät angesprochen, dass innerhalb des Adressbereichs/der Linie liegt, so verhält sich der eibNode passiv. Liegt das angesprochene Gerät außerhalb des Bereichs, so sendet der eibNode die entsprechenden Telegramme über den UDP-Message Händler auf das LAN. Der weitere eibNode, der diesen Adressbereich bearbeitet, nimmt nun dieses Telegramm an und sendet es in seinen Bereich. Die Adressbereiche müssen, wie es auch bei Linien- und Bereichskoppler notwendig ist, eindeutig sein.

Im Prinzip ist es egal, welche Geräteadresse der eibNode hat. Bei der LineMask von 255 beschreibt nur der Bereich und die Linie den Adressbereich unter dem eibNode. Dennoch, um die Übersicht in einer EIB-Anlage und im entsprechenden ETS Projekt zu behalten, sollten alle eibNodes in einer Anlage die Geräteadresse 1 bekommen (z.B. 4.7.1). Ein Linienkoppler mit der Adresse 0 darf nicht existieren. Somit ist der eibNode das erste Gerät in der Liste für diese Linie.



## 4 VERBINDUNG LAN

---

Die Verbindung zum PC erfolgt über:

- ein LAN-Kabel (Betrieb im Netzwerk)
- ein gedrehtes LAN-Kabel (Betrieb mit nur einem PC)

Verbinden Sie den PC mit dem eibNode (entweder direkt oder über ein Netzwerk) und starten Sie das Konfigurationstool (siehe Abschnitt Konfigurationstool).

## 5 KONFIGURATIONSTOOL

Das Konfigurationstool dient zur Inbetriebnahme eines eibNodes. Mit der Software wird eine Verbindung zum eibNode hergestellt und die Parameter IP-Adresse, Physikalische Adresse und Weitere werden vergeben.

### 5.1 KONFIGURATIONSTOOL STARTEN

Starten Sie das Konfigurationstool mit einem Doppelklick auf das Programmicon. Das Programmfenster erscheint:

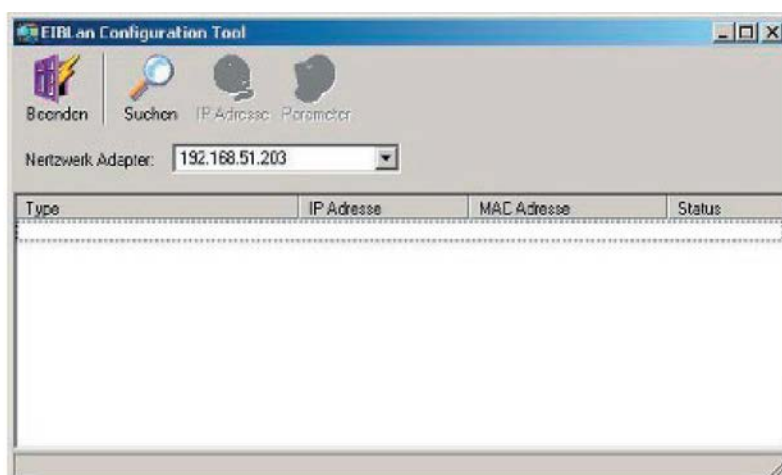


Abbildung 2: eibNode Konfigurationstool

Das Programm erkennt automatisch die IP-Adresse(n) des PCs und trägt sie als Netzwerk Adapter ein. Klicken Sie auf den Button "Suchen", um die Suche nach eibNodes im Netzwerk zu starten. Die Suche dauert einige Sekunden.

Das Konfigurationstool listet alle eibNodes im Netzwerk auf. eibNodes, die sich in einem anderen Nummernkreis befinden, als der PC, werden rot markiert.

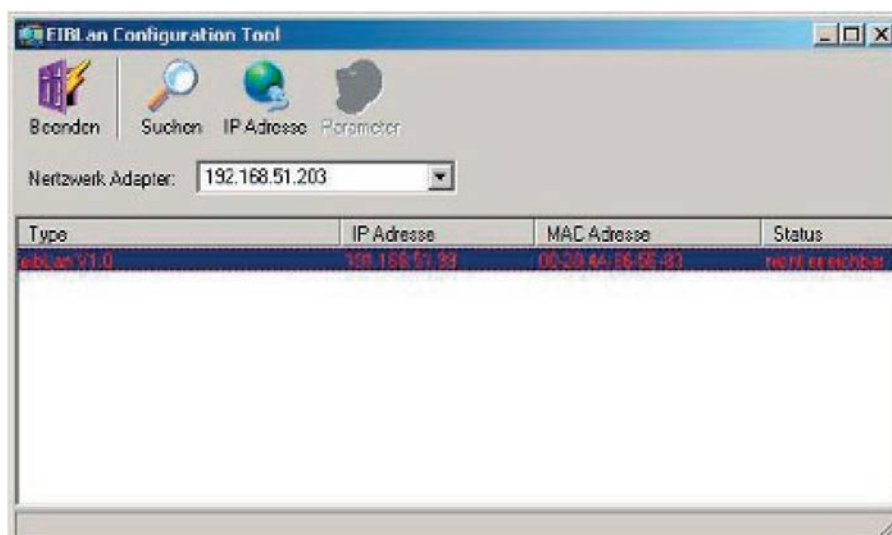


Abbildung 3: eibNode erkannt



## 5.2 IP-ADRESSE ÄNDERN

Um die IP-Adresse zu ändern, klicken Sie auf den Button IP Adresse:



Abbildung 4: IP Adresse ändern

Geben Sie eine neue IP-Adresse ein und bestätigen Sie mit OK. Die Änderungen werden nun in den eibNode programmiert. Der Fortschritt wird in der Statusleiste angezeigt (unten links).

**Der eibNode startet automatisch neu und ist nach wenigen Sekunden wieder erreichbar.**

Nach dem Programmieren befindet sich der eibNode im gleichen Nummernkreis, wie der PC und wird nicht mehr rot dargestellt:

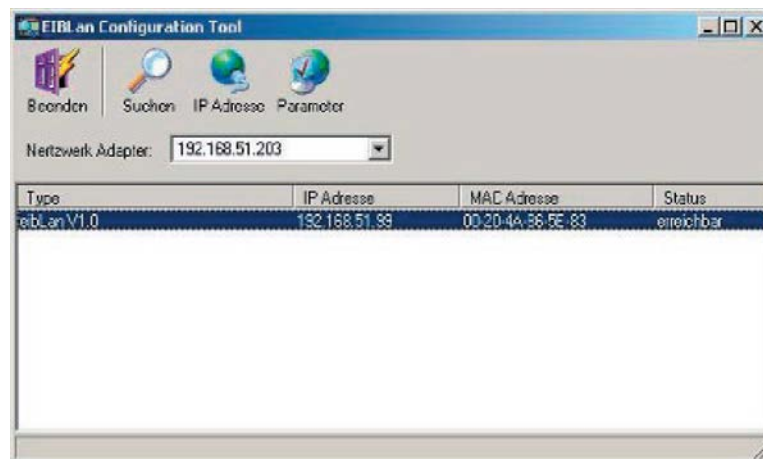


Abbildung 5: eibNode neugestartet

Wenn Sie jetzt auf den Button IP Adresse klicken, können Sie die IP-Adresse, die Netzwerkmaske und das Standard Gateway ändern:

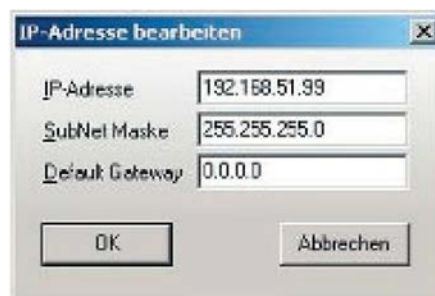


Abbildung 6: IP-Adresse bearbeiten

**Hinweis: Die IP-Adresse kann nur innerhalb des aktuellen Nummernkreises geändert werden. Um eine IP-Adresse aus einem anderen Nummernkreis zu vergeben, muss das Konfigurationstool beendet und die IP-Adresse des PCs geändert werden. Starten Sie anschließend wieder das Konfigurationstool.**

## 5.3 PARAMETER BEARBEITEN

Klicken Sie auf den Button Parameter:

The screenshot shows a window titled 'Parameter bearbeiten'. It contains 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 (with a 'Default' button)
- UDP Send IP-Adresse: 192.168.51.255 (with a 'Default' button)
- EIB Net-ID: 0 (with a 'Default' button)
- Broadcast NetID: (empty field with a 'Default' button)
- Anlagenglobale Hauptgruppen unabhängig der NetID: A list box containing checkboxes for EIB Hauptgruppe 0, 1, 2, 3, and 4.
- phys. EIB Adresse: 1.2.21
- Buttons: OK and Abbrechen.

Abbildung 7: Parameter bearbeiten

### UDP-PORT

Wählen Sie den UDP-Port aus dem Auswahlménü. Klicken Sie auf den Button Default, um den Standard-Port 1634 zu übernehmen.

### UDP SEND IP-ADRESSE

Geben Sie die Sende IP-Adresse für UDP-Übertragungen ein. Klicken Sie auf den Button Default, um die Standard- Adresse 192.168.51.255 zu übernehmen.

### EIB NET-ID

Geben Sie die EIB Net-ID ein. Klicken Sie auf den Button Default, um die Standard ID 0 zu übernehmen.

### BROADCAST NETID

Geben Sie die Broadcast NetID ein. Klicken Sie auf den Button Default, um die Standard ID 0 zu übernehmen.

### ANLAGENGLOBALE HAUPTGRUPPEN

Markieren Sie die EIB-Hauptgruppen, die global für die gesamte EIB-Anlage, unabhängig von der NetID, gelten sollen.

### PHYSIKALISCHE EIB ADRESSE

Geben Sie eine freie physikalische EIB Adresse ein. Bitte beachten Sie den Abschnitt Physikalische Adresse im EIB.



## 5.4 LINEMASK

Normalerweise wird eine EIB-Anlage in Bereiche, Linien und Geräte aufgeteilt. Eine phys. Adresse ist ebenfalls in Bereich, Linie und Gerät aufgeteilt. (z.B. 4.7.11 bedeutet Bereich 4, Linie 7 und Gerät 11). Die jeweiligen Linien und Bereiche werden mit Linien (LK)- und Bereichskoppler (BK) verbunden.

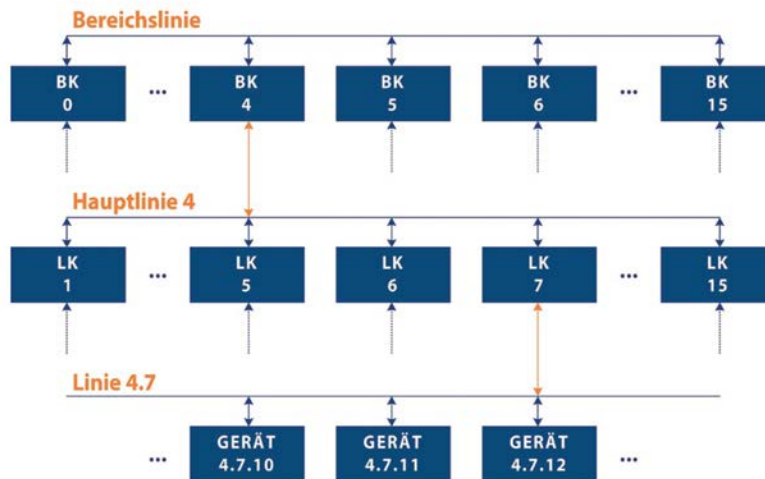


Abbildung 8: Schema phys. Adresse

Neben den Group Message Telegramme (Schalttelegramme), deren Datenwege über die Filtertabellen bestimmt werden, routen die Linien und Bereichskoppler die Single Device Telegramme (Programmiertelegramme) anhand ihrer eigenen phys. Adresse.

Ein Linienkoppler hat immer die Geräteadresse 0. Alle Geräte mit gleicher Bereichs- und Linienadresse, aber unterschiedlicher Geräteadresse liegen unterhalb des Linienkopplers. Im Beispiel liegt das Gerät 4.7.11 unter dem Linienkoppler 4.7.0. Alle anderen Geräte liegen nicht in dieser Linie.

Ein Bereichskoppler hat immer die Geräteadresse 0 und die Linienadresse 0. Alle Geräte mit gleicher Bereichsadresse, aber unterschiedlicher Linien- und Geräteadresse liegen unterhalb des Bereichskopplers. Im Beispiel liegt das Gerät 4.7.11 unter dem Linienkoppler 4.7.0 und unter dem Bereichskoppler 4.0.0.

Für die Arbeitsweise dieses Prinzips muss man sich neben der Schreibweise einer phys. Adresse auch die technische Schreibweise vergegenwärtigen. Die Adresse wird in einer zwei Byte Zahl dargestellt. Dabei gilt die folgende Aufteilung:

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

Abbildung 9: Bytedarstellung phys. Adresse

In der hexadezimalen Schreibweise ist die phys. Adresse 4.7.11 als zwei Byte zahl 470Bh. Ein Linienkoppler mit der Adresse 4.7.0 (oder 4700h) verwaltet alle Adresse von 4.7.0 bis 4.7.255 (oder von 4700h bis 47FFh). Ein Linienkoppler berechnet seine „Zuständigkeit“ anhand des niederwertigen Bytes oder der Geräteadresse. Er benutzt die Formel:

Oder-Vergatterung:  $(\text{eigene Adr. or } 255) = (\text{Zieladr. or } 255)$

anders geschrieben  $(\text{eigene Adr. or } 00FFh) = (\text{Zieladr. or } 00FFh)$

Der Linienkoppler nutzt eine LineMask von 255 (00FFh).

Ein Bereichskoppler bestimmt seine „Zuständigkeit“ anhand der Linien- und Geräteadresse. Er benutzt die LineMask von 4095 (in Hex 0FFFh). Der Bereichskoppler 4.0.0 verwaltet somit alle Adressen von 4.0.0 bis 4.7.255.

Auch er benutzt die Formel des Linienkopplers:

$$(\text{eigene Adr. or OFFFh}) = (\text{Zieladr. or OFFFh})$$

Er kann unter sich 16 Linien (15 Linienkoppler und eine Bereichsline) verwalten. Hierdurch ergibt sich die typische Baumstruktur.

Der eibNode muss durch seine beiden unterschiedlichen Medien (EIB und LAN) eine andere phys. Struktur haben.

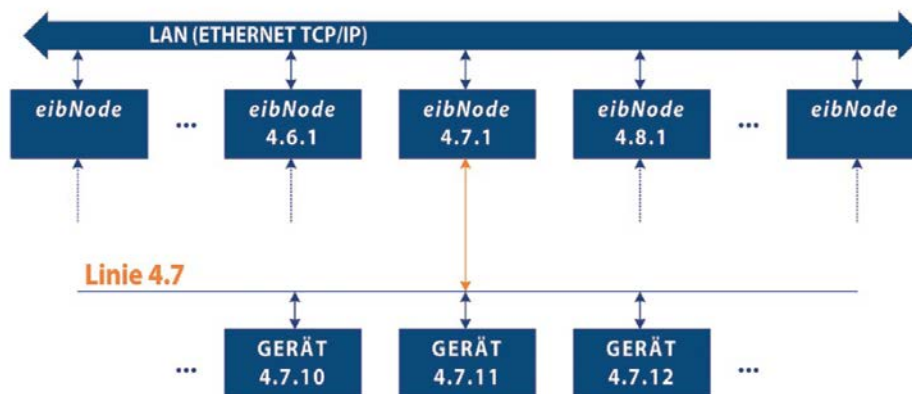


Abbildung 10: Struktur KNX/EIB und LAN - Linie

Die eibNodes hängen alle „parallel“ gleichberechtigt am LAN. Unabhängig von der jeweiligen IP-Adresse besitzt jeder eibNode eine eigene phys. EIB-Adresse, die mit der Programmierung über die ETS bestimmt wird. Sie muss, wie bei Bereichs- und Linienkopplern eindeutig sein. Anders als in der Baumstruktur des EIB existieren hier aber keine Bereichslinien bzw. Hauptlinien.

Im Prinzip arbeiten die eibNodes zur Bestimmung der „Zuständigkeit“ wie Bereichs- und Linienkoppler. Es gilt die Formel:

$$(\text{eigene Adr. or LineMask}) = (\text{Zieladr. or LineMask})$$

Die LineMask ist aber beim eibNode einstellbar. Je nach Einstellung ergibt sich, wie viele Bereichskoppler, Linienkoppler und Geräte unter einem eibNode adressiert werden können. Die LineMask von 255 (00FFh) bildet die Linienstruktur des EIB nach. Es können dann 256 eibNodes eingesetzt werden und jeder eibNode verwaltet eine EIB-Linie von bis zu 254 Geräten (256 Adressen minus Geräteadresse 0 minus Adresse des eibNodes). (Siehe Bild oben). In der Regel kommen 64-2 = 62 Geräte zum Einsatz.

Die LineMask von 4095 (OFFFh) bildet die Bereichsstruktur des EIB nach. Es können bei dieser Einstellung 16 eibNodes eingesetzt werden unter jedem eibNode können 15 Linienkoppler mit jeweils 255 Geräten sowie weitere 239 Geräte (256 Adressen minus Geräteadresse 0 minus Adresse des eibNodes minus 15 Linienkoppleradressen) in der Hauptlinie eingesetzt werden.



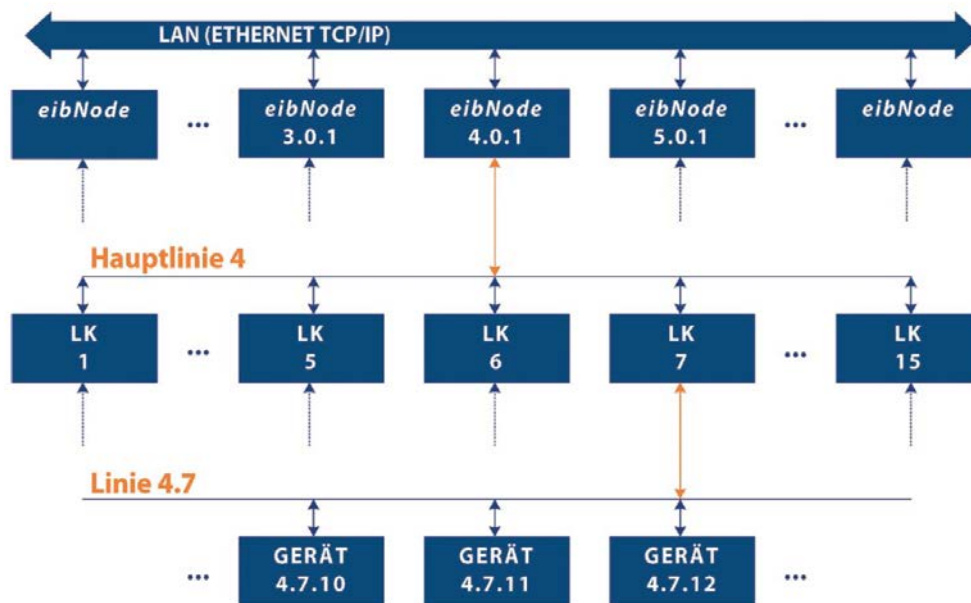


Abbildung 11: Struktur KNX/EIB und LAN - Hauptlinie

Durch die freie Wahl der LineMask sind aber auch andere Aufteilungen möglich (theoretische Werte):

LineMask	Hex	eibNode	Bereichsk.	Linienkpl	Geräte
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

Abbildung 12: Aufteilungen

eibNode = Anzahl der einsetzbaren eibNodes

Bereichskpl = Anzahl der Bereichskoppler pro eibNode

Linekpl = Anzahl der Linienkoppler pro eibNode

Geräte = Anzahl der sonstigen EIB Geräte pro eibNode

Da die eibNodes nicht alle die gleiche LineMask haben müssen ist natürlich auch eine Kombination dieser Tabelle möglich. In Anlagensegmenten mit viel Buslast beispielsweise ersetzt der eibNode den Linienkoppler, in anderen Bereichen den Bereichskoppler oder wird sogar über Bereichskoppler positioniert.

Der Vorteil des eibNodes gegenüber dem klassischen Koppler ist seine höhere Geschwindigkeit im Backbone. Entsprechend sollten Anlagenteile mit höherer Buslast (zyklische Sender, Heizungssteuerung usw.) weniger Geräte unter dem eibNode haben als Anlagenbereiche, die relativ wenig Buslast haben. Die sinnvolle Aufteilung ergibt sich durch eine Analyse der Buslasten.

## 5.5 NETID

---

Wenn man mehrere EIB-Anlagen an einem Ort plant können diese Anlagen über den LAN zusammengeschaltet werden. Jeder eibNode hat eine NetID. Neben den Netzwerkparametern (IP-Adresse, SubNet und UDP-Message Händler) bilden die eibNodes mit der gleichen NetID eine EIB-Anlage. Innerhalb dieses Bereichs darf es, wie in einer klassischen EIB-Anlage, keine Adressüberschneidungen bei den Gruppenadressen geben und die phys. Adressierung muss eindeutig und der Netztopologie angepasst sein.

Die NetID unterscheidet nun unterschiedliche EIB-Anlagen. Der gesamte Adressbereich der Gruppenadressen und der phys. Adressen wird um eine Zahl, der NetID, erweitert. Dennoch kommunizieren alle eibNodes über das gleiche Netzwerk (gleicher UDP-Port).

Eine zentrale Visualisierung oder Ähnliches hat nun mit dem Zugang zum LAN bei entsprechender Berücksichtigung der NetID alle Daten von mehreren EIB-Anlagen und kann auch zu allen Anlagen entsprechende Schalthandlungen senden.

Die NetID ist eine Ein-Byte Zahl von 0 bis 255. Jede Message über den UDP-Message Händler wird mit der NetID des Senders versendet. Ein eibNode als Empfänger behandelt normalerweise nur die Messages mit der eigenen NetID. Eine Ausnahme gibt es lediglich bei bestimmten Group Messages Telegrammen (Schalttelegramme). Dazu aber später mehr.

Da die eibNodes diese eindeutige Zuordnung bei Single Device Telegrammen (Programmiertelegammen) haben, wird jede EIB-Anlage mit einem ETS-Projekt unabhängig von anderen EIB-Anlagen erstellt und programmiert.

Dies hat nicht nur den Vorteil, dass parallel programmiert werden kann, sondern auch die Unabhängigkeit der einzelnen Anlagen. Adressüberschneidungen können nicht vorkommen. Entsprechend brauchen sich die Inbetriebnahme nicht abzugleichen. Die Inbetriebnahme wird kalkulierbarer. Auch zum Nachrüsten von EIB-Anlagen braucht nun nicht mehr das alte ETS-Projekt erweitert zu werden, es kann ein komplett neues Projekt aufgebaut werden. Dennoch sind alle Daten am LAN parallel vorhanden.

Zusätzlich zur NetID können für Group Messages Telegramme (Schalttelegramme) Hauptgruppenbereiche für die eibNodes transparent geschaltet werden. Diese besonderen Zentraladressen werden dann unabhängig ihrer NetID vom Empfänger verarbeitet.



## 6 EIBNODE PARAMETRIERUNG MIT EIBVISION

### 6.1 DAS EIBNODE PARAMETRIERUNGS TOOL

Es existieren 2 Möglichkeiten einen eibNode mit Aufgaben zu versehen. Klassisch können Sie ein eibNode Projekt anlegen, wie unten beschrieben. Alternativ können Sie aber auch ein eibVision Projekt anlegen. Hier haben Sie die Möglichkeit die Logikfunktionen mit Eingängen (z.B. Schaltern) und Ausgängen (z.B. Leuchtmelder) zu belegen umso Ihre Logiken zu testen. Die eibVision verfügt über einen so genannten Testmodus der es ermöglicht eine angeschlossene EIB-Anlage zu simulieren. Vorteil, die Logikfunktionen müssen nicht real beim Kunden getestet werden. Die genaue Beschreibung entnehmen Sie bitte dem eibVision Handbuch.

#### 6.1.1 EIN EIBNODE PROJEKT ANLEGEN

Nach dem Start des Programms erscheint das Programmfenster.

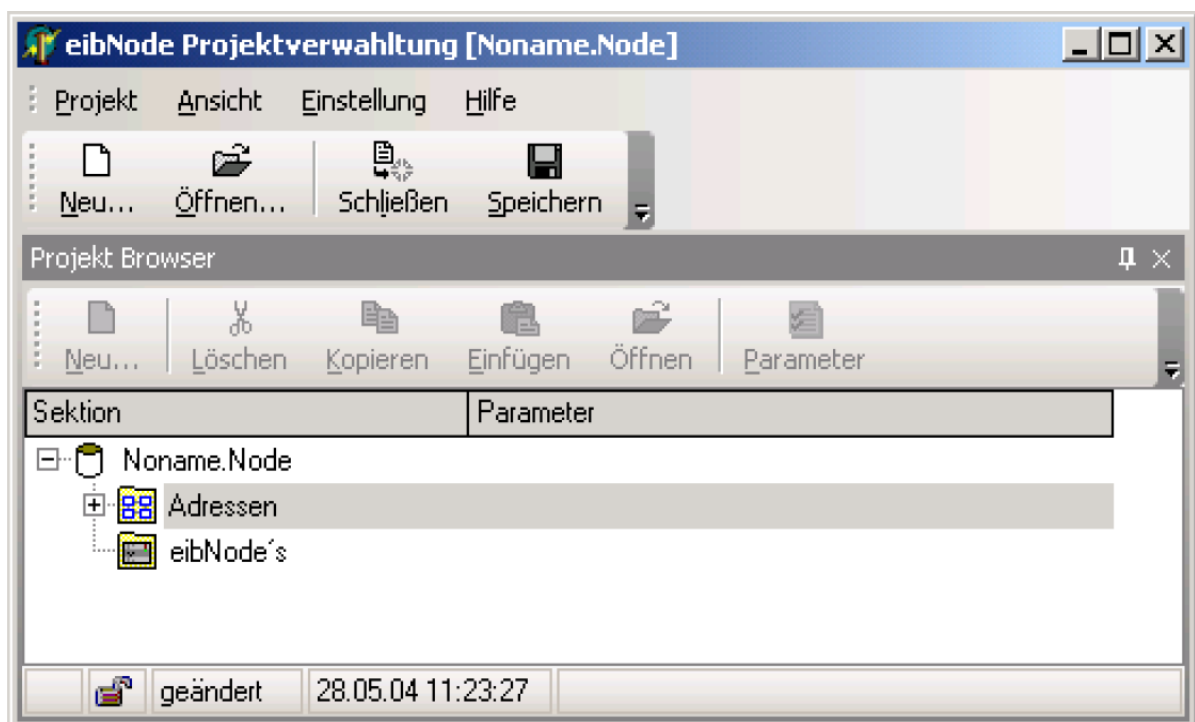


Abbildung 13: eibNode Projektverwaltung

Klicken Sie auf *Neu*. Im Projekt Browser wird ein neues Projekt angelegt. Klicken Sie mit der rechten Maustaste auf den Ordner *eibNodes* im Verzeichnisbaum und wählen Sie *Neu*, um einen eibNode einzufügen.

Es erscheint ein Dialogfenster, in dem die Optionen eingegeben werden:

Geben Sie hier die IP-Adresse des eibNode ein.

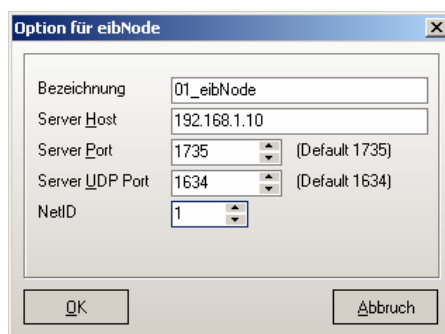


Abbildung 14: eibNode Option

**Anmerkung:** Die Bezeichnung darf nicht länger als 15 Zeichen sein. Alle weiteren Zeichen werden abgeschnitten. Bei Nummerierungen stellen Sie die laufende Nummer am besten der Textbezeichnung voran (z.B. 01\_eibNode, 02\_eibNode usw.).

Server Host: die IP-Adresse oder die DNS-Adresse des eibNode (z.B. IP-Adresse 192.168.1.10) Speichern Sie die Eingaben mit **OK**.

## 6.1.2 FUNKTIONEN PARAMETRIEREN

Mit Doppelklick auf einen angelegten eibNode erscheint das Parametrierungsfenster:

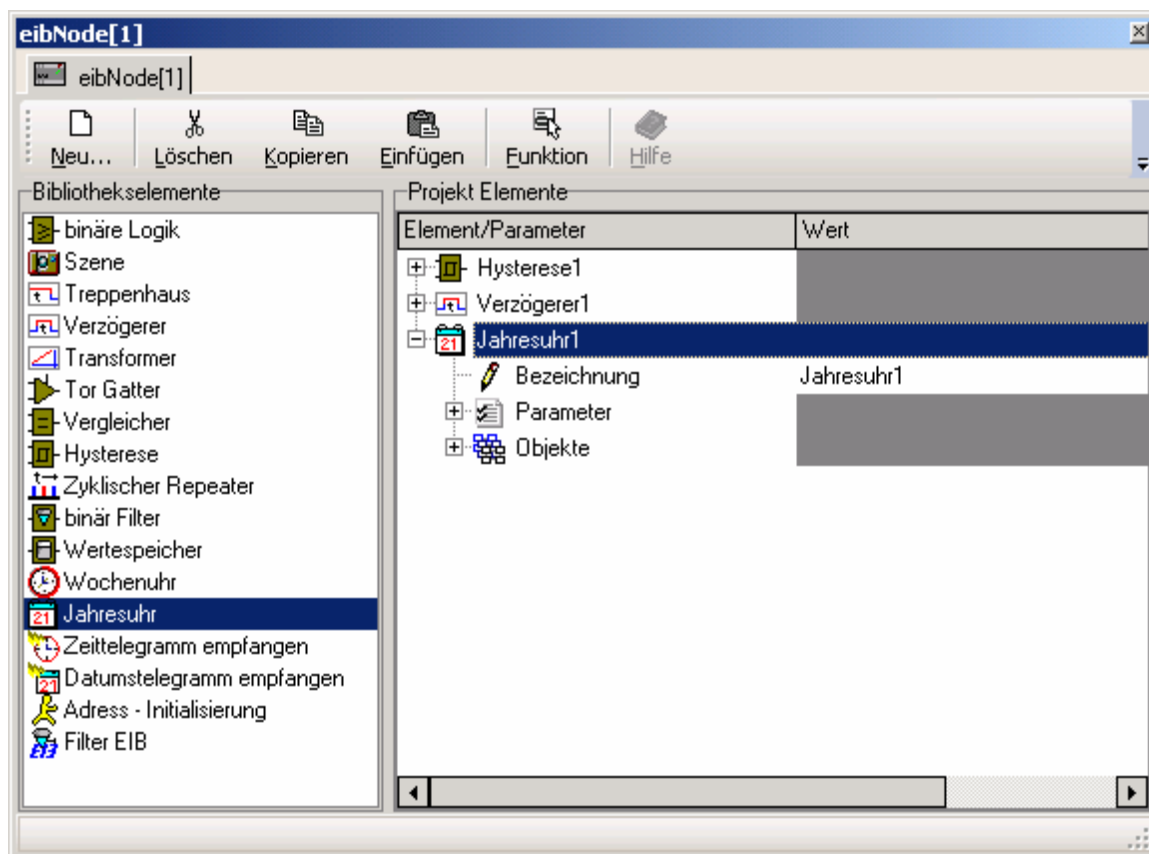


Abbildung 15: Parameter eingeben

Hier werden die Funktionen parametrieren. Im linken Fensterteil befinden sich alle zur Verfügung stehenden Elemente. Im rechten Fensterteil erscheinen alle von Ihnen in das Projekt eingefügte Elemente.

Um ein Element in ein Projekt einzufügen, ziehen Sie es mit gedrückter Maustaste in den rechten Fensterbereich oder markieren Sie ein Element mit der Maus und klicken auf die Schaltfläche **Neu**, bzw.



Klicken Sie mit der rechten Maustaste in den rechten Fensterbereich und wählen *Neu*. Im Feld *Bezeichnung* können Sie dem Element einen Namen geben. Wenn Sie auf die + bzw. – Symbole klicken, werden die Elementoptionen ein/ausgeblendet.

Nun können Sie die Parameter einstellen und die Objekte mit Gruppenadressen versehen. Die Beschreibungen der Elemente finden Sie weiter hinten in dieser Dokumentation.

**Anmerkung:** Die Bezeichnung der Elemente darf nicht länger als 15 Zeichen sein. Alle weiteren Zeichen werden abgeschnitten. Bei Nummerierungen stellen Sie die laufende Nummer am besten der Textbezeichnung voran (z.B. 01\_Szene, 02\_Szene usw.).

### 6.1.3 PROJEKT ÜBERSETZEN UND ÜBERTRAGEN

Wenn Sie alle Funktionen parametrieren haben, kann das Projekt in den eibNode übertragen werden. Klicken Sie im Projektierungsfenster auf Funktion und wählen Sie BMX Compiler.

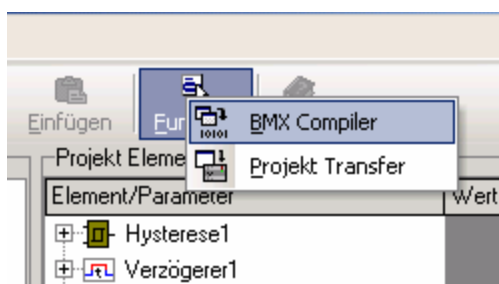


Abbildung 16: BMX Compiler auswählen

Das Projekt wird dadurch übersetzt. Falls Sie bei der Parametrierung notwendige Einstellungen vergessen haben, erhalten Sie entsprechende Fehlermeldungen. Es wird für jedes Element immer nur der erste erkannte Fehler angezeigt. Korrigieren Sie die Einstellungen und kompilieren Sie das Projekt erneut. Unter Umständen müssen Sie diesen Vorgang häufiger wiederholen, bis alle Fehler beseitigt sind. Wenn das Projekt erfolgreich übersetzt wurde, wählen Sie Projekt Transfer zum Übertragen des Projekts in den eibNode. Stellen Sie dabei sicher, dass die Verbindung zwischen dem PC und dem eibNode besteht.

### 6.1.4 PROJEKT SPEICHERN

Projekte können auf dem PC oder einem Datenträger gespeichert werden. Zum Speichern eines Projektes klicken Sie auf die Schaltfläche Speichern, oder wählen Sie Speichern aus dem Menü Projekt, und wählen Sie einen Projektnamen und den Speicherort.

## 6.2 DIE BEFEHLSZENTRALE

Mit der Befehlszentrale kann überprüft werden, ob ein eibNode korrekt arbeitet und Telegramme empfangen/gesendet werden. Zum Aufrufen der Befehlszentrale wählen Sie im Menü Ansicht den Eintrag Befehlszentrale.

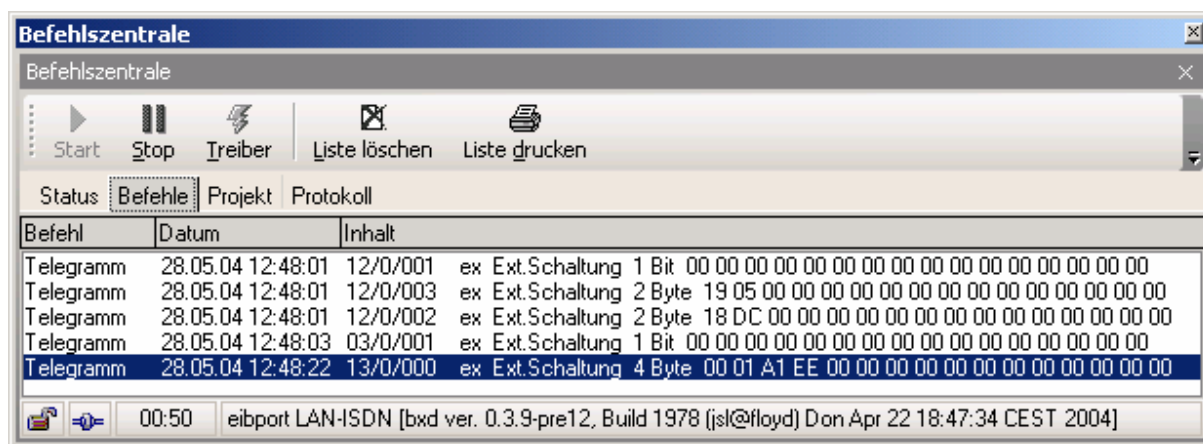


Abbildung 17: Befehlszentrale

Um die Verbindungsdaten einzugeben, klicken Sie auf die Schaltfläche Treiber.

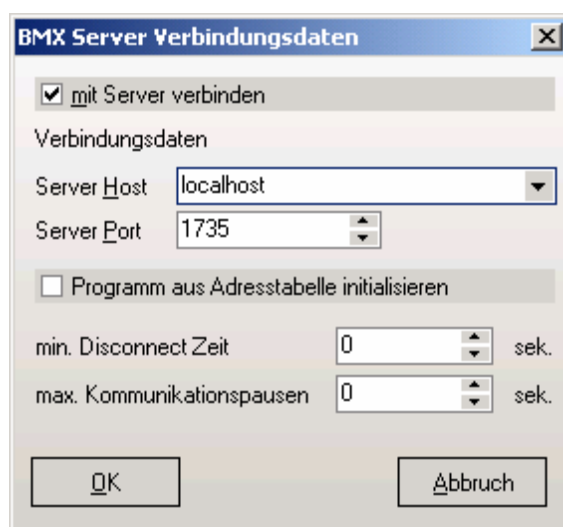


Abbildung 18: BMX Server Verbindungsdaten

Geben Sie hier die Verbindungsdaten des eibNode an.

Mit einem Klick auf die Schaltfläche Start wird die Verbindung aufgebaut.

Unter dem Reiter Befehle werden alle empfangenen und gesendeten Telegramme aufgelistet



## 6.3 DIE ELEMENTE

---

### 6.3.1 BINÄRE LOGIK

---

Mit diesem Element können EIS 1-Telegramme logisch verknüpft werden. Es stehen die binären Logikgatter und, oder, exklusiv oder, Objektwert, nicht und, nicht oder, nicht exklusiv oder, nicht Objektwert zur Verfügung.

#### PARAMETER

---

**Gatterform**

Hier wird die Art des Logikgatters ausgewählt.

**Anzahl Eingänge**

Hier wird die Anzahl der Eingänge angegeben, die logisch verknüpft werden sollen. Es stehen maximal 31 Eingänge zur Verfügung.

**Immer senden**

Wenn true gesetzt ist, wird das Ergebnis jeder logischen Verknüpfung immer über den Ausgang gesendet. Es wird auch gesendet, wenn sich die Zustände an den Eingängen nicht ändern und damit auch der Ausgang nicht ändert.

Wenn false gesetzt ist, wird nur dann gesendet, wenn sich das Ergebnis ändert.

**Filter**

Hier wird angegeben, welche Art von Telegrammen (EIN- und/oder AUS-Telegramme) logisch verknüpft werden sollen.

- EIN u. AUS senden: alle Telegramme werden auf die Eingänge gesendet.
- nur EIN senden: nur EIN-Telegramme werden auf die Eingänge gesendet.
- nur AUS senden: nur AUS-Telegramme werden gesendet.

#### OBJEKTE

---

**Ausgang**

Über das Ausgangsobjekt wird das binäre Ergebnis der logischen Verknüpfung der Eingänge gesendet.

**Eingang x**

Auf die Eingänge werden die 1 Bit-Telegramme (EIS 1) gesendet, die logisch verknüpft werden sollen. Die Anzahl der Eingänge wird mit dem Parameter Anzahl Eingänge bestimmt. Es stehen maximal 31 Eingänge zur Verfügung.

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

### 6.3.2 SZENE

---

In einer Szene können mehrere Telegramme zusammengefasst werden und über ein Start- Telegramm gesendet werden. Szenen können in der Parametrierung festgelegt werden und während der Laufzeit des Programms gespeichert werden. Mit diesem Element können z.B. Lichtszenen erstellt werden, die bestimmte Stimmungen erzeugen.

#### PARAMETER

---

**Anzahl Ausgänge**

Hier wird die Anzahl der Ausgänge angegeben, über die Telegramme gesendet werden. Es stehen maximal 28 Ausgänge zur Verfügung.

**Start bei ein**

- true: die Szene startet, wenn ein EIN-Telegramm auf das Start/Stop-Objekt gesendet wird.
- false: die Szene startet, wenn ein AUS-Telegramm auf das Start/Stop-Objekt wird.

**Stop erlaubt**

- true: die Szene wird gestoppt, wenn ein EIN-Telegramm auf das Start/Stop-Objekt gesendet wird.
- false: die Szene kann nicht gestoppt werden.

**Speichern bei ein**

- true: die aktuellen Zustände der Ausgang-Objekte werden als Szene gespeichert, wenn ein EIN-Telegramm auf das speichern-Objekt gesendet wird.
- false: es kann keine Szene gespeichert werden.

**Anwenderdialog**

Ist true gesetzt, dann kann der Anwender während der Laufzeit des Programms eigene Szenen erstellen.

**Verzögerungszeit (s)**

Hier wird in Sekunden angegeben, mit welcher Verzögerungszeit eine Szene startet.

**Dialog**

Mit einem Klick auf Startwerte im Dialog öffnet sich ein Dialogfenster, in dem die Startwerte der einzelnen Ausgang-Objekte definiert werden. Hier wird eine Szene voreingestellt.

**OBJEKTE**

---

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann **freigegeben**, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Start/Stop**

Über dieses Objekt wird eine Szene gestartet und gestoppt (abhängig von den Parametern).

**Speichern**

Über dieses Objekt werden die Zustände der Ausgang-Objekte als Szene gespeichert.

**Ausgang x**

Über die Ausgang-Objekte werden die in einer Szene gespeicherten Telegramme gesendet.

## 6.3.3 TREPPENHAUS

---

Dieses Element bildet einen Treppenhauslichtautomaten nach. Wenn ein Taster für das Treppenhauslicht betätigt wird, schaltet das Licht ein und nach Ablauf einer bestimmten Zeit wieder aus. Durch erneutes Drücken eines Tasters, noch während das Licht eingeschaltet ist, wird die Einschaltzeit automatisch verlängert. Die Einschaltdauer ist einstellbar. Das Ausschalten der Beleuchtung, noch bevor die Zeit abgelaufen ist, kann eingestellt werden.

**PARAMETER**

---

**Neustart möglich**

- true: durch erneutes Senden eines EIN-Telegramms auf das Eingang-Objekts, wird die Einschaltdauer um die Verzögerungszeit verlängert.
- false: die Verlängerung der Einschaltzeit ist nicht möglich.

**Vorzeitiger Abbruch möglich:**

- True: durch Senden eines AUS-Telegramms auf das Eingang-Objekt wird der Ausgang sofort auf 0 gesetzt.





- false: der vorzeitige Abbruch ist nicht möglich.

**Ausgang invertieren**

Der Zustand des Ausgangs wird invertiert.

- true: ein EIN-Telegramm am Eingang bewirkt ein AUS-Telegramm am Ausgang.
- false: der Ausgang wird nicht invertiert.

**Verzögerungszeit (s)**

Hier wird in Sekunden angegeben, nach welcher Zeit der Zustand am Ausgang geändert wird.

---

**OBJEKTE****Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Über dieses Objekt wird abhängig von den Parametern ein Telegramm gesendet.

**Eingang**

Über dieses Objekt wird abhängig von den Parametern der Treppenhauslichtautomat gestartet/gestoppt.

---

## 6.3.4 VERZÖGERER

Dieses Element verzögert die Weiterleitung eines Telegramms vom Eingang zum Ausgang um die einstellbare Verzögerungszeit.

---

**PARAMETER****Verzögerungszeit (s)**

Hier wird in Sekunden angegeben, nach welcher Zeit ein Telegramm vom Eingang zum Ausgang weitergeleitet wird.

**Verzögerung**

Hier wird angegeben, welche Telegramme verzögert werden.

- Alle: Alle Telegramme werden verzögert.
- AUS Telegramme: nur AUS-Telegramme werden verzögert.
- EIN Telegramme: nur EIN-Telegramme werden verzögert.

---

**OBJEKTE****Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Über dieses Objekt wird abhängig von den Parametern ein Telegramm gesendet.

**Eingang**

Ein Telegramm, das auf dieses Objekt gesendet wird, wird abhängig von den Parametern verzögert zum Ausgang weitergeleitet.

## 6.3.5 TRANSFORMER

---

Dieses Element konvertiert ein Telegramm von einem EIS-Typ in einen anderen EIS-Typ.

### PARAMETER

---

#### Eingang Format

Hier wird angegeben, welches Format (EIS-Typ) das zu konvertierende Telegramm hat.

#### Ausgang Format

Hier wird angegeben in welches Format (EIS-Typ) das Telegramm konvertiert werden soll.

#### Anwenderdialog

Wenn true gesetzt ist, kann der Anwender während der Laufzeit des Programms einige Parameter des Elements ändern.

#### Nur bei Veränderung

- Ist true gesetzt, dann findet die Konvertierung nur dann statt, wenn sich das Eingangstelegramm ändert. Bei false wird jedes Telegramm am Eingang konvertiert.

#### Dialog

Hier lässt sich in einem Dialogfenster die Übertragungsfunktion bearbeiten.

### OBJEKTE

---

#### Freigabe

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

#### Ausgang

Über dieses Objekt wird das konvertierte Telegramm gesendet.

#### Eingang

Auf das Eingang-Objekt wird das zu konvertierende Telegramm gesendet.

## 6.3.6 TOR GATTER

---

Beim Tor Gatter werden Telegramme vom Eingang zum Ausgang wie durch ein Tor weitergeleitet. Durch ein „Öffnen/Schließen“-Telegramm wird das Tor freigegeben oder geschlossen. Die Weiterleitung von Telegrammen lässt sich so steuern.

### PARAMETER

---

#### Tor offen bei ein

- True: das Tor ist beim Senden eines EIN-Telegramms auf das Tor Objekt offen.
- False: das Tor ist beim Senden eines EIN-Telegramms auf das Tor Objekt geschlossen.

#### Tor für beide Richtungen

#### Abgleich bei Torfreigabe

- Wenn true gesetzt ist, überprüft das Element bei Torfreigabe über das Tor Objekt, ob die Zustände am Eingang und Ausgang gleich sind. Sind die Zustände ungleich, dann wird das Telegramm vom Eingang an den Ausgang weitergeleitet.
- Bei false erfolgt kein Abgleich.

#### Nur bei Veränderung

Wenn true gesetzt ist, dann wird nur gesendet, wenn sich der Zustand am Eingang ändert. Wenn false



gesetzt ist, dann wird immer gesendet.

## OBJEKTE

---

### **Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

### **Tor Objekt**

Auf dieses Objekt wird das Telegramm gesendet, das das Tor öffnet/schließt.

### **Ausgang**

Über dieses Objekt werden Telegramme gesendet, die durch das Tor gelaufen sind.

### **Eingang**

Auf das Eingang-Objekt werden die Telegramme gesendet, die durch das Tor laufen sollen.

## 6.3.7 VERGLEICHER

---

Dieses Element vergleicht den Inhalt von zwei Telegrammen oder den Inhalt eines Telegramms mit einem konstanten Wert. Es stehen 5 Vergleichertypen zur Verfügung: gleich, größer, größer gleich, kleiner, kleiner gleich. Als Ergebnis eines Vergleichs sendet das Element ein Telegramm mit dem Inhalt EIN, wenn die gewählte Bedingung erfüllt ist (z.B. gleich), und AUS, wenn die Bedingung nicht erfüllt ist. Es können nur Telegramme des gleichen Formats verglichen werden. D.h., die Eingänge müssen auf den gleichen EIS-Typ parametrierung werden.

## PARAMETER

---

### **Eingang 1 Format, Konstante**

Hier wird der EIS-Typ des Telegramms am Eingang angegeben. Je nach EIS-Typ stehen weitere Parameter, wie Minimum, Maximum, Faktor und Offset, zur Verfügung. Wenn kein EIS-Typ ausgewählt wird (untypisiert), kann eine Konstante angegeben werden.

### **Eingang 2 Format, Konstante**

Hier wird der EIS-Typ des Telegramms am Eingang angegeben. Je nach EIS-Typ stehen weitere Parameter, wie Minimum, Maximum, Faktor und Offset, zur Verfügung. Wenn kein EIS-Typ ausgewählt wird (untypisiert), kann eine Konstante angegeben werden.

### **Nur bei Veränderung**

- Wenn true gesetzt ist, dann werden die Vergleichsergebnisse nur bei Veränderungen an den Eingängen gesendet.
- Wenn false gesetzt ist, wird immer gesendet.

### **Vergleicher Typ**

Hier wird die Art des Vergleichers gewählt. Es stehen die Typen gleich, größer, größer gleich, kleiner, kleiner gleich zur Verfügung.

## OBJEKTE

---

### **Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

### **Ausgang**

Über dieses Objekt werden Telegramme mit dem Ergebnis von Vergleichen gesendet.

### **Eingang 1, Eingang 2**

Auf diese Objekte werden die zu vergleichenden Telegramme gesendet. Soll ein Telegramminhalt mit einer Konstanten verglichen werden, dann muss das Objekt, das mit einer Konstanten parametrier ist, frei bleiben.

## 6.3.8 HYSTERESE

---

Mit diesem Element wird durch Bestimmen von oberer und unterer Schaltschwelle ein Toleranzbereich festgelegt, innerhalb dessen kein Telegramm gesendet wird. Die EIS-Typen von Eingang und Schwellen müssen gleich sein.

Beispiel:

Die Raumbeleuchtung ist mit einem Helligkeitssensor gekoppelt. Wenn ein bestimmter Helligkeitswert unterschritten wird, schaltet die Beleuchtung ein; wird dieser Wert wieder überschritten, schaltet die Beleuchtung aus. Bei ständig wechselnden Helligkeitsverhältnissen (Sonne – Wolken) im Bereich des Schaltwertes führt das zum ständigen Ein- und Ausschalten der Beleuchtung. Um das zu vermeiden, bestimmt man eine Hysterese. Es wird eine untere und eine obere Schaltschwelle definiert. Zwischen den beiden Schaltschwellen gibt es einen Toleranzbereich. Helligkeitsschwankungen innerhalb des Toleranzbereiches lösen kein Schalten aus. Erst beim Unterschreiten der unteren Schaltschwelle schaltet die Beleuchtung ein; beim Überschreiten der oberen Schaltschwelle schaltet die Beleuchtung aus.

### PARAMETER

---

#### Nur bei Veränderung

- Wenn true gesetzt ist, dann wird am Ausgang nur gesendet, wenn sich der Zustand am Eingang ändert. Bei false wird immer gesendet.

#### Ausgang invertieren

- Wenn true gesetzt ist, dann werden die Telegramme am Ausgang invertiert.

#### Eingang

Hier wird der EIS-Typ der Eingangstelegramme gewählt. Je nach EIS-Typ stehen weitere Parameter, wie Faktor, Offset usw., zur Verfügung.

#### obere Schwelle

Hier wird der EIS-Typ der Telegramme für die obere Schwelle angegeben. Bei untypisiert muss ein konstanter Wert eingegeben werden.

#### untere Schwelle

Hier wird der EIS-Typ der Telegramme für die untere Schwelle angegeben. Bei untypisiert muss ein konstanter Wert eingegeben werden.

### OBJEKTE

---

#### Freigabe

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

#### Ausgang

Über den Ausgang werden die Schalttelegramme gesendet.

#### Eingang

Auf dieses Objekt werden die Eingangstelegramme gesendet.

#### Obere Schwelle

Auf dieses Objekt werden die Telegramme mit den Werten für die obere Schwelle gesendet.

#### Untere Schwelle

Auf dieses Objekt werden die Telegramme mit den Werten für die untere Schwelle gesendet.



## 6.3.9 ZYKLISCHER REPEATER

Dieses Element wiederholt Telegramme in einer einstellbaren Intervallzeit. So wird ein Telegramm, das nur einmal auf den Eingang gesendet wird, zyklisch wiederholt und über den Ausgang gesendet.

### PARAMETER

**Verzögerungszeit (s)**

Hier wird die Intervallzeit für das wiederholte Senden der Telegramme eingestellt.

**Wiederhole Zustand =0**

Wenn true gesetzt ist, werden Telegramme mit dem Inhalt 0 wiederholt.

**Wiederhole Zustand <>0**

Wenn true gesetzt ist, werden Telegramme mit dem Inhalt ungleich 0 wiederholt.

**Repeater Objekt**

Wiederhole Eingang: Telegramme vom Eingang-Objekt werden wiederholt

Wiederhole Ausgang: Telegramme vom Ausgang-Objekt werden wiederholt

**Eingang sofort senden**

Wenn true gesetzt ist, dann wird beim Start des Programms sofort gesendet, ohne zuerst die Verzögerungszeit ablaufen zu lassen (z.B. für Testzwecke).

### OBJEKTE

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Abhängig von der Parametrierung werden über dieses Objekt Telegramme wiederholt gesendet.

**Eingang**

An dieses Objekt werden die Telegramme gesendet, die über den Ausgang zyklisch wiederholt werden sollen.

## 6.3.10 BINÄRER FILTER

Mit diesem Element können 1Bit-Telegramme (EIS 1) gefiltert werden. Es werden nur Telegramme mit dem Inhalt 0, bzw. nur Telegramme mit dem Inhalt 1 zum Ausgang weitergeleitet.

### PARAMETER

**Zustand =0 übertragen**

Ist true gesetzt, dann werden alle Telegramme mit dem Inhalt 0 zum Ausgang übertragen. (Der Parameter Zustand <>0 muss auf false gesetzt sein).

**Zustand <>0 übertragen**

Ist true gesetzt, dann werden alle Telegramme mit dem Inhalt 1 zum Ausgang übertragen. (Der Parameter Zustand =0 muss auf false gesetzt sein).

### OBJEKTE

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben,

wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Über das Ausgang-Objekt werden abhängig von den Parametern die ungefilterten Telegramme gesendet.

**Eingang**

Auf das Eingang-Objekt werden die zu filternden 1Bit-Telegramme (EIS 1) gesendet.

## 6.3.11 MATHEMATIK

---

Mit diesem Element können mathematische Operationen (Addieren, Subtrahieren, Multiplizieren, dividieren) durchgeführt werden. Hierzu können alle EIS-Typen verwendet werden.

**PARAMETER**

---

**Funktion**

Hier wird die Art der mathematischen Operation gewählt (Addieren, Subtrahieren, Multiplizieren, dividieren).

**Anzahl Eingänge**

Hier wird die Anzahl der Eingänge angegeben. Es stehen maximal 10 Eingänge zur Verfügung.

**Immer senden**

Wenn true gesetzt ist, wird das Ergebnis jeder mathematischen Verknüpfung immer über den Ausgang gesendet. Es wird auch gesendet, wenn sich die Zustände an den Eingängen nicht ändern und damit auch der Ausgang nicht ändert. Wenn false gesetzt ist, wird nur dann gesendet, wenn sich das Ergebnis ändert.

**Ausgang**

Hier wird das EIS-Format des Ausgangs parametrierbar. Je nach EIS-Format stehen die Parameter Minimum, Maximum, Faktor und Offset zur Verfügung.

**Eingang x**

Hier wird das EIS-Format des Eingangs x parametrierbar. Je nach EIS-Format stehen die Parameter Minimum, Maximum, Faktor und Offset zur Verfügung.

**OBJEKTE**

---

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Über das Ausgang-Objekt werden abhängig von den Parametern die Ergebnisse von mathematischen Operationen gesendet.

**Eingang**

Auf das Eingang-Objekt werden die mathematisch zu verknüpfenden Telegramme gesendet.



## 6.3.12 WERTESPEICHER

---

Das Element Wertespeicher besitzt zwei Speicher für Telegramminhalte. Sie haben die Bezeichnungen *Speicher* und *Default Speicher*. Der *Default Speicher* wird bei der Parametrierung des Elementes definiert und kann im Laufe des Programms nicht verändert werden. Der Speicher kann hingegen durch ein Speicher Ereignis mit einem neuen Telegramminhalt beschrieben werden. Beide Speicherinhalte können durch entsprechende Ereignisse gesendet werden.

### Typisches Beispiel:

Für eine Einzelraum Heizungsregelung wird der Sollwert über EIS6 8 Bit gefahren. Beim Öffnen der Fenster soll die Heizregelung ausgeschaltet werden und beim Schließen des Fensters soll als Sollwert der letzte Sollwert vor dem Öffnen des Fensters wieder eingestellt werden. Das Fenster bekommt einen Fensterkontakt, der EIN sendet beim Öffnen und AUS sendet beim Schließen des Fensters. Der *Default Speicher* wird beim Parametrieren auf EIS6 mit Inhalt 0 gesetzt. Als *Speicher Ereignis* wird die *Flanke Ein* parametriert. Das *Default Sende Ereignis* wird auf Ein und das *Sende Ereignis* auf Aus parametriert. Wird nun das Fenster geöffnet sendet der Kontakt ein EIN Telegramm. Da das Fenster vorher geschlossen sein musste, erkennt der Wertespeicher eine *Flanke Ein* und speichert den aktuellen Sollwert und sendet, da auch das *Default Sende Ereignis* erkannt wird, den Inhalt des *Default Speichers*, ein EIS6 mit Inhalt 0, so dass die Heizung ausgeschaltet wird. Wird nun das Fenster wieder geschlossen, so wird das *Sende Ereignis* mit dem AUS Telegramm erkannt, und der gespeicherte EIS6 Inhalt wird am Ausgang gesendet.

### PARAMETER

---

#### Default Sende Ereignis

Das Default Sende Ereignis leitet sich aus dem Objekt Zustand und dem Empfang von Telegrammen am Eingangsobjekt ab. Tritt das Ereignis ein, so wird der parametrierte Inhalt des Default Sendewerts am Ausgang gesendet. Folgende Möglichkeiten stehen zur Verfügung:

- **keine:** Das Ereignis tritt nie ein.
- **Flanke Ein:** Nach Empfang eines EIN Telegramms wenn der aktuelle Objektwert AUS ist.
- **Flanke Aus:** Nach Empfang eines AUS Telegramms wenn der aktuelle Objektwert EIN ist.
- **Ein:** Nach Empfang eines EIN Telegramms.
- **Aus:** Nach Empfang eines AUS Telegramms.
- 

#### Default Sendewert

Über ein Dialog wird der *Default Sendewert* definiert, der zum Ereignis *Default Sende Ereignis* am Ausgang gesendet wird. Definiert wird der Telegramminhalt (Datenformat und Dateninhalt). Es ist wichtig das der Default Sendewert oder die Speicher Vorbelegung definiert wird, da das Element dadurch das Datenformat des Ausgangs erfährt.

#### Sende Ereignis

Das Sende Ereignis ist wie das Default Sende Ereignis definiert. Zu Diesem Ereignis wird der Inhalt des Speichers

#### Speicher Ereignis

Das Speicher Ereignis ist wie das Default Sende Ereignis definiert. Zu diesem Ereignis wird der Objektzustand des Ausgangsobjektes in den Speicher übernommen.

#### Speicher Vorbelegung

Über ein Dialog wird die Speicherbelegung zum Start des Elementes definiert. Sollte kein Speicher Ereignis eintreten, bevor das Sende Ereignis eintritt, wird dieser Wert am Ausgang gesendet. Es ist wichtig das der Default Sendewert oder die Speicher Vorbelegung definiert wird, da das Element dadurch das Datenformat des Ausgangs erfährt.

### OBJEKTE

---

#### Ausgang

Über das Ausgangsobjekt sendet das Element nach den entsprechenden Ereignissen und erfährt auch zum Speicher Ereignis des zu Speichernden Wert.

## Eingang

Über das Eingangsobjekt werden die Ereignisse abgeleitet. Es ist ein EIS1 1 Bit Objekt.

## Freigabe

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

## 6.3.13 WOCHENUHR

Mit diesem Element lassen sich für eine Woche Zeiten festlegen, zu denen automatisch bestimmte Telegramme gesendet werden. So kann z.B. morgens die Beleuchtung eingeschaltet und abends ausgeschaltet werden. Dabei können auch Sondertage (z.B. Feiertage) berücksichtigt werden. Sondertage werden zusätzlich mit der Jahresuhr parametriert.

### PARAMETER

#### Anwenderdialog

Wenn true gesetzt ist, kann der Anwender während der Laufzeit die Schaltzeiten ändern.

#### Mit Sondertagen

Wenn true gesetzt ist, können zusätzlich zu den Wochentagen Schaltzeiten für bis zu drei Sondertage (z.B. Feiertage) bestimmt werden. Sondertage werden zusätzlich mit der Jahresuhr parametriert.

#### Sende bei Start

Wenn true gesetzt ist, dann wird beim Start des Programms das Ausgangstelegramm gesendet (z.B. für Testzwecke).

#### Überschreibbar

- Ist false gesetzt, dann kann ein Anwender die Funktion der Uhr manuell außer Kraft setzen. Manuelles Schalten ist möglich.
- Bei true ist manuelles Schalten nicht möglich.

#### Dialog

Im Dialog werden die Schaltzeiten bestimmt. Die Parametrierung erfolgt grafisch mit der Maus. Stand:01.06.04

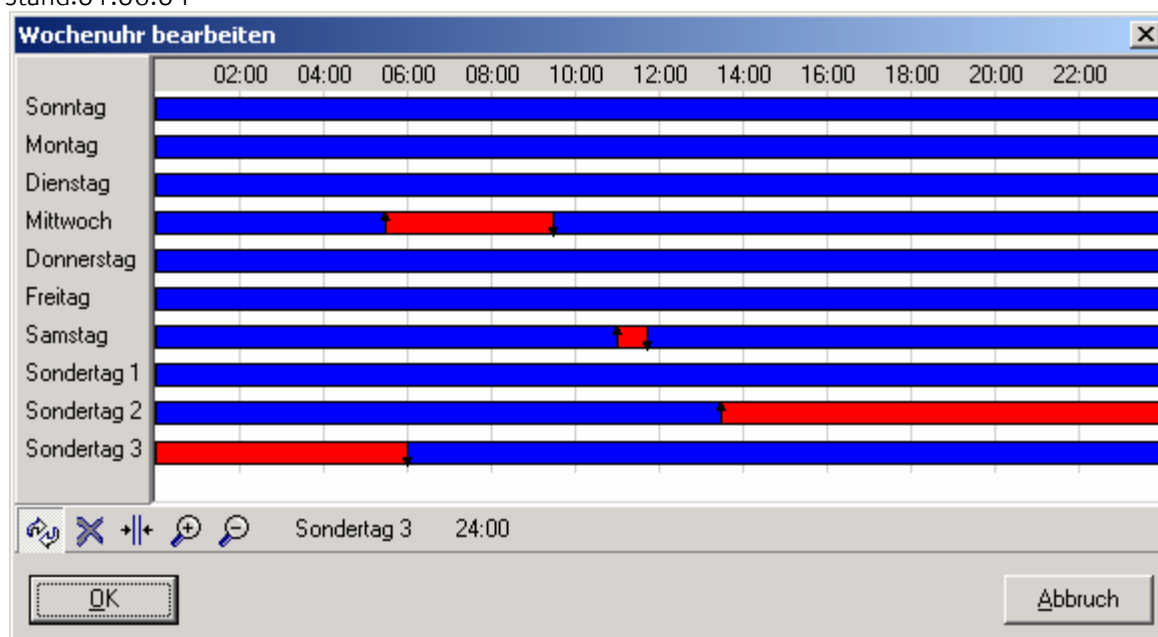


Abbildung 19: Schaltzeiten





Schaltzeiten bestimmen: ein Mausklick in eine blau markierte Zeile bestimmt den Einschaltzeitpunkt (Pfeil nach oben). Ein weiterer Mausklick bestimmt den Ausschaltzeitpunkt (Pfeil nach unten). Schaltzeiten werden rot markiert.

Schaltzeitpunkt löschen: mit einem Mausklick auf einen Schaltzeitpunkt (Pfeil) wird dieser gelöscht.

Schaltzeitpunkt verschieben: mit gedrückter Maustaste können Schaltzeitpunkte verschoben werden.

Ansicht vergrößern/verkleinern: zum Vergrößern mit gedrückter Maustaste den gewünschten Zeitbereich markieren; zum Verkleinern einfach in die Zeitleiste klicken.

#### **Bei Modiwechsel senden**

Dieser Parameter erscheint nur, wenn mit Sondertagen aktiviert ist. Bei true wird sofort zum Beginn eines Sondertages der parametrierte Schaltzustand gesendet. Bei false wird erst zu den parametrierten Schaltzeitpunkten gesendet.

### OBJEKTE

---

#### **Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

#### **Ausgang**

Über das Ausgang-Objekt werden die Schalttelegramme zu den parametrierten Zeiten gesendet.

## 6.3.14 JAHRESUHR

---

Mit diesem Element lassen sich für ein Jahr Zeiten festlegen, zu denen automatisch bestimmte Telegramme gesendet werden.

### PARAMETER

---

Wenn true gesetzt ist, kann der Anwender während der Laufzeit des Programms die Schaltzeiten ändern.

#### **Binär Modus**

##### **Sende bei Start**

Wenn true gesetzt ist, dann wird beim Start des Programms das Ausgangstelegramm gesendet (z.B. für Testzwecke).

#### **Überschreibbar**

- Ist true gesetzt, dann kann ein Anwender die Funktion der Uhr manuell außer Kraft setzen. Manuelles Schalten ist möglich.
- Bei false ist manuelles Schalten nicht möglich.

#### **Dialog**

Im Dialog werden die Schaltzeiten bestimmt. Die Parametrierung erfolgt grafisch mit der Maus.

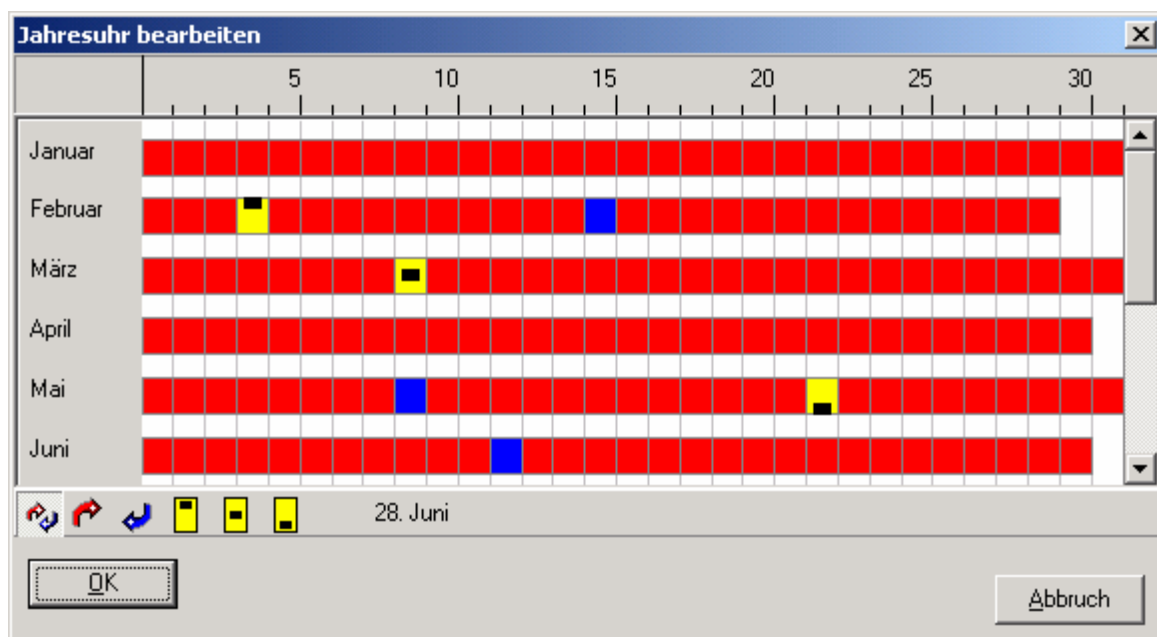


Abbildung 20: Jahresuhr

Für jeden Tag im Jahr können Schaltzeiten bestimmt werden. Rot markierte Tage bedeuten „ausgeschaltet“; blau markierte Tage bedeuten „eingeschaltet“. Diese Symbole stehen für die drei Sondertage aus dem Element Wochenuhr. Wenn ein Tag in der Jahresuhr so markiert ist, dann gelten die in der Wochenuhr unter Sondertag 1 – 3 parametrierten Schaltzeiten. Das erste Symbol steht für Sondertag 1, das zweite für Sondertag 2, das dritte für Sondertag 3.

## OBJEKTE

### Freigabe

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

### Ausgang

Über das Ausgang-Objekt werden die Schalttelegramme zu den parametrierten Zeiten gesendet.

## 6.3.15 ZEITTELEGRAMM SENDEN

Mit diesem Element können Uhrzeit-Telegramme gesendet werden (EIS 3).

## PARAMETER

### Mit Wochentag

Ist true gesetzt, dann wird zusätzlich zu der Uhrzeit auch der Wochentag gesendet.

### Verzögerungszeit (s)

Hier wird in Sekunden die Zeit angegeben nach deren Ablauf das Zeittelegramm gesendet wird.

## OBJEKTE

### Freigabe

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

### Ausgang



Von diesem Objekt wird das Zeitlegramm gesendet.

## 6.3.16 DATUMSTELEGRAMM SENDEN

---

Mit diesem Element können Datums-Telegramme gesendet werden (EIS 4).

### PARAMETER

---

#### **Verzögerungszeit (s)**

Hier wird in Sekunden die Zeit angegeben nach deren Ablauf das Zeitlegramm gesendet wird.

### OBJEKTE

---

#### **Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

#### **Ausgang**

Von diesem Objekt wird das Datumstelegramm gesendet.

## 6.3.17 ZYKLISCHER SENDER

---

Der zyklische Sender ist ein binäres Element, das wie ein Metronom in regelmäßigen zeitlichen Abstand Telegramme erzeugen kann. Das Element wird in der Regel zur Synchronisation von logischen Gattern genutzt. Es hat ein Freigabe- und ein Ausgangsobjekt. Nach Ablauf einer Verzögerungszeit wird am Ausgang ein EIS1 1Bit „Aus“, ein EIS1 1Bit „Ein“ oder der invertierte aktuelle Zustand (Toggeln) gesendet.

**Achtung: Das Zeitraster wird über einen Windows internen Zeittakt berechnet, der nicht Uhrzeitsynchron und Prozessorauslastungsabhängig ist. Das Zeitraster ist also nicht Uhrgenau sondern jeweils +/- eine Sekunde.**

Das Element besitzt neben einem Ausgangsobjekt auch ein Freigabeobjekt. Wenn das Freigabeobjekt keine Adresse besitzt, so gilt das Element immer als freigegeben. Wenn das Freigabeobjekt eine Adresse besitzt, so muss der Objektzustand ungleich 0 sein um die Freigabe zu erhalten.

### PARAMETER

---

#### **Verzögerungszeit (s)**

Die Verzögerungszeit ist die Zeit in Sekunden, die zwischen zwei Sendeaktionen ablaufen muss. Eine Verzögerungszeit von „0“ Sekunden ist nicht erlaubt.

#### **Senden**

Für die Sendefunktion stehen die Möglichkeiten: Nur Aus, Nur Ein und Toggeln zur Verfügung. Beim Toggeln wird der Ausgangsobjektzustand invertiert gesendet. Sollte das Element am Ausgang ein Telegramm empfangen, so wird als nächstes Telegramm beim Toggeln der invertierte Zustand dieses Telegramms gesendet.

#### **Rücksetzen bei Freigabe**

Normalerweise taktet das Element seine Verzögerungszeit immer durch, so dass das Zeitraster immer eingehalten wird, egal ob dem Element eine Freigabe erteilt wurde oder nicht. Ist das Element über das Freigabeobjekt aktiviert, so wird jeweils nach Ablauf der Verzögerungszeit ein Telegramm gesendet. Ist die Freigabe nicht erteilt, so wird lediglich die Sendefunktion unterdrückt. Der Zeittakt arbeitet aber kontinuierlich weiter. Wird dem Element die Freigabe erteilt, so bleibt der Zeittakt erhalten, unabhängig wann die Freigabe erteilt wurde. Ist das Flag für das Rücksetzen bei Freigabe gesetzt, so wird die Verzögerungszeit nach erteilter Freigabe unabhängig des aktuellen Zeitrasters neu gestartet. Dadurch wird sichergestellt, dass nach der Freigabe nach Ablauf der vollständigen Verzögerungszeit das nächste Telegramm gesendet wird.

**Erweiterte Anzeige**

Dieser Parameter hat keine Funktion.

**OBJEKTE**

---

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Ausgang**

Auf dieses Objekt wird das Telegramm zyklisch gesendet.

## 6.3.18 ZEITTELEGRAMM EMPFANGEN

---

Mit diesem Element können Uhrzeit-Telegramme empfangen werden (EIS 3). Das Element vergleicht die interne Uhr des eibNode mit der Uhrzeit, die von einer EIB-Uhr auf den Bus gesendet wird. Bei einer zu großen Abweichung wird die interne Uhr des eibNode neu gestellt.

**PARAMETER**

---

**Datum berücksichtigen**

Ist true gesetzt, dann wird zusätzlich zu der Uhrzeit auch das Datum empfangen.

**Max. Delta (s)**

Hier wird in Sekunden die maximale Differenz zwischen der internen Uhrzeit des eibNode und der Bus-Uhrzeit angegeben. Ist die Differenz größer, dann wird die interne Uhr des eibNode neu gestellt. Da EIB-Uhren in der Regel alle 60 Sekunden die aktuelle Zeit auf den Bus senden, sollte max. Delta größer als 60 Sekunden sein.

**OBJEKTE**

---

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben, wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Eingang**

Auf dieses Objekt wird das Zeittelegramm gesendet.

## 6.3.19 DATUMSTELEGRAMM EMPFANGEN

---

Mit diesem Element können Datumstelegramme empfangen werden (EIS 4). Das Element vergleicht das interne Datum des eibNode mit dem Datum, das von einer EIB-Uhr auf den Bus gesendet wird. Bei Abweichungen wird das interne Datum des eibNode automatisch korrigiert.

**PARAMETER**

---

**Max. Delta (s)**

Dieser Parameter hat hier keine sinnvolle Funktion.

**OBJEKTE**

---

**Freigabe**

Über das Freigabeobjekt kann die Freigabe gesteuert werden. Wird keine Adresse eingetragen, so ist das Element immer freigegeben. Ist eine Adresse eingetragen, so ist das Element nur dann freigegeben,



wenn der Objektzustand ungleich Null ist. Wenn das Element nicht freigegeben ist, ignoriert es alle Telegramme im Eingang.

**Eingang**

Auf dieses Objekt wird das Datumstelegramm gesendet.

## 6.3.20 ADRESSEN - INITIALISIERUNG

---

Mit diesem Element können Gruppenadressen mit einem bestimmten Zustand vorbelegt (initialisiert) werden. Wenn eine EIB-Anlage in Betrieb genommen wird, sind die Zustände der Gruppenadressen zunächst nicht bekannt. Um eventuelle Fehlfunktionen oder ungewollte Funktionen bei der Inbetriebnahme zu vermeiden, können mit diesem Element Zustände vordefiniert werden.

**PARAMETER**

---

**Anzahl Ausgänge**

Entsprechend der Anzahl der Gruppenadressen, die initialisiert werden sollen, wird hier die Anzahl der Ausgang-Objekte angegeben.

**Dialog**

Hier öffnet sich eine Dialogbox, in der die Zustandswerte definiert werden, mit denen die Ausgang-Objekte initialisiert werden.

**OBJEKTE**

---

**Ausgänge**

Über die Ausgang-Objekte werden die Initialisierungstelegramme für die Gruppenadressen gesendet.

## 6.3.21 FILTER EIB

---

Mit diesem Element werden Filtertabellen bearbeitet. In Filtertabellen wird festgelegt, welche Gruppenadresstelegramme vom eibNode Gateway durchgelassen werden und welche gesperrt werden. So werden z.B. Telegramme an Gruppenadressen, die nicht unterhalb eines eibNode liegen, blockiert. Unnötiger Datenverkehr wird vermieden. Filtertabellen werden grafisch mit der Maus bearbeitet. Deshalb gibt es keine Objekte bei diesem Element.

**PARAMETER**

---

**Dialog**

Im Dialog werden die Gruppenadressen festgelegt, die eibNode sperren/durchlassen soll.

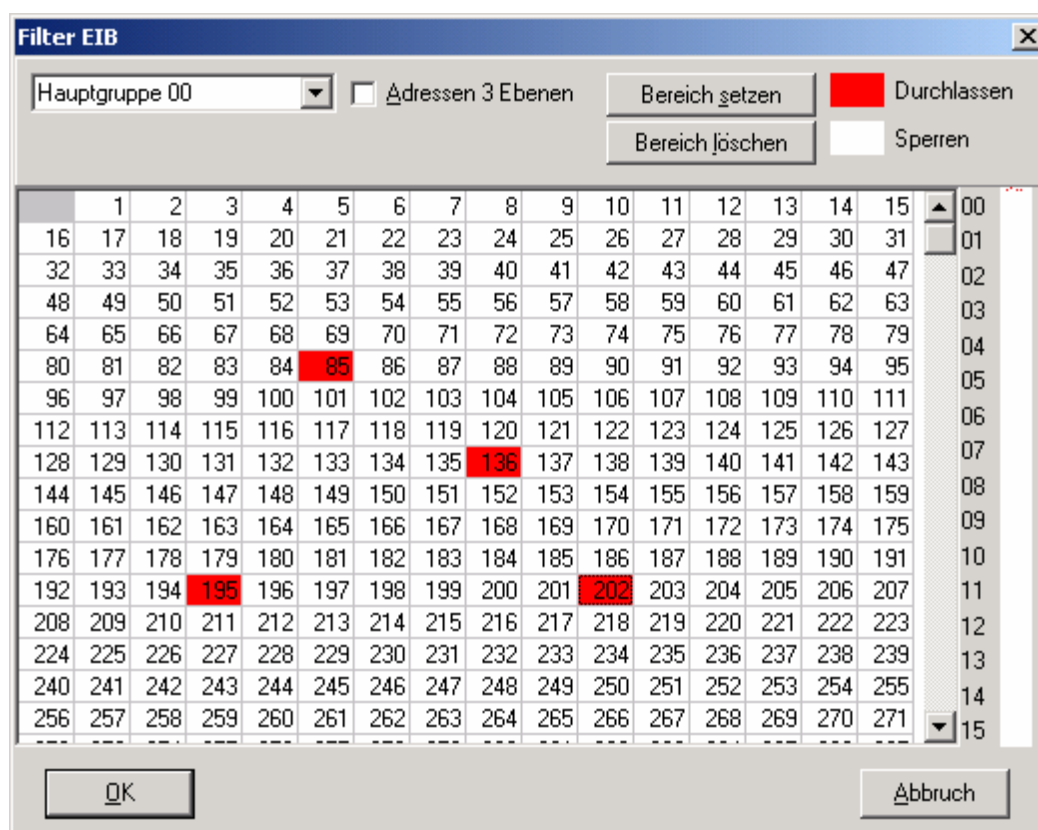


Abbildung 21: Filter EIB

Einzelne Gruppenadressen werden mit der Maus markiert. Um die Gruppenadressen eines ganzen Bereiches zu sperren bzw. frei zu geben, klicken Sie auf Bereich löschen bzw. Bereich setzen.



## 7 ANHANG

### STATUS LED'S

Der momentane Betriebszustand des eibNodes wird durch die an der Frontseite des Gerätes platzierten LED's signalisiert. Hierbei zeigen die LED's folgende Zustände:

#### Bootphase:

Während der Bootphase leuchtet die POWER-LED ca. 30 Sekunden orange und die Programmier LED für den EIB ist aktiv. Das Ende der Bootphase wird durch einen Test aller LED's signalisiert. Die POWER-LED leuchtet dann grün.

Wenn alle LED's im Sekundentakt im Gleichtakt blinken liegt ein Hardwarefehler vor.

#### POWER-LED

Aus	Das Gerät ist nicht betriebsbereit. Es liegt keine Betriebsspannung an.
Rot	Nicht belegt / Hardwaredefekt
Grün	Das Gerät ist betriebsbereit und hat sich konfiguriert.
Orange	Der eibNode befindet sich in der Bootphase ( ca. 30 Sekunden)
Blinken	Sekundentakt: Die BCU ist noch nicht mit der ETS konfiguriert.
Blinken	Alle 2 Sekunden: Die BCU ist mit unlogischen Daten konfiguriert worden. (z.B. unsinnige IP-Parameter)

#### BMX-LED

Aus	Kein Datenaustausch mit dem eibNode Programmkernel
Rot	Telegramm über den UDP-Message Händler ausgetauscht.
Grün	Eine IP-Verbindung zum eibNode Programmkernel ist aufgebaut.
Orange	Datenaustausch über eine IP-Verbindung wird angezeigt

#### EIB-LED

Aus	Der eibNode hat keine Verbindung zum EIB
Rot	Blinkend; der eibNode versucht Kontakt zum EIB aufzubauen
Grün	Der eibNode hat Kontakt zum EIB. (Wird zyklisch überwacht.)
Orange	Der eibNode tauscht Daten mit dem EIB aus.

#### LAN-LED

Aus	Der eibNode hat keine Verbindung zum LAN
Rot	Nicht belegt
Grün	Der eibNode hat eine physikalische Verbindung zum LAN
Orange	Der eibNode tauscht Daten mit dem LAN aus.



---

## TECHNISCHE DATEN

---

Betriebsspannung:	10-30V DC
Leistungsaufnahme:	<= 5VA
Busspannung:	über den EIB

### **Wichtig:**

**Die gedrosselte EIB-Busspannung darf nicht als Betriebsspannung verwendet werden!**

Der eibNode besitzt zur Datenpufferung der internen Hardwareuhr einen Gold-Cap Kondensator. Dieser garantiert eine Pufferung bei Betriebsspannungsausfall von ca. 1/2 Stunde.

Klimabestätigung:	nach EN50090-2-2
Umgebungstemperatur:	0-30 °C
Lagertemperatur:	-20 - +70 °C
Rel. Feuchte (nicht kondensierend):	5% - 80%

### **Mechanische Daten:**

Maße: Reiheneinbaugerät:	(Breite 6TE á 18mm)
Gehäuse:	Kunststoff
Gewicht:	ca. 0,4 kg
Montage:	Hutschiene nach EN 50022-35x7,5
Schutzart:	IP20 (nach EN60529)