EIB-ADAPTER V2

otte
ELEKTRONIK

Copyrights

Stand of the documentation 01/2004

The information and details in this document and in the software have been arranged according to best knowledge and with greatest care and been checked for correctness according to the state-of-the-art. Divergent details nevertheless cannot be excluded between the document and the product with a last safety. Otte Elektronik GmbH assumes no liability for faults or resultant damages. No liability is assumed also for damages which result from the use of the device, the use of applications or faulty circuits in the device. Otte Elektronik GmbH reserves the right to change, to complete or to improve the document or the product without a previous announcement.

Trademark

All product names in this document can be registered trademarks. All trademarks in this document are used only for the identification of the respective product.

Hiytory of changes

Date	Version	Term of change
April 2002	1.0	First translation form german version
September 2003	1.1	complete revision for the ***actual Softwareversion of ProPB***
Dezember 2003	2.0	Extended Functionality Firmware V10, changed hardware
Januar 2004	2.1	Firmwareversion V11

Table of Contents

Safety instructions

During projecting, installation, maintenance and starting the safety and accident prevention regulations valid for the specific case of operation must be heeded. Among other things the following regulations are valid:

VDE 0100 **Regulation for setting up electric power plants to 1000 volts.**

VDE 0113 **Electrical plants with electronic operating supplies**

VDE 0160 **Equipment of electric power plants and electrical operating supplies**

Accident prevention regulations, particularly VBG4, EN regulations, national standards, regulations of the operator etc..

Attention:

The EIB-Adapter in connection with the touch display isn't permitted as exclusive hardware for the control and supervision of essential and life-support systems!

The company Otte Elektronik GmbH isn't liable for the function of the EIB-Adapter in interplay with the displays within the EIB system because the function ability is influenced by projecting the EIB system!

Introduction

Read through these instructions carefully and follow the safety labelings.

Design of these instructions

The manual lying in front of you is built up so that also experienced PC users and beginners can both be successful.

In the chapter "Installation" you learn how EIB-Adapters have to be connected to the display.

In the chapter "Logic function" we introduce to you considering on a few small examples how you can save money with the display. In the following chapter "alarms" we show you how you can program an alarm with the display easily.

In the chapter "Special features" and "Some tips" we would like to point out some problems to you which occur again and again while projecting the display.

We have done our best to write this manual in a simple way so that you don't put it away immediately. It shall accompany you at daily work during projecting. You are supported by various screenshots which clarify the individual work steps. Perhaps the screenshots don't correspond to your software version any more. However, it should be possible to recover the similarity in the respectively current version.

How you can see on the basis of the size of the orginal documentation and the size of this documentation, we can’t explain all functions to you in detail. We have therefore always given you the essential digits of the orginal manuals where you can get more information. Additionally you shouldn’t put away too far the following manuals and should leaf through them if possible.

- Operation Manual
- PLC Connection Manual (chapter 1.1, 1.2)

For suggestions and constructive criticism please contact us.

Components of the EIB-Adapter

The EIB-Adapter is delivered with the following individual components. Please check the delivering package for completeness.

- the EIB-Adapter
  - a cable with a 25-pin DSUB Connector for GP2000-Series or
  - a cable with a 9-pin DSUB Connector for SMARTOUCH-Series
- a driver disk
- a piece of recloseble fastening tape

For the complete use of the EIB-Adapter you still need the following components:

- any touch display of the Pro-face product series
- an EIB system.
- a 24 V power supply with a sufficient power (depending on the display type between 12 Watt and 50 Watt)
- the programming software ProPB for Windows in the current version (only for projecting)

Please contact your distributor or us if parts should be missing.

Otte Elektronik GmbH
Hannoversche Strasse 66
38116 Braunschweig
Deutschland
Tel. : (+49) 0531 / 25 12 7-10
Fax. : (+49) 0531 / 25 12 7-18
Email: Info@otte-elektronik.de
http://www.otte-elektronik.de

The touch display and EIB

The touch displays can send telegrams and reports as well as switching states.

In connection with the software the touch display is programmed so that control and monitoring elements are displayed on the screen of the touch display.

Addionally the software allows the construction of ground plan views, alarm and reporting windows, bar view and numeric displays. The touch display becomes a high-quality module and console.

With the so-called "D-Scripts" it is possible to implement almost every logic in the touch display.

Functionality of the EIB-Adapter

The EIB-Adapter is the direct connection between EIB and display. A special BCU is integrated in the EIB adapter. The EIB is galvanically isolated from the display. A serial interface isn't necessary.

In the EIB-Adapter the requests of the display are transferred in the languages of the EIB. In doing so the adapter minimizes all read requests of the display to a minimum. The adapater asks only for telegrams on the bus which it doesn't know yet. For this the EIB-Adapter can build up internally a copy for all group addresses. During operating the EIB-Adapter permanently listen to the bus (to the telegrams Value-Write and Value-Confirm).

Write requests are sent to the bus directly and put down in the memory area.

The software

With the software the touch display is programmed and configured on a standard PC.

The graphic surface of the software is easy to dominate and the programming can be done largely with the mouse.

A programming language doesn't need to be learned, but knowledges about the EIB and the EIB-software ETS are required.

For the programming of the "D-Scripts" basic knowledges in programming can be useful but are not mandatory.

Compatibility

The EIB-Adapter is designed for the Pro-face displays of the series GPx77, GP2000 and SMARTOUCH. For these devices you have to install the enclosed driver. If you want to use the EIB-Adapter with the displays of the GPx70 series you can switch the EIB-Adapter in the “Translator”-mode. For this set the address D255200 to 1, for reset to 0.

For the communication with the display the EIB-Adapter for the GPx70 series uses the Siemens S5-3964R driver with the communication adjustment 19200, 8, n, 1. The assignment of the group addresses to the X-addresses is identical to the EIB-Translator.

For further information please ask for our **EIB-Translator**-Manual.

Installation

The wiring of the EIB-Adapter

- Connect the touch display to the DC24V output of your power supply.
- Connect the enclosed cbale to the EIB-Adapter an. (RJ12-Connector)
- For the SMARTOUCH-Display you need the 9-pin DSUB-Connector. From this connector you have to connect the black cable to the -24V of the SMARTOUCH Poer-Connectot and the red cable to the +24V.
- Connect the 25-pin connector, resp. the 9-pin connector to the connector of the Display. The power for the EIB-Adapter comes directly from this connector.
- Connect the EIB to the EIB-Connector.
- In the package you find a short piece of recloseble fastening tape. With this tape you can glue the EIB-Adapter to the backside of the display.
- Ready!

It is recommendable to program the touch display before the installation.

The usage without custom software isn't meaningful!

First steps

Installation of the software

For the EIB-Adapter you need a special equipment driver for the touch display of the GPx77 and GP2000 family. This driver isn't a component of the orginal Pro-face program. You find this driver on our homepage in the internet or on request by e-mail. After the installation of the ProPb software the driver must be copied manually into the contents directory as follows:

- Copy the file "EIB-Ad.TBL" into the directory "plctbl" and
- The files "*.epn" and "*.hpn" into the directory "protocol"

Please take care that the file LCN-AD.TBL isn't in this directory. If this should be the case, please move this file into another directory!

Address modes of the EIB-Adapter

The EIB-Adapter decodes the EIS telegrams EIS 1 to EIS 14. So that the display can communciate with the EIB, it is necessary to make some definitions in the projecting software of the company Pro-face.

There are different address types which are listed for you in the following:

Address type "D"

The address type "D" is for the communication with the EIB-Adapter. Only some few addresses in this memory area are important at the moment. All addresses not listed here aren't usable from the user.

Address:	Meaning:
D004000	Physical address of the EIB adapter
D004002	Version number of the software
D004003	Error information
D004004	serial number of EIB-Adapter
D004005	Code for functionality. See Chapter „Extended Functionality“
D004006	Divisor for EIS5 Telegramms (from version 11) see Chapter „EIS 5“
D005000-D005119	see chapter “Cyclic requests at the EIB-Adapter”
D006000-D006249	See Chapter „Extended Functionality“-> „EIS 15“
D008000-D008255 D009000-D009255	See Chapter „Extended Functionality“-> “Minimizing of cyclic reading“
D010000-D008249 D011000-D009249	See Chapter „Extended Functionality“-> “Central On/Off“
D255200	see chapter “Compatibility”

The physical address of the EIB-Adapter must be written once at least. The EIB-Adapter keeps this address till it is written once more.

The physical address must be transferred in the binary representation into the adapter.

Example:

Physical address to ETS	Representation as a hexadecimal numerical value:
1.1.22	0 x1116
2.1.1	0 x1201

3.2.1	0 x2301

Address type "LS"

The address type "LS" is built up a little differently than the address type "D". The address type "LS" stands for “local system” area, i.e. almost all values remains in the display.

In the word access:	LSxxxx	with	xxxx	= 0000 ~ 2047 , resp.
In the bit access	Lsxxxxyy	with	xxxx	= 0000 ~ 2047 and
			yy	= bit of position 00 ~ 15

Furthermore the "LS" area disintegrates into 4 sections:

- The "System Data Area" of LS0000 to LS0019: Here you can find information like the current screen number, time, date and conditions within the display. You learn details in the "PLC Connection Manual" on the pages 1-5.
- The "Reading Area" isn't available in connection with the EIB.
- The " User’s Area" of LS0020 to LS2031: This area is fully available. It can even be battery backed, i.e. the data remain unchanged also after power failure.
- The "Special Relays" of LS2032 to 2047: Here you can find information about the time which is needed to update the contents of a screen, error codes, D-script errors. You learn more details in the "PLC Connection Manual".

The remaining address types:

All other address types are for the communication with the EIB:

The address types are:

Adresstyp	EIB telegrams
1EIShhmuuu (7EIShhmuuu)	EIS 1, EIS 7
8EIShhmuuu	EIS 8
2EISUhhmuuu, 2EISDhhmuuu	EIS 2
13EIShhmuuu (14EIShhmuuu)	EIS 13, EIS 14
6EIShhmuuu	EIS 6
5EIShhmuuu	EIS 5
3EISAhhmuuu, (4EISAhhmuuu), 3EISBhhmuuu, (4EISBhhmuuu), 3EISChhmuuu, (4EISChhmuuu)	EIS 3, EIS 4
9EIShhmuuu (11EIShhmuuu, 12EIShhmuuu)	EIS 9, EIS 11, EIS 12
10EIShhmuuu	EIS 10

Remark: The declarations in brackets are accepted as input, but represented in the input without bracket!

You can read the exact assignment of the EIS telegrams in the appendix "EIB telegrams".

To the addressing:

At bit access the input value is interpreted as follows:

TYPhhmuuub with

hh = [0..15] (This is the main group in the area from 0 to 15.)

m = [0..7] (This is the middle group in the area from 0 to 7)

uuu = [0... 255] (This is the subgroup in the area from 0 to 255)

B = [0... 9, A. . . F] (Bit number within the word)

At word access the input value is interpreted as follows:

TYPhhmuuu with

hh	=	[0..15]	(This is the main group in the area from 0 to 15.)
m	=	[0..7]	(This is the middle group in the area from 0 to 7)
uuu	=	[0... 255]	(This is the subgroup in the area from 0 to 255)

Unfortunately, the input of the 2 stepped addressing isn't directly possible with the EIB-Adapter. If necessary switch the ETS over on 3 stepped.

Unfortunately, the middle group in the area from 0 to 7 cannot be checked during projecting.

So it is therefore possible to enter the address 1EIS129001, e.g. This would be main group 12, middle group 9 and subgroup 1. This address isn't interpreted by the EIB-Adapter. Write requests to this address lead to no reaction, read requests become 0 basically.

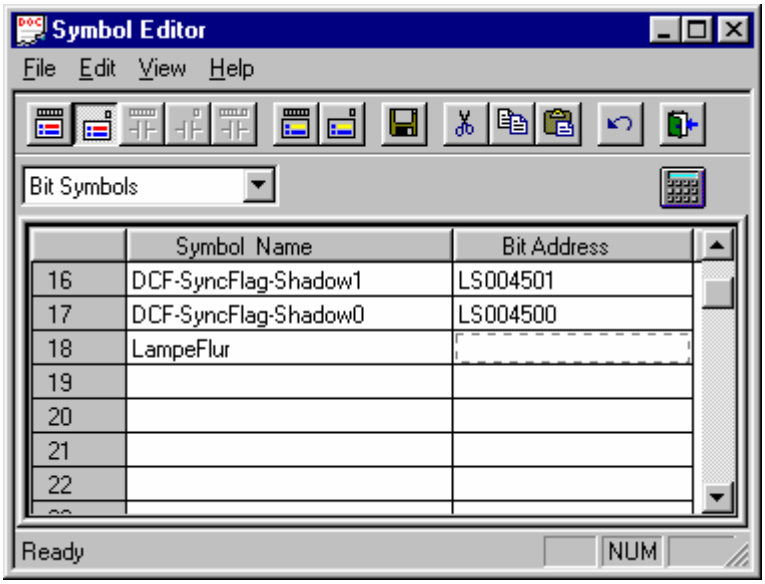
Symbolic names

A very comfortable and simple input of addresses is possible with the help of the "symbol editor". E.g. to this you give the text "LampeFlur" at the placing of a switch for a lamp in the corridor instead of the actual address (e.g. 1EIS0200300). After you have pushed “enter” you get the following error message:

You confirm this (push OK) and you have defined a new symbolic address.

If you need this address somewhere else, you simply press the button (on the right besides the address field) and select your address "LampeFlur". Ready! How is the program informed which address actually hides behind the symbol? Complete all program parts to this apart from the project manager and start the symbol editor.

You then find your entered symbolic name in the symbol editor:



List the corresponding correct address in the column "Bit Address" or "Word Address" now and the addresses are already fixed for all screens. If a change of the address should arise later, you need to change it only here and you are ready. It isn’t necessary to check all screens. Additionally you also can export the addresses to use for later projects or documentation.

It is also possible to use our Konvertertool 2 which reads out the database of your EIB system and transfers the names of all group addresses to symbolic labels. This tool can be downloaded from our homepage.

Logic functions

The topic "D-Script" is explained detailed in the orginal manuals (Operation Manual; Tay Reference Manual). We would like to give you some tips for the use of these "D-Scripts" as a logic function on or in the touch display in the following.

Some important preliminary notes:

1.) The touch display isn't a SPS, i.e. it is possible to do some functions with the D-Script by use a script. The number of functions should, however, remain “easily comprehensible”. The cycle time of the display is reduced strongly at the intensive use of scripts till it can partly come to a crash of the display.

2.) Amoung others the trigger are available for falling and changing flank in a D-Script. These two triggers have to be used with caution. Both triggers are triggered when changing the screen pages!

Now here some examples:

1st example: Logical combination of 2 group addresses to one group address

You then program as follows:

- Put the "timer duration" value for 10 seconds. This value suffices in almost all cases.
- Write the following line to the "Formula" field:

```
1. if (([b:Eingang1]== 1) and ([b:Eingang2]== 1))
2. {
3.     if ([b:Ausgang] == 0)
4.     {
5.         set([b:Ausgang])
6.     }
7.     endif
8. }
9. else
10.{
11.    if ([b:Ausgang] == 1)
12.    {
13.        clear([b:Ausgang])
14.    }
15.    endif
16.}
17.endif
```

- In line 1 it is checked whether the group addresses "Eingang1" and "Eingang2" are switched on. If this is the case, the output is set (line 5) if the output is 0. If both entrances shouldn't be switched on, then the output is deleted (line 13) and the exit is switched if the output is 1. You could switch the output signal also directly here, the exit, however, is sent newly each time if the D-Script is started, in our example every 10 seconds. However, this isn't acceptable in most cases.

2nd example: Group address causes alarm and switches over the screen

- For this task you put the trigger on “rising edge” and writes the address in the appropriate field.
- As a script you write the following:

```
1. [w:LS0008] = 100
2. set[b:LS001401]
```

- The script executes actions following now:
- The screen is called with the number 100 in the 1st line. (The storage position LS0008 contains the current screen number. If it is described then the side is changed!)
- The buzzer is switched on by the bit 1 in the storage position LS0014 of the display in the 2nd line.
- The script is ready. But caution! The buzzer remains on. You should certainly define a switch on the page 100 which deletes the bit 1 in the storage position LS0014 again.
- **Caution!** Should the bus user send only "on" telegrams then the alarm is aroused exactly once and then never again! E.g. this passes at a movement sensor which is programmed so that it sends "on" telegrams cyclical at movements. Since the script reacts to the rising flank, however, (from "0" to "1"), the script reacts only once. There are two solution ways now:
- You program the movement detector (or your actor) so that it also send an "off" telegram at “completion of the movement", or
- You complete the script with the line “clear ([b:Alarmsensor])”. After every "on" telegram an "end" telegram is immediately sent but you recognize every alarm.

Alarms

With the display it is possible to create several types of alarms, alarms which are shown through a continuous text on the lower screen independent from the current screen page up to alarm history which shows "come, gone, confirmed" respectively with time. Since the latter function is well seldom used, we would like to confine ourselves to the "alarm summary" and the simple "alarm list." We already have shown you a more complex possibility in " 2nd example: Group address causes alarm and switches over the screen ".

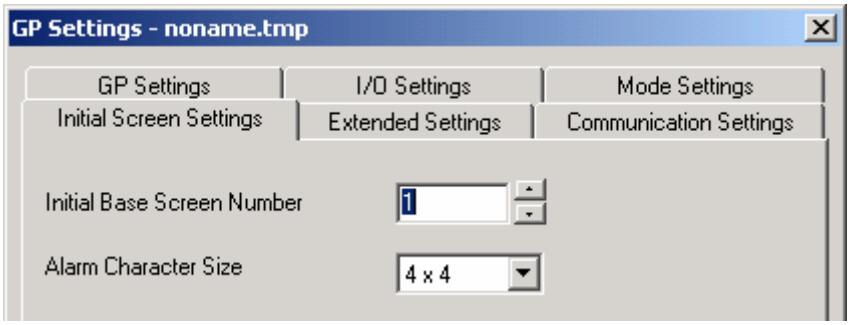
3rd example: "Alarm summary"

The alarm summary is the alarm programmed most simply. Start the alarm editor in the "Project manager":

In the alarm editor please select the "base alarm" from the menu "alarm" or press the button "base alarm".

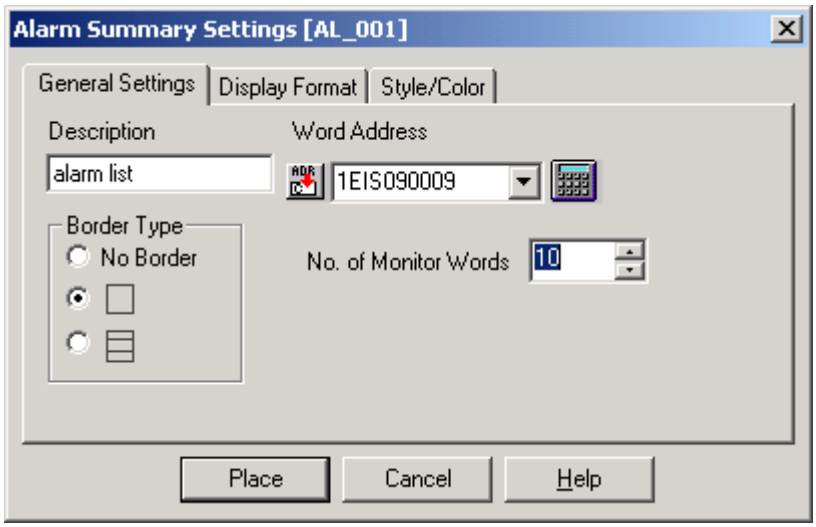
In the table list the group address in the field "bit address" now or select the corresponding symbolic name from the list which reports the alarm. You switch the field "type" over to "report". Still list an arbitrary text in the last column and format it according to your color wishes. Only transfer now and ,as soon as the alarm is triggered, you can see it at the lower screen as a current text. With several upcoming alarms the alarm texts are put behind each other. The texts run on the lower screen margin as long as they aren't avaible any more.

The type height of the alarm can be adjusted in the GP set-up in the file card "Initial Screen Settings":



4th example: "Simple list alarm"

At the beginning you do the same things like in the “3rd example: Alarm summary” for the construction of a "simple list alarm", but write the value "overview" in the column "type". In addition, the group addresses of the alarms must succeed one by another, e.g. group addresses 9/0,9/1,... 9/10.



Store the alarms and open in the editor the screen on which you want to show the alarms. There you select the button "alarm overview"  from the “parts".

Desciption:

You write a description in the "alarm overview" dialog again.

Word Address:

The address or the symbolic name of the first alarming group address is typed in the field "word address".

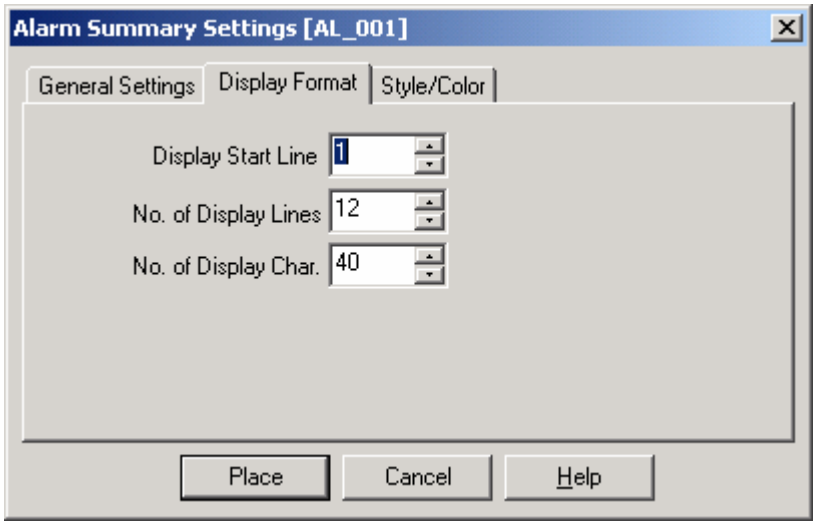
No. of Monitor Words:

In the field "No. of Monitor Words" you have to type the number of successive alarms.

Border Type:

You still can adjust the appearance of the table in the field "Border Type".

Then choose the file card "Display Format".



Display Start Line:

You list the line number in the field "Display Start Line" on which you have written down the first error message. Only the alarms are shown which are listed in the list from this line number. Are here addresses, which are already typed as “Message”, so they are ignored!

No. of Display Lines/ Display Char.:

With the entries “No. of Display Lines" and “No. of Display Char." you can adjust how much lines or characters you would like to represent.

Special features at the EIB

In this chapter the special features are explained in connection with the projecting of the display with the EIB:

Read flag

One of the probably most important things which you must pay attention to by projecting of the display at the EIB is the read flag. This, however, isn't necessary only at the visualization with the touch display but also at every other visualization.

Over the read flag you adjust for every group address whether it can be read by the bus or not.

Imagine now you have put the read flag for a group address twice and ask for this group address now. You get two answers! And which is the right one? Both can have other states, switched over other group addresses. Therefore you only put exactly one read flag per group address! And of course this should be put in the actor. You aren't interested in the state of the switch with which you switch this lamp but in the lamp itself, and this one could have another state through the superior group address than the switch.

Transfer flag

What was already described for the read flag applies also to the transferred flag. There are quite a number of components, e.g., for stairwell light on the market which change their state automatically. These components usually have the quality, that they send state change only when you have put the transferred flag! Check it! Unfortunately, there are also components which even don’t do this if the transferred flag is set. Every visualization always show you the state of the last action externally and not, that the light is actually out again.

Cyclic request at the EIB-Adapter

The EIB-Adapter has the function of “cyclic read”. For this you have to write down directly the group address to read in the address areas D005000 to D005119.

Directly means:

group address:	entry in Hex
0/0/1:	0x0001
1/1/1:	0x0901

like they are transferred in the EIB.

calculation (2-stepped):	subgroup + main group * 2048
calculation (3-stepped):	subgroup + middle group * 256 + main group * 2048

The entry can be done with a simple D-Script which is called once when push the button. The button can (also should) delete after initiation.

The addresses written down there are called cyclic (one per second).

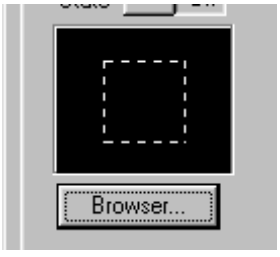
Central off/central on

The question about the function central off/central on is asked again and again. The problem about the central function simply is that at the EIB an actor can listen to more than one group address and will accept this switching state. We imagine the following scenario now. We have three lamps with the group addresses 3/1, 3/2 and 3/3. Additionally all 3 lamps can be turned off or on about the group address 4/0. You switch the lamp 1 and lamp 2 on and lamp 3 off. Now the question: Is this “central on” or “central off” now? The answer could be: “I don’t know!” But the electronics doesn't know this state! What has to be done? You should switch central functions always without monitoring therefore only as a button, namely a button for "on" and one for "off".

But a new problem appear now! On the display you have projected all five switches (three switches for "on/off" for the lamps and two switches for the central function). You switch all lamps on with the central switch now. Foresee there all lamps in the plant go on but the switches on your display stay in the state in which they were before! What has happened?

You have sent an "on" telegram about the group address 4/0, all 3 lamps have heard it and reacted to it! But none of the lamps has returned an “on” telegram on their real address. Therefore the visualization knows nothing about this state change. There are two possibilities now of clearing this:

1. You are lucky and all your actors send your condition change after setting the transfer flags or
2. You change the switch for the central function so that the first switch switches the first group address (3/1) and then put two invisible switches directly on the first switch with the two other group addresses (3/2 and 3/3). The display sends now three telegrams on the bus by pressing the button. The display is updated correctly now or



3. you use cyclic read (see chapter „Cyclic requests at the EIB-Adapter).
4. you use the extended functionality

Update of the screen data

At the visualization on the EIB you get a small problem. If you start a visualization, it doesn't know the states of the several bus users. The visualization must therefore ask all interesting participants once. This can sometimes last some time. Because of

that the display gets the information that all group addresses have the condition "0" or "off" when a new side with group addresses which are unknown to the EIB-Adapter is build up. The state changes, however, automatically if the EIB-Adapter has received the correct information from the participant. Should this not be the case after some minutes either, you should connect the ETS to the EIB and check what kind of actions are on the bus at the time when you call this site. Pay attention particularly to a high bus load, (i.e. many telegrams per second) and to busy reports here. You should as well check whether the EIB-Adapter also actually gets an answer to its question. Perhaps you should use our EIB-Logger for this. More information about the EIB-Logger can be find on our homepage.

Detection of incoming telegrams

With the EIB-Adapter it is possible now to check for telegrams EIS 1, EIS 2 and EIS 8 if a new telegram is received from the EIB-Adapter. For this bit 8 is set for 5 display requests when receiving.

Example:

A movement sensor only sends “on” telegrams to the address 1/0/6 (EIS1). So the address 1EIS0100060 will always be set to 1. Now, with the address 1EIS0100068 you can check with a D-Script if the telegram comes or not.

EIS 5 telegram

This function is available starting with firmware version 11!

EIS 5 telegram will always transfered with 2 fractional digits. It is partly very effortful (or also only one) "to cut" these 2 fractional digits off in the projecting software Pro/PB. However, it usually is nonsensically, e.g.: showing these to the room temperature at the actual indication with 2 fractional digits. Therefore can one to adjust the number of fractional digits to be come off about the storage position D004006. (Of course it is produced again when sending on the bus.)

The storage position D004006 can arbitrarily be often changed, this however only with a little effort on a display it for example possibly therefore reconciles values with 2 to represent others with a 1 fractional digit.

If you would like to have one fractional digit on a display then you write here a D-Script, which one when calling this side (Trigger > t, [W : LS0000] == current display) writes the value 1 into the storage position D004006 ([W : D004006] = 1).

You also must put this value on the desired value in the same way on all other sides.

To show a side several values with a different number from fractional digits, you must copy the values to a LS area and then show this value by means of a D script.

e.g.:

```
• [W:D004006] = 1                    // 1 fractional digit
• [W:LS0100] = [W:5EIS010123]
• [W:D004006] = 2                    // no fractional digit
• [W:LS0102] = [W:5EIS010125]
• [W:D004006] = 0                    // 2 fractional digits
• [W:LS0104] = [W:5EIS010127]
```

Extended functionality

The EIB adapter is delivered in two different functionalities starting with firmware version 10.

There isn't an essential expansion of the functionality in the standard version.

Expansion of the standard functionality:

Selections of the serial number on the address D004004

Selections of the performance range on the address D004005

The following functions have there been in the extended performance range (D004005 = 1):

A. Support of EIS-15 telegram

B. minimization of cyclical reading, by been aimed read.

C. Reduction of the bus burden at the function " central On/Off"

The extended functionality can afterwards be switched freely any time. To this a reversibly unambiguous code is written for the serial number (D004004) in the address.

Caution!

Describing the storage position D004004 is possible for at most 20times. So you should be sure, which value you write into this register.

Caution!

The data stored, after a tension loss of the EIB adapter cannot durably all the data are typed in once more become data of the extended functionality. To this it offers itself to use to the D scripts of the display.

Suggestion following to this:

1. Make an empty screen, add your logo, address etc..
2. You prepare a D-Script for current screen number with the Trigger f->t with the Trigger [W:LS0000] == actual screen number
3. Type the D-Script in in accordance with the following description.
4. Complete the D script with the entry:
   [W:LS0008] = the next screen number
5. Enter these screen number as the start screen number in the GP-Setup.
6. The display starts with your logo now and immediately switches over after the processing of the script on the real start screen. However, they can have this taken care also manually. Put an invisible switch on the whole screen, wich switche to the normal start screen.

While preparation of this D-Script it could happen the amount of tags per screen is exceeded. The tag is a kind of variable in the display. The amount of tags per displays varies between 128

and 384 pieces. How many tags you are using, you can find in the menu "Screen" > "Screen information". If you should have used more then allowed, then you must call several Screens according to the method mentioned above after each other and split the D-Sscript in section respectively.

Another possibility to write all data to a cf-card and load it by a D-Script into the EIB-Adapter. This is accurately described in the manuals of of pro-face in detail.

EIS15 Telegrams

It is necessary in many systems to send EIS15 telegrams over the bus and then to be able to show them on a display, too.

To this the address range D006000-D006249 is reserved in the EIB adapter.

It is possible within this to use 25 EIS15 telegrams. The addresses have the following meaning:

Address	Function
D006000	EIB group address in a binary representation (like in the bus telegram)
D006001- D006007	Memory area for the telegram
D006009	Trigger word for the EIB telegram If a one is written in this address, then the data in the address range D006001-D006007 will be send to the address defined in D006000 immediately. The value is immediately written back to 0 This storage position is when receiving an ice 15, put on 2 it must by telegram describes this address with 0 deleted again be.
D006010	EIB group address for 2nd EIB telegram
D006011- D006017	Data for 2nd EIB telegram
D006019	Trigger for 2nd EIB telegram
…	
D006249	Trigger for 25th EIB telegram

How to use this functionality:

For displaying the EIS15-telegram use a “S-Tag”. You set up the trigger bit address on e.g. D0060091, the word address on e.g. D006001. The number of chars should bet set to 14. Place the tag. You still must in addition write down the group address on the table over the D-Script mentioned above of course.

For sending an EIS15 telegram use the part “keyboard entry”. Also you must enter here: the word address (for example D006001) and the length in chars (14). Important: You still have to place a word switch, which sends out the telegram after typing (write a 1 tor D006009), too.

Minimization of cyclical reads (Read after Write)

Modern dimmers often don't offer the possibility of configuring them so that they send their new value at condition changes on the bus automatically. The only solution of this problem till now, is to send a read command on these dimmers cyclically. The question but then was, is how frequent actually. When the cycle time is adjusted too little then one reaches a high bus burden, us the cycle time adjusted too greatly, the operator waits to long. Now here is the solution for EIB-Adpater:

The EIB adapter can in the memory area D008000 -- D008255 and D009000-D009200 are configured so, that if a telegram "Value-write" or "Value-confirm" were recognized by the EIB adapter, the EIB adapter sends out a "Value-Read" telegram automatically. To this the address of the "Value-Read" telegram is always the address to be imposed listed in the straight address and in the address following on this.

The memory area is after a receiption of a corresponding telegram searched (starting at D008000) until a 0 is in the adress.

How to setup this functionality:

Unfortunately, since there isn't any possibility at present of importing these data from the database of the ETS, you have to set it up "by hand". The data must be copied by means of D-Script to the memory area of the display.

This the group addresses must be entered directly again:

Calculation (2 stepped): Subgroup+main group of * 2048

Calculation (3 stepped): Subgroup+middle group of * 256+main group of * 2048

Example:

A dimmer with the following values:

On/Off on 0/1/200, Dimm up/down on 0/1/201, Value on 0/1/202.

The following D-Script is the result:

```
1. [W:D008000] = 0*2048+1*256+200          // 0/1/200
2. [W:D008001] = 0*2048+1*256+202          // 0/1/202
3. [W:D008002] = 0*2048+1*256+201          // 0/1/201
4. [W:D008003] = 0*2048+1*256+202          // 0/1/202
5. [W:D008004] = 0                         // end
```

Central On/Off

Central On/Off or other functions taken similarly are a problem for the most visualizations. If one switches a number of lights on or off via the central function, then the conditions of the individual lights aren't represented correctly in the visualization.

The EIB adapter solves this problem as follows:

In the EIB adapter a table is deposited in which the central group address is at first position and all "sub-group addresses" on the positions 2 to 10.

If a telegram is received on the central group address now, then merely the contents of the internal memory (that is the now current condition of the central group) become copied on the internal memory places of the "sub-group addresses".

The memory area provided for it is the area of "D010000" to "D010249" and "D011000" to "D011249".

The memory is so organised, that the final char “1” to “9” is the "sub-group addresse" and the central group address always lies on the final char 0.

The memory will be searched until a 0 zero for a central group address will be found. The “Copying” of the "sub-group addresses" is continued also respectively up to finding an address with the value 0.

With our converter tool, which you can download from our home page free of charge, you can extract this d data directly from the database. Then simply copy these data via the clipboard into the corresponding D-Script.

Some tips

We would like to refer the following rules to you:

- Befor you plan and realize, take the time and “play” first with the software and the touch display. Try out single switches and check them for their behavior on the bus. Later, the time which you are using up now pays off again.
- Handle thriftily with graphic elements. The touch display is very compact and seems overloaded and confused fast.
- Build up your illustrations hierarchically - from the overview to detail.
- Avoid full graphic. All standard symbols can be varied generously.
- Use symbols instead of fix addresses on the screen. You can then assign the correct addresses to the symbolic names in the "symbol editor". Later, you don't have to correct an address on all pages, if a group address moves (see "Symbolic names ").
- Check the read flags via ETS. Per visualized object exactly one read flag must be set. These read flags should be set to a meaningful way in an actor and not in a switch. Additionally it is advisable to use the transfer flags in all objects which can send their state change from itself (stairwell light).

Appendix A "EIB telegrams"

The following EIB telegrams are supported by the EIB Translator:

The exact construction is explained in the chapter "address modes".

Overview:

EIB telegrams	Address type
EIS 1, EIS 7	1EIShhmuuu (7EIShhmuuu)
EIS 2	2EISUhhmuuu, 2EISDhhmuuu
EIS 3, EIS 4	3EISAhhmuuu, (4EISAhhmuuu), 3EISBhhmuuu, (4EISBhhmuuu), 3EISChhmuuu, (4EISChhmuuu)
EIS 5	5EIShhmuuu
EIS 6	6EIShhmuuu
EIS 8	8EIShhmuuu
EIS 9, EIS 11, EIS 12	9EIShhmuuu (11EIShhmuuu, 12EIShhmuuu)
EIS 10	10EIShhmuuu
EIS 13, EIS 14	13EIShhmuuu (14EIShhmuuu)

Remark: The details in brackets are accepted at the input, represented in the input without brackets, however!

EIS 1, switching:

Data length: 1 bit

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not usable		Always 0					0: off 1: on

Remark: "Not usable" meant: reserved for "Value write" / "Value response" / "Value Read" coding.

Used address modes: 1EIShhmuuu

Examples:

Group address:		
	Bit object	Word object
1/0/7	1EIS0100070	1EIS010007
2/1/255	1EIS0212550	1EIS021255
13/2/12	1EIS1320120	1EIS122012

There is the possibility at the objects EIS 1, EIS 7, EIS 8 and EIS 13 of recognizing incoming telegrams within the program. To this the 8th bit (1EIShhmuuu8) is put for exactly 5 queries, therefore for short time.

EIS 2, dimming

Data length: 4 bits

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not usable		Always 0		Direction	Steps/ increments		
					Code:	Increments	
				Upward: 1 Downward: 0	000	Stop	
					001	1	
					010	2	
					011	4	
					100	8	
					101	16	
					110	32	
					111	64	

Only the “increments at 1” are implemented at EIS 2. You use the address modes 2EISUhhmuuu (U for up) for it for “direction upward” and 2EISDhhmuuu (D for down) for “direction downward”. A “bit set command” or a value assignment with value < > 0 starts the change of the dimmer, a “bit release command” or a value assignment with the value 0 stops the change.

Examples:

Group address:		
	Bit object	Word object
15/3/7 (bright diming)	2EISU1530070	2EISU153007
2/4/255 (bright diming)	2EISU0242550	2EISU024255
13/2/12 (bright diming)	2EISU1320120	2EISU132012
1/0/7 (dark diming)	2EISD0100070	2EISD010007
2/6/255 (dark diming)	2EISD0262550	2EISD026255
13/7/12 (dark diming)	2EISD1370120	2EISD137012

EIS 3, time

Data length: 3 bytes

First data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Weekday			Hour (binary value)				
000	No day		00000			0	
001	Monday		00001			1	
010	Tuesday		00010			2	
011	Wednesday		00011			3	
100	Thursday		...				
101	Friday						
110	Saturday		10111			23	
111	Sunday		other values are invalid				

Second data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	Minute (binary value)					
		000000	0				
		000001	1				
		000010	2				
		000011	3				
		...					
		111011	59				
		other values are invalid					

Third data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	Second (binary value)					
		000000	0				
		000001	1				
		000010	2				
		000011	3				
		...					
		111011	59				
		other values are invalid					

EIS 3 telegrams will transfer if the value for second is written.

Please use the following objects:

3EISChhmuuu for the second value

3EISBhhmuuu for the minute value

3EISAhhmuuu for the hour/day value

Splitting day and hour a D-Script has to be used if necessary.

Examples:

Group address:		
	Bit object	Word object
1/0/7 (minute)	---	3EISB010007
2/4/255 (minute)	---	3EISB024255
13/3/12 (minute)	---	3EISB133012
1/4/7 (hour)	---	3EISA014007
2/6/255 (hour)	---	3EISA026255
13/6/12 (hour)	---	3EISA136012

EIS 4, date

Data length: 3 bytes

First data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	Day in the month (binary value)				
			00000			0	
			00001			1	
			00010			2	
			00011			3	
			...				
			11111			31	

Second data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	0	0	0	Month (binary value)			
				0000		not used	
				0001		1	
				0010		2	
				0011		3	
				...			
				1100		12	
				other values are invalid			

Third data byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Year (binary value) offset to the year 1900							
			00000000	1900			
			00000001	1901			
			00000010	1902			
			00000011	1903			
			...				
			11111111	2155			

EISs 4 telegrams will transfer if the value for the day is written.

Please use the following objects:

3EISChhmuuu for the day value

3EISBhhmuuu for the month value

3EISAhhmuuu for the year value

Examples:

Group address:		
	Bit object	Word object
1/6/7 (day)	---	3EISC016007
2/6/255 (day)	---	3EISC026255
13/5/12 (day)	---	3EISC135012
1/5/7 (month)	---	3EISB015007
2/7/255 (month)	---	3EISB027255
13/5/12 (month)	---	3EISB135012
1/3/7 (year)	---	3EISA013007
2/3/255 (year)	---	3EISA023255
13/3/12 (year)	---	3EISA133012

EIS 5, value

Data length: 2 bytes

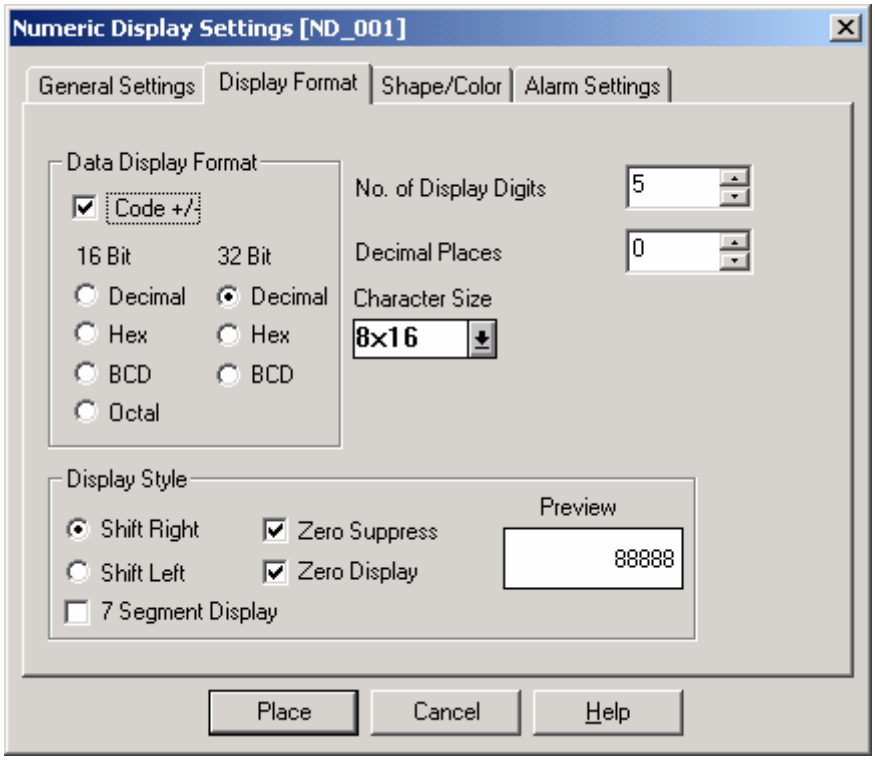
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
sign	exponent				Mantisse										
0: + 1: -					• The predefined dissolving is 0.01 • Negative values are represented in the two bit binary complement										

Formula:

Value = sign * 0.01 * Mantisse * 2^{exponent}

Please pay attention that you have to adjust 32 bit values codec +/- and 2 decimal places when you project the values on the display.

Attention: Since this address is 32 bits "wide", it needs the address indicated in the EIB adapter and the address following on this at the transfer. The use of a second EIS 5 object on the same side (or script) with the address following on this leads to a faulty representation of both values. Distribute these on two different sides or change the object address!



Please use the following objects for EIS 5 telegrams: 5EIShhmuuu.

Examples:

Group address:		
	Bit object	Word object
1/4/7	---	5EIS014007
2/5/255	---	5EIS025255
1/3/12	---	5EIS013012

EIS 6, scaling

Data length: 1 byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Binary value							
		00000000		off			
		00000001		minimum (0.4%)			
		...					
		11111111		maximum (100%)			

Please use the following objects for EIS 6 telegrams: 6EIShhmuuu.

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	6EIS010007
2/5/255	---	6EIS025255
13/6/12	---	6EIS136012

EIS 7, drive control

EIS 7 is used mainly to control shutters.

Data length: 1 bit

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not usable		Always 0					0: close/down 1: up/open

resp.

Data length: 1 bit

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not usable		Always 0					0: stop/step up 1: stop/step down

Use the EIS 1 telegram for it.

EIS 8, priority

Data length: 1 bit

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Not usable		Always 0				Priority	Value

The following truth table clarifies the behavior between "priority" and "value".

Priority		Actual state	exit
Priority	Wert		
0	Don´t care	0 1	0 1
1	0 1	Don´t care	0 1

For EIS 8 please use the object: 8EIShhmuuu.

To switch the individual state you can adjust either the priority directly (0 ... 3) with a value assignment or achieve every arbitrary state with two bit switches, e.g. 8EIS0401110 switches to address 4/0/111 the switching state and 8EIS0401111 switches the priority on even this address.

Examples:

Group address:		
	Bit object	Word object
1/0/7 (switch)	8EIS0100070	8EIS010007 = 0|1
2/1/255 (switch)	8EIS0212550	8EIS021255 = 0|1
13/2/12 (switch)	8EIS1320120	8EIS122012 = 0|1
1/0/7 (priority)	8EIS0100071	8EIS010007 = 0|2
2/1/255 (priority)	8EIS0212551	8EIS021255 = 0|2
13/2/12 (priority)	8EIS1320121	8EIS122012 = 0|2

There is the possibility at the objects EIS 1, EIS 7, EIS 8 and EIS 13 of recognizing incoming telegrams within the program. To this the 8th bit (8EIShhmuuu8) is put for exactly 5 queries therefore for short time.

EIS 9, float

Data length of 4 bytes

Data bytes correspond to the IEEE754 coding.

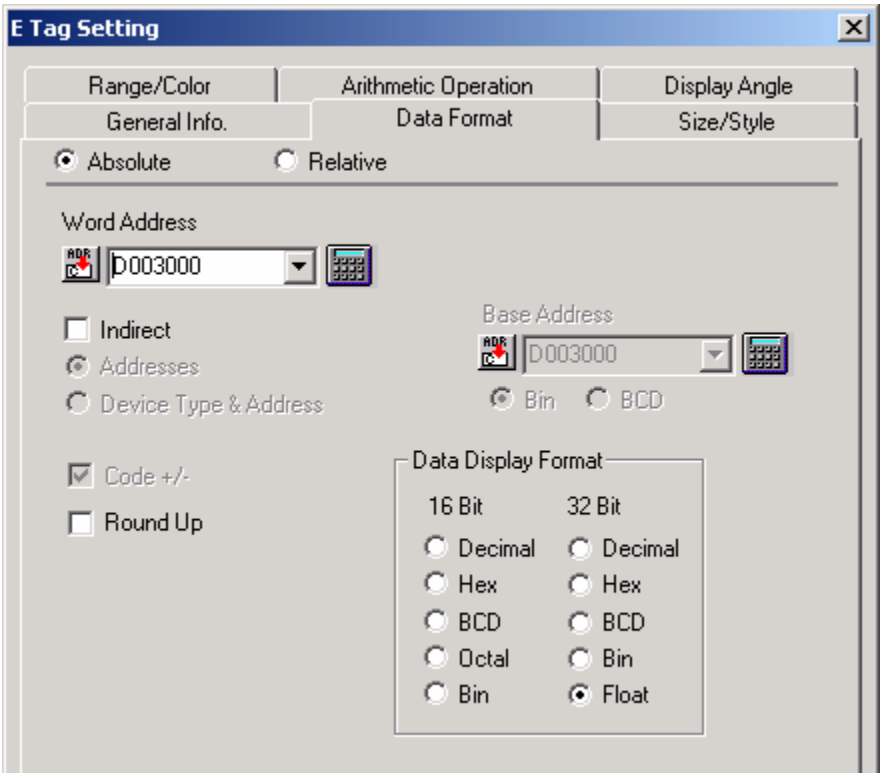
Bit 31	Bit 30... 23	Bit 22... 0
sign of the mantisse 0: + 1: -	exponent (8 bits)	mantisse (23 bits) encoded binarily

This format is described as an "IEEE 754 single format" and is exact on 6 decimal places.

Formula:

$$\text{Value} = \text{sign} * \text{base} * 2^{\text{exponent}}$$

The EIS 9 telegram can only be shown for lack of functions of the display software. The display is only possible with the "E-Tag". Adjust this as follows:



Attitudes in the other menus can be carried out arbitrarily.

For EIS 9 please use the object of 9EIShhmuuu.

Attention: Since this address is 32 bits "wide", it needs the address indicated in the EIB adapter and the address following on this at transfer. The use of a second EIS 9 object on the same side (or script) with the address following on this leads to a faulty representation of both values. Distribute these on two different sides or change the object address!

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	9EIS010007
2/5/255	---	9EIS025255
13/6/12	---	9EIS136012

EIS 10, 16 bit counter

Data length: 2 bytes

Bit 15... 0
• Positive values are encoded binarily • Negative values are in the two bit binary complement. • Whether negative numbers are used depends on the application.

For EIS 10 please use the object: 10EIShhmuuu.

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	10EIS010007
2/5/255	---	10EIS025255
13/6/12	---	10EIS136012

EIS 11, 32 bit counter

Data length: 4 bytes

Bit 31... 0
• Positive values are encoded binarily • Negative values are in the two bit binary complement. • Whether negative numbers are used depends on the application.

Please pay attention that you adjust 32 bits when you project the values on the display.

For EIS 11 telegrams please use the following objects: 9EIShhmuuu.

Attention: Since this address is 32 bits "wide", it needs the address indicated in the EIB-Adapter and the address following on this at the transfer. The use of a second EIS 11 object on the same side (or script) with the address following on this leads to a faulty representation of both values. Distribute these on two different sides or change the object address!

Examples:

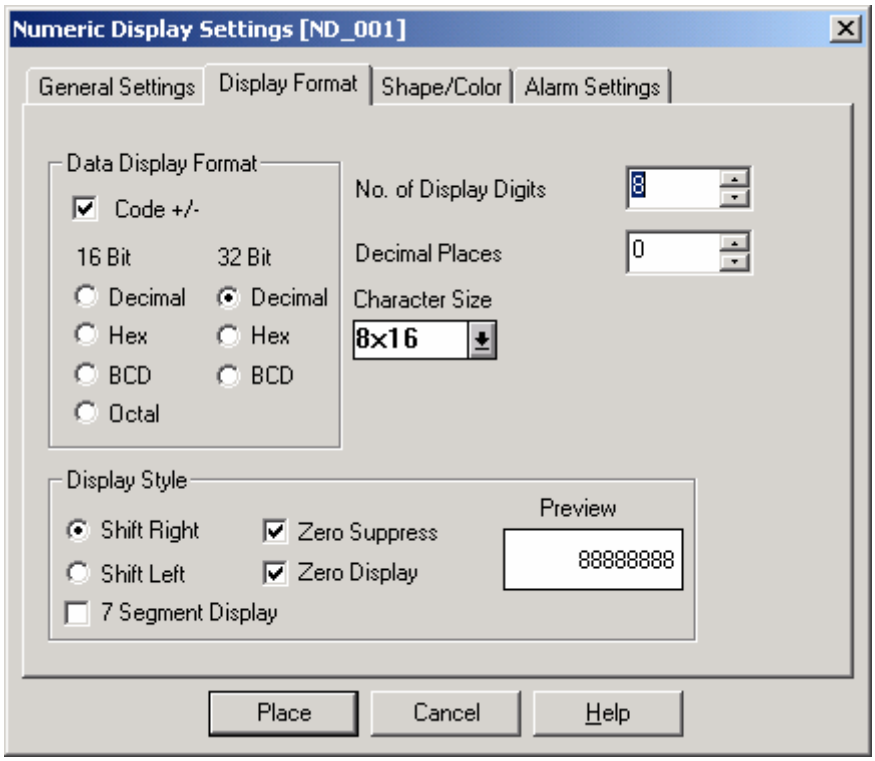
Group address:		
	Bit object	Word object
1/0/7	---	9EIS010007
2/5/255	---	9EIS025255
13/6/12	---	9EIS136012

EIS 12, Access

Data length: 4 bytes

Bit 31... 8	Bit 7... 4	Bit 3...0
Entry code	Flags: Bit 4: Encode Bit 5: Direction Bit 6: Permission Bit 7: error	Index

Please take into account that you adjust bits at the Projektierung of the values on the display 32.



For EIS 12 telegrams please use the following objects: 9EIShhmuuu.

Attention: Since this address is 32 bits "wide", it needs the address indicated in the EIB-Adapter and the address following on this at the transfer. The use of a second EIS12 object on the same side (or script) with the address following on this leads to a faulty representation of both values. Distribute these on two different sides or change the object address!

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	9EIS010007
2/5/255	---	9EIS025255
13/6/12	---	9EIS136012

EIS 13, ASCII character

Data length: 1 byte

Bit 7... 0
ASCII sign

The EIB-Adapter writes down the value on the low byte of the corresponding address. The “S-Tag” used to show puts, however, the high byte first and then the low byte in the standard attitude. Because the high byte is always zero, however, the “S-Tag” will never represent a value. For the solution of this problem you can on the one hand surround the attitude for the string order or on the other hand copy the value from an EIS address into a LS address and simultaneous multiply with 256 by means of a "D-Scriptes". When sending you must divide correspondingly by 256 and copy from LS into EIS.

For an EIS 13 telegram use the object of 13EIShhmuuu.

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	13EIS010007
2/5/255	---	13EIS025255
13/6/12	---	13EIS136012

There is the possibility at the objects EIS 1, EIS 7, EIS 8 and EIS 13 of recognizing incoming telegrams within the program. To this the 8th bit (13EIShhmuuu8) is set for exactly 5 queries for short time.

EIS 14, 8 bit counter

Data length: 1 byte

Bit 7... 0
• Positive values are encoded binarily • Negative values are in the two bit binary complement. • Whether negative numbers are used depends on the application.

For an EIS 14 telegram use the object: 13EIShhmuuu

Examples:

Group address:		
	Bit object	Word object
1/0/7	---	13EIS010007
2/5/255	---	13EIS025255
13/6/12	---	13EIS136012

EIS 15, EIB character string

Data length: 14 bytes (MAXDATA)

Byte 1..14
• Every byte represents a sign. • If the string to be transferred is shorter than 14 signs, it has to be completed with the sign zero (CHR (0)). (Zero Terminated) • Unused bytes have to be covered with zero.

See “Extended Functionality” -> “EIS 15”

Technical specifications

Environmental conditions

Temperature:

operation	5 to 35° Celsius (41 to 95° F)
Storage	-25 to 60° C (-13 to 140° F)

Atmospheric humidity:

operation	10 to 80% without condensate formation
Storage	10 to 85% without condensate formation

Operating conditions:

Normal office or private use conditions. An excessive dust collection has to be avoided. Never exposing the device to direct solar radiation or a strong light source.

The EIB adapter corresponds to the EMV directives 89/336/EEC and is to the EN 55022 class B and to the EN 55024 certified.

Otte Elektronik GmbH
Hannoversche Str. 66
38116 Braunschweig
Tel. : (49) 0531- 25 12 7-10
Fax.: (49) 0531- 25 12 7-18
E-mail: info@otte-elektronik.de
http://www.otte-elektronik.de