EIBA Handbook Series

Release 3.0

Volume 3: System Specifications

Part 7: Interworking

Chapter 1: EIB Interworking Standards

12.03.1999

Table of Contents

1. Introduction

In an EIB-System, different devices from the same or from different manufacturers shall work together. Therefore, there shall be a way that devices can talk to each other. This is accomplished first of all by implementing the EIB-protocol defined in this Handbook. But in addition, there is a need that devices can understand the messages received from other devices. This process is called interworking.

Interworking and Conformance

Products shall conform to these interworking requirements in order to pass the EIBA certification (see Volume 5 “Certification”), so that they may be marked with the EIB logo.

2. Interworking Standards for Group-Communication-Objects (EIS)

This clause gives definitions and rules for interworking of functions which are given by applications.

The goal is to define the values and the interpretation of data included in the communication objects for functions. These standardized functions are called EIB-functions.

General remark: the LSB of all octets in this chapter is situated on the right hand side.

2.1 The Interworking Concept

The definitions of the data for the EIB-functions are necessary to have proper communication and understanding between bus devices. In addition, it will become possible, that an interoperation between devices of different manufacturers takes place.

The definition of the values is called EIB-function typing, see clause 2.6.

If the above mentioned conditions are fulfilled, the process of such a communication between bus devices is called interworking.

The interworking requirements are defined by the EIB Interworking Standards (EIS).

2.2 Extension of the EIB Interworking Standards

The list of EIB Interworking Standards may be extended by submitting proposals to EIBA.(Note that an EIB group telegram may contain up to 14 data bytes.)

In the same way, the normal EIB Interworking Standards can accommodate for established international standards. Indeed, as a future-proof standard for electrical installation, the EIB-infrastructure can be used for many applications, such as building automation etc.

2.3 General Requirements for Interworking

2.3.1 Busload

2.3.1.1 Repetition Rate

This repetition rate shall be selected very carefully as it influences the busload (risk of overload).

Note: At the planning stage of an installation the possible busload shall be assessed.

The following concept is strongly recommended:

1. For manual operations, the time between telegrams that are generated by inputs of an application shall be at least 200 ms.
2. For automatic operations, the repetition rate of these inputs should be in the range of seconds or minutes.

Note: In most control cases, minutes are sufficient, e.g. for measuring wind, temperature, brightness, time etc...

The following aspects shall be covered:

- **Delta Value**

  If an application generates information on the bus depending on a delta value, a limitation-mechanism shall be provided. The manufacturer shall describe this mechanism.

- **Message on Request**

  If this mode is selected, the same requirement as for delta value applies for the device sending the request.

2.3.1.2 Transmission Priorities

The priorities shall be selected very carefully to keep the traffic on the bus as low as possible.

The priorities for EIB telegrams are defined for the various media in Part 3/2 "Medium Dependent Layers":

00_b	system priority
10_b	alarm priority
01b	high priority
11_b	low priority

As the default value, the low priority (11_b) shall be selected. The default priority setting is checked during EIS conformity testing.

2.3.1.3 Interpretability of Telegrams

Each group-telegram shall contain all information so that the receiving device is able to interpret the telegram independently of other group-telegrams.

The manufacturer shall describe telegrams not covered by interworking standards.

2.3.1.4 Requirements for EIB Bus Devices powered by Ancillary Power Supply (APS)

If an application module (AM) of an EIB Bus device is powered by an APS, it shall not disturb the bus with e.g. BUSY acknowledgments in case of receiving group messages if this application module is not powered.

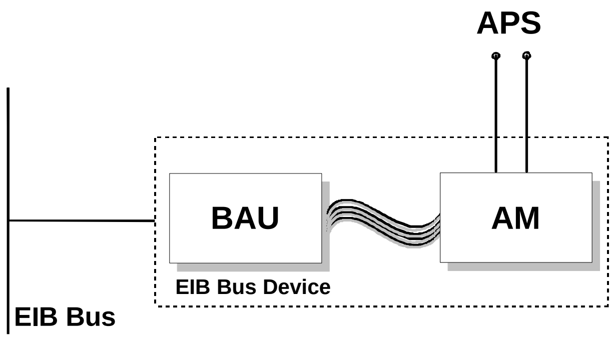


Fig. 3/7/1-1: AM with APS

2.3.2 Application of EIS

2.3.2.1 General

If a function is used where an EIS exists, this EIS shall be used.

It is not permitted to combine functions which have already been EIS standardized with non-EIS standardized functions in a same communication object.

Non-used bits in the telegram data-field shall be set to 0.

2.3.2.2 EIS Declaration

An EIB-application may consist of several interworking standards. The manufacturer shall declare through an appropriate PIXIT (see Volume 5 "Certification") and also in the product's data sheet (EIBA Form 920901, item 4) which of the interworking standards are implemented in his application.

Examples of filled in proforma are in Volume 5 "Certification".

2.3.2.3 Description of Device - EIS-Functionality

It shall be kept in mind, that the functionality of a connection between e.g. a sensor and an actuator depends not only on the EIB-functions defined in clause 2.6, but in addition also to the implemented parameters in the sensor and/or the actuator. Therefore, for a given application, the devices shall be selected very carefully and in strong correlation to this application.

The description of the EIS functionality declares which device-functions are related to the declared interworking standard.

The declaration shall be made according to the PIXIT requirements given in Volume 5 "Certification".

The declared information shall be checked in the EIS conformity test set-up, as described in Volume 5 "Certification".

2.3.2.4 Description of Non-EIB Standardized Functions

For function areas where an EIS has not (yet) been agreed and/or is not yet in preparation a manufacturer shall clearly define and describe the functions implemented within his application in great detail to permit EIB-certification of that product.

This description shall be given by filling in the PIXIT proforma of Volume 5 "Certification" and shall in addition contain the description of the used data-format.

2.4 Physical Values

A physical value consists of the value itself and the unit of measurement. The values are transmitted via the bus, the units of measurement are not transmitted.

The encoding of the values for transmission are described in the interworking standard EIS 5 (see clause 2.6.6) and EIS 9 (see clause 2.6.10). These EIS also provide means for handling the value and the units of measurement within software tools.

2.5 Recommendations

1. Communication objects should not be used bidirectionally.

2. If a device provides status-information, one (or more) separate communication object(s) should be used.

3. It is recommended, that necessary time delays are located in the actuators or the application controller and not in the sensors.

2.6 EIB-Function Typing

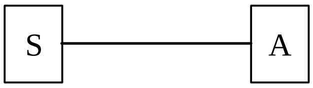
2.6.1 General

The name of the EIB-function is related to the first application, in which the EIB-function is used. But this EIB-function may also be used for other applications, e.g. a heating control may be realized with the help of the EIB-function "Dimming" (see EIS 2). Therefore each EIB-function may support various applications.

In addition, a given application can make use of more than one EIB-function. An example is given under EIS 2 "dimming".

EIS OVERVIEW		
EIS Nr.	**EIB-Function**	**Clause**
EIS 1	switching	2.6.2
EIS 2	dimming	2.6.3
EIS 3	time	2.6.4
EIS 4	date	2.6.5
EIS 5	value	2.6.6
EIS 6	scaling	2.6.7
EIS 7	drive control	2.6.8
EIS 8	priority	2.6.9
EIS 9	float value	2.6.10
EIS 10	16-bit counter value	2.6.11
EIS 11	32-bit counter value	2.6.12
EIS 12	access	2.6.13
EIS 13	ASCII Character	2.6.14
EIS 14	8-bit counter value	2.6.15
EIS 15	Character String	2.6.16

2.6.2 EIB-Function EIS 1: Switching



2.6.2.1 Description and Purpose of the EIB-Function

With help of the one bit EIB-function "Switching", it is possible to switch a load, connected to the actuator. This EIS may also be used to clear or set bit flags, e.g. enable/disable an EIB-function.

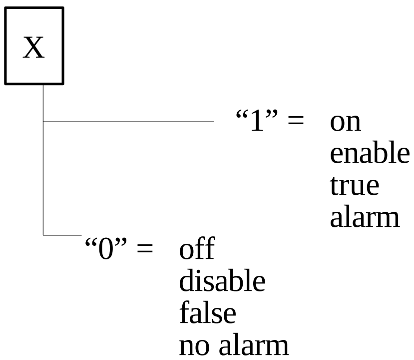
2.6.2.2 Value Size

1 bit

Code:	10	(on / off)
		(enable / disable)
		(alarm / no alarm)
		(true / false)

Symbol: EIB_Switch

2.6.2.3 Definition



2.6.2.4 Remarks

All other functions are realized by setting parameters, e.g. on/off-delays, time switching, toggling. These parameters may be selected by the manufacturer.

2.6.3 EIB-function EIS 2: Dimming

2.6.3.1 Description and Purpose of the EIB-Function

The EIB-function "dimming" consist of three EIB-subfunctions, called : "position" (switching on/off and status), "control" (relative dimming) and "value" (absolute dimming).

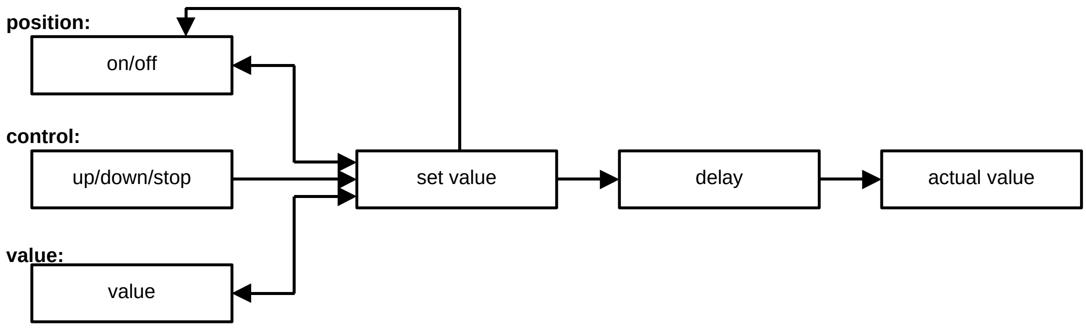
The EIB-sub function "position" supports the switching condition of the dimming actuator.

The EIB-sub function "control" is used to increase or decrease the set value in the dimming actuator (relative dimming). Via this EIB-sub function it is possible to switch the dimming actuator on, but not to switch it off.

The EIB-sub function "value" directly effects the set value (absolute dimming). Via this EIB-sub function, it is possible to switch the dimming actuator on and off.

For a dimming actuator, all of the three EIB-sub functions shall be implemented, for a dimming sensor this is not required.

Regardless of the other EIB-sub functions in a dimming actuator, the last EIB-sub function received shall be executed.



2.6.3.2 EIB Sub-function Position

Description and purpose of the EIB-sub function

With the EIB-sub function "position", it is possible to switch a dimming actuator on and off with a simple EIB-switch. This EIB-sub function reflects the on/off status of the actuator.

On every change of the on/off status, regardless of the trigger event, and on every write to the EIB-sub function "position" via the bus, a send-request for this EIB-sub function will be given.

Value size

See EIS 1.

Definition

See EIS 1 on/off.

Remarks

1. In a group of dimming actuators, only one of them may send back its status on the same group address.
2. By choosing related communication parameters in the BAU it is possible to prevent the sending of the status. "Send request" should be disabled by default.

2.6.3.3 EIB Sub-function Control

Description and purpose of the EIB-sub function

With the help of the EIB-sub function "control", it is possible to increase or decrease the set value in steps, or stop the movement. The stepcode is also transmitted by this EIB-sub function "control". "Step-break" denotes, that the actuator stops at the given value.

It is possible to switch on an actuator by using the EIB-sub function "control".

Value size

4 bit

Code: 20

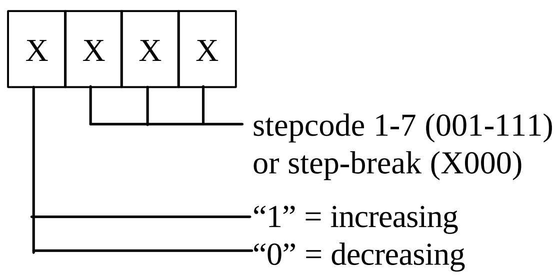
Symbol: EIB_dimming_control

Definition

The stepcode indicates the amount of intervals into which the range of value 0...100% is subdivided.

Numbers of intervals = 2 (stepcode - 1)

EIB_Dimming_Control



Remarks

None.

2.6.3.4 EIB Sub-function Value

Description and purpose of the EIB-sub function

With the EIB-sub function "value", a position within the range between "value low" and 100% can be set directly. By using the EIB-sub function "value", it is possible to switch on an actuator by writing a value <>"0" and switch off by writing a "0".

Value size

8 bit, see EIS 6

Code:	6001	see EIS 6
Symbol:	EIB_scaling_lux	see EIS 6

Definition

See EIS 6 and clause 2.6.3.5.

Remarks

This EIB-sub function "value" does not reflect the status of the dimming actuator. It can be different from the set- or the actual value.

2.6.3.5 Behavior

Status

off	dimming actuator switched off
on	dimming actuator switched on, constant brightness, at least minimal brightness dimming
dimming	actuator switched on, moving from actual value in direction of set value

Events

position = 0	off command
position = 1	on command
control = up dX	command, dX more bright dimming
control = down dX	command, dX less bright dimming
control = stop	stop command
value = 0	dimming value = off
value = x%	dimming value = x% (not zero)
value_reached	actual value reached set value

The step size dX for up and down dimming may be 1/1, 1/2, 1/4, 1/8, 1/16, 1/32 and 1/64 of the full dimming range (0 - FFh).

Parameters

The standard function requires no parameters at all.

Standard behavior of an dimming actuator:

DESCRIPTION	STANDARD VALUES
dimming speed	a sweep from 0 to 100% in 4 sec shall be possible
turn-on condition	set value = FFh
reaction on absolute dimming values	jmp = jump to value
reaction on bus at power-fail and return	off = switch off when power fails, no action when power returns.

If parameters are implemented, the standard behavior above shall be included.

Data

The dimming actuator works internally with the set- and the actual value. The actual value follows the set value. By dimming the follow up is delayed.

Status Diagram

In the status diagram only the standard transitions are given. Parameter dependent transitions are not given. Where "send status" is written, it is recommended to give the send-request during the event processing.

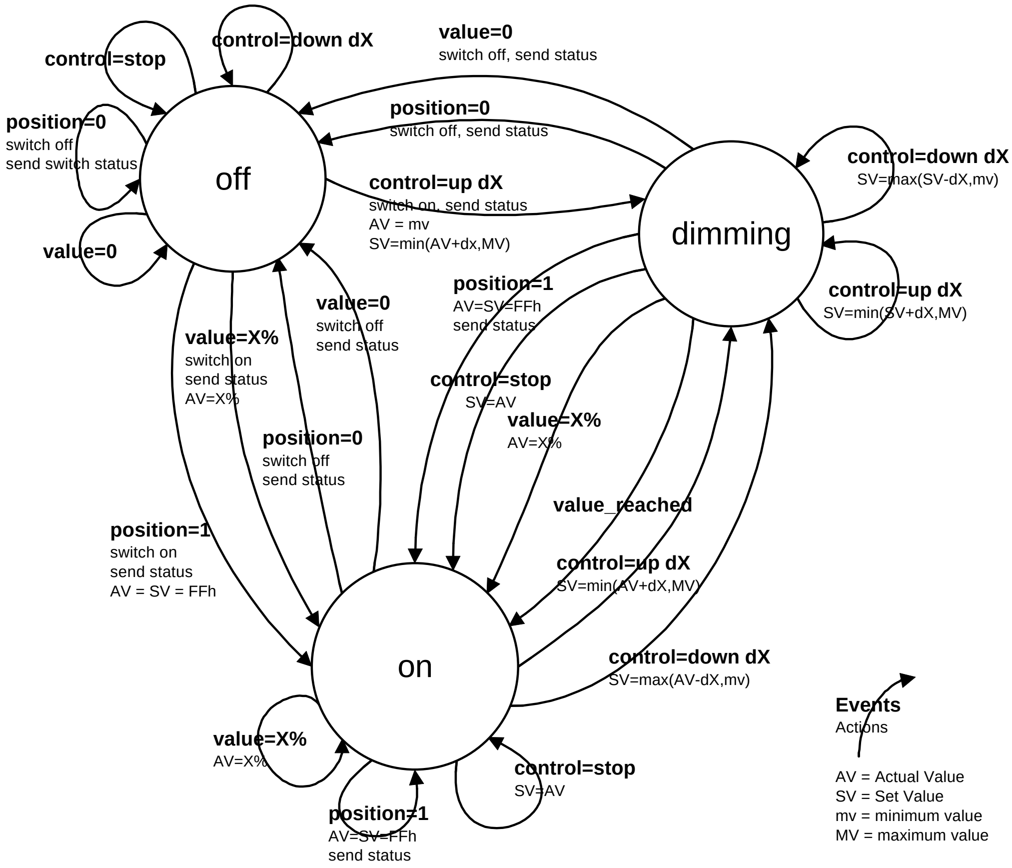
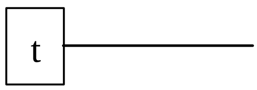


Fig. 3/7/1-2: Dimmer Status Diagram

2.6.4 EIB-Function EIS 3: Time



2.6.4.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_time" gives the current time to all EIB devices which process this information.

2.6.4.2 Value Size

3 bytes

Code: 30

Symbol: EIB_time

2.6.4.3 Definition

EIB_Time_of_Day	d	d	d	h	h	h	h	h	1. byte

d = day (1=monday,7=sunday)(0=no day)
h = hours (binary 0…23)

0	0	m	m	m	m	m	m	2. byte

m = minutes (binary 0…59)

0	0	s	s	s	s	s	s	3. byte

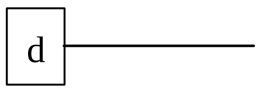
s = seconds (binary 0…59)

2.6.4.4 Remarks

The EIB-function "EIB_time" is sent cyclically by a system clock. It may be also possible to read the "EIB_time".

It shall be possible to adjust the transmission rate of a system clock device by a parameter in the EEPROM.

2.6.5 EIB-Function EIS 4: Date



2.6.5.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_date" gives the current date to the EIB devices which process this information.

2.6.5.2 Value Size

3 bytes

Code: 400

Symbol: EIB_date

2.6.5.3 Definition

EIB_date

0	0	0	D	D	D	D	D	1. byte

D = day (binary 1 .. 31)

0	0	0	0	M	M	M	M	2. byte

M = month (binary 1 .. 12)

Y	Y	Y	Y	Y	Y	Y	Y	3. byte

Y = Year (binary 0 .. 255)[1]

2.6.5.4 Remarks

2.6.5.4.1 Century Encoding

The following interpretation shall be carried out by devices receiving the underneath EIS 4 telegram and carrying out calculations on the basis of the entire 3rd byte:

Byte 1								Byte 2								Byte 3							
0	0	0	D	D	D	D	D	0	0	0	0	M	M	M	M	0	Y	Y	Y	Y	Y	Y	Y

if Byte 3 contains value ≥ 90 : interpret as 20th century

if Byte 3 contains value < 90: interpret as 21st century

This format covers the range 1990 to 2089.

Example:

YYYYYYY = 99_d equals 1999

YYYYYYY = 0_d equals 2000

YYYYYYY = 4_d equals 2004

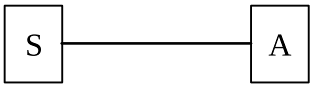
2.6.5.4.2 System Clock

The EIB-function "EIB_date" is cyclically sent by a system clock. It is also possible to read the "EIB_date".

It shall be possible to adjust the transmission rate of the system clock by means of an application parameter.

[1] 0 = year 1900
255 = year 2155

2.6.6 EIB-Function EIS 5 : Value



2.6.6.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Value" is used to transmit values representing physical values via the bus.

2.6.6.2 Value Size

2 bytes

Code:	50XX	XX = see clause 2.6.6.5
Symbol:	EIB_value_X	X = see clause 2.6.7.5

2.6.6.3 Definition

EIB_value	S	E	E	E	E	M	M	M	M	M	M	M	M	M	M	M

S = Sign of the mantissa.
E = Exponent in basis 2 (0...15).
M = Mantissa in a two's complement representation (-2048 ... 0 ... 2047).
EIB_value = $(-1)^{(S)} * (0.01*M) * 2^{(E)}$
Range of value : - 671 088.64 ... 0 ... + 670 760.96
Resolution : $0.01 * 2^{(\text{Exponent})}$

2.6.6.4 Remarks

Simple applications

For simple applications, it is possible to generate numbers with a fixed exponent, however only in sensors.

<u>Example</u>: In a temperature measurement application, a range of ±160 (°C) and a resolution of 0.1 (°C) is sufficient; a fixed exponent of 3 is appropriate.

Before submitting an EIB product for certification, of which the attributed application program contains an EIS 5 communication object (with communication way IN) supporting only one fixed component (e.g. 3), the applicant shall first address an application for a system extension to the EIBA System Group.

This topic is still under discussion in EG 3.5 and the System Group.

How to code "EIS 5"/Value data

- **Two's complement**

Two's complement allows the encoding of negative values.

- **Principle**

A negative value is obtained by :

1. first inverting ALL bits of the absolute value (i.e. 1 → 0, 0 → 1)
2. then adding 1

This is the two’s complement notation for negative values.

Example: Decimal "10" has as its binary representation 01010; to obtain "-10".

invert 01010 → 10101

add 10101+1 = 10110

(the 1st bit of the result is the so-called sign-bit : 1 for negative values, 0 for positive ones*)*

- **Table applied to EIS 5**

S	E	E	E	E	M	M	M	M	M	M	M	M	M	M	M	Value
0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	2047
0	X	X	X	X	1	1	1	1	1	1	1	1	1	1	0	2046
0	X	X	X	X	0	0	0	0	0	0	0	0	0	1	1	3
0	X	X	X	X	0	0	0	0	0	0	0	0	0	1	0	2
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	1	1
0	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0
1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	1	-1
1	X	X	X	X	1	1	1	1	1	1	1	1	1	1	0	-2
1	X	X	X	X	1	1	1	1	1	1	1	1	1	0	1	-3
1	X	X	X	X	0	0	0	0	0	0	0	0	0	0	1	-2047
1	X	X	X	X	0	0	0	0	0	0	0	0	0	0	0	-2048

- **Application**

To encode the value -30°C with a precision of 1/100:

– first rewrite the value

-30°C = -3000.1/100°C

(precision determined implicitly by the value)

– the 11 bit mantissa allows a maximum value of 2047 to be encoded; in this case, we will have to introduce an exponent of 2^1, since 3000 > 2047:

$$3000 = \underbrace{1500}_{\text{mantissa}} \cdot \underbrace{2^1}_{\text{exponent}}$$

1500 is encoded as

$$\underbrace{0}_{\text{sign}} \quad \underbrace{101\ 1101\ 1100}_{\text{mantissa}}$$

– changing the sign as of 1500 to -1500 gives:

$$\underbrace{1}_{\text{sign}} \quad \underbrace{010\ 00100100}_{\text{mantissa}}$$

– as the final result we get:

S	E	E	E	E	M	M	M	M	M	M	M	M	M	M	M	Value
1	0	0	0	1	0	1	0	0	0	1	0	0	1	0	0	-3000

2.6.6.5 Units of Measurement

General

The units of measurement are not transmitted via the bus. Nevertheless, they are part of the EIB-function "EIB_value". With the help of the same code ("50XX") and symbol ("EIB_value_X”) for sensors and actuators, a proper communication is possible.

Codes and units

UNIT of MEASUREMENT	CODE	SYMBOL	RANGE OF VALUE RESOLUTION
temperature	5001	EIB_value_temp	-273...+670760 °C
temperature difference	5002	EIB_value_temd	+/- 670760 K
temperature gradient	5003	EIB_value_temg	+/- 670760 K/h
intensity of light	5004	EIB_value_lux	0...670760 Lux
wind speed	5005	EIB_value_wsp	0...670760 m/s
air pressure	5006	EIB_value_pres	0...670760 Pa
time difference	5010	EIB_value_time1	+/- 670760 s
time difference	5011	EIB_value_time2	+/- 670760 ms
voltage	5020	EIB_value_volt	+/- 670760 mV
electrical current	5021	EIB_value_curr	+/- 670760 mA

Other units of measurement are under consideration.

2.6.7 EIB-Function EIS 6: Scaling

2.6.7.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_scaling" is used to transmit relative values with a resolution of 8 bit.

This EIB-function represents, e.g. the relative brightness in the range between "value low" and 100%.

2.6.7.2 Value Size

8 bit

Code: 60XX

Symbol: EIB_scaling_X

2.6.7.3 Definition

EIB_scaling	X	X	X	X	X	X	X	X	
	0	0	0	0	0	0	0	0	= reserved / off
	0	0	0	0	0	0	0	1	= “value low”
		.. ..							
	1	1	1	1	1	1	1	1	= 100%

2.6.7.4 Remarks

None.

2.6.7.5 Units of Measurement

UNIT of MEASUREMENT	CODE	SYMBOL	RANGE OF VALUE
relative brightness	6001	EIB_scaling_lux	0...100%
relative humidity	6002	EIB_scaling_wet	0...100%
wind direction	6003	EIB_scaling_wdr	0...360°

Other units of measurement are under consideration.

2.6.8 EIB-Function EIS 7: Drive Control

2.6.8.1 Description and Purpose of the EIB-Function

The EIB-function "drive control" consists of two EIB-sub functions called "move" and "step".

With help of the EIB-sub function "move", it is possible to set the drive in motion or to change the direction of the movement.

The EIB-sub function "step" is used to stop the movement. In addition, a gradual movement is possible.

2.6.8.2 EIB Sub-function Move

Description and purpose of the EIB-sub function

With the help of the EIB-sub function "move", it is possible to set the drive in motion or to change the direction of the movement.

Value size

1 bit

Code: 70

Symbol: EIB_drive_move

Definition

EIB_drive_move [X]

"1" = move down / close
"0" = move up / open

Remarks

No member of a group shall answer to a read-access from the communication system as an answer may cause a stopped drive to be set in motion.

2.6.8.3 EIB Sub-function Step

Description and purpose of the EIB-sub function

With help of the EIB-sub function "step", it is possible to stop the drive in motion or to step the movement.

Value size

1 bit

Code:	71XX
Symbol:	EIB_drive_step

Definition

EIB_drive_step [X]

"1" = stop/step down
"0" = stop/step up

Remarks

No member of a group should answer on a read-access from the communication system, since an answer may cause a **moved drive** in steps to stop or a stopped drive to move in steps.

2.6.8.4 Behavior

Status

stopped	no movement
moving	movement
stepping Up	stepwise movement up
stepping Down	stepwise movement down

Events

move = 0	movement upwards
move = 1	movement downwards
step = 0	stop or stepwise movement up
step = 1	stop or stepwise movement down
time-out =	time of movement for full range or stepping time is over

Parameters

The standard version has no parameters at all.

Standard behavior of an actuator is :

DESCRIPTION	STANDARD VALUES
move-time specifies the duration of moving the drive	no limit or at least "very" long time
step-time specifies the time of one step	no step or at least "very" small time

If parameters are implemented, the above-mentioned behaviors should be included.

Status diagram

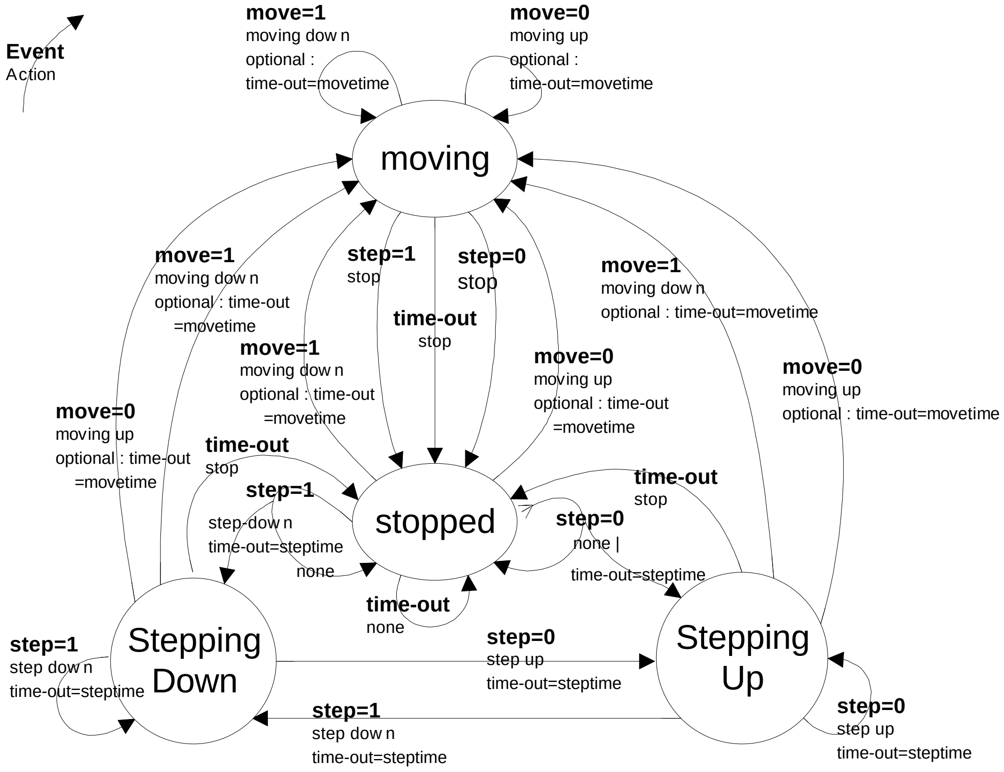


Fig. 3/7/1-3: EIS 7 state diagram

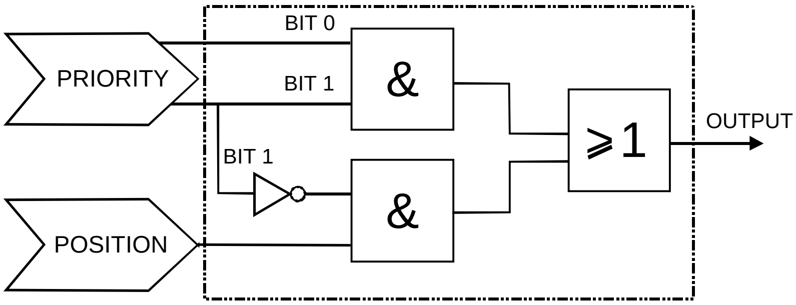
2.6.9 EIB-Function EIS 8: Priority

2.6.9.1 Description and Purpose of the EIB-Function

The EIB-function "Priority" consists of two EIB-subfunctions called : "EIS_priority_position" (switching on/off) and "EIS_priority_control".

The EIB-sub function "EIS_priority_position" supports the switching condition of the application concerned.

The EIB-sub function "EIS_priority_control" is used either to impose the output value (always ON or always OFF) or to link the output directly to the EIB-sub function "EIS_priority_position" status.



Note: If timers or logical connections are implemented in an actuator, the influence on changes of an EIS 8 are not defined !

2.6.9.2 Sub-function EIS_Priority_Position

Description and purpose of the EIB-sub function EIS_priority_position

With the EIB-sub function "EIS_priority_position", it is possible to switch an actuator on and off under control of EIS-sub function "EIS_priority_control".

Value Size

See EIS 1

Definition

See EIS 1 on/off

Remarks

None

2.6.9.3 Sub-function EIS_Priority_Control

Description and purpose of the EIS-sub function EIS_priority_control

The EIB-sub function "EIS_priority_control" allows priority-control of an actuator in conjunction with an EIS 1. If the control is set up, the EIS-sub function imposes the output status, otherwise the output is driven out by the EIB-sub function "EIS_priority_position".

Value size

2 bits

Code:	80
Symbol:	EIS_priority_control

Definition

X X

"0" = off, disable (Bit 0)
"1" = on, enable

"0" = no control (Bit 1)
"1" = "control

Bit 1	Bit 0	
0	0	no control
0	1	no control
1	0	control, function value 0 (off; disable)
1	1	control, function value 1 (on, enable)

Remarks

None

2.6.9.4 Behavior

Status

The output status results from a combination of the 2 EIB-sub functions (see truth table).

Events

(see paragraph truth table)

Parameters

The standard function requires no parameter at all.

Data

(see truth table)

Truth tables

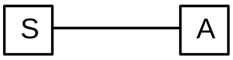
The internal behavior is tightly linked to the two EIB-sub function status as defined in the truth table below:

EIS_Priority_control		EIS_priority	
Bit 1	Bit 0	_position	Output
0	X[2]	0	0
0	X[2]	1	1
1	0	X[2]	0
1	1	X[2]	1

The EIS 8 works on an actuator as follows:

> If the bit 1 of EIS_priority_control is set to "O", then the value of bit 0 of EIS_priority_control does not matter. The output of the actuator reacts on the EIS_priority_position value.
>
> If the bit 1 of EIS_priority_control is set to “1", the output of the actuator reacts on the value of bit 0 of EIS_priority_control. The EIS_priority_position value does not matter.

2.6.10 EIB-Function EIS 9: Float Value



2.6.10.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Float Value" is used to transmit data representing physical values via the bus in the IEEE floating point format.

2.6.10.2 Value Size

4 bytes

Code:	90XX	XX = see clause 2.6.10.4
Symbol:	EIB_value_X	X = see clause 2.6.10.4

[2] X = don’t care

2.6.10.3 Definition

The values are encoded in the IEEE-floating point format according to IEEE 754.

The byte order is as follows:

Bit	31	30 23	22		0
	Sign	exponent	mantissa		
Byte	1		2	3	4

2.6.10.4 Units of Measurement

General

The units of measurement are not transmitted via the bus. Nevertheless, they are part of the EIB-function "EIB_FloatValue". With the help of the same code ("90XX") and symbol ("EIB_value_X) for sensors and actuators, a setup of a proper communication is possible.

Codes and units

UNIT of MEASUREMENT	CODE	SYMBOL	UNITS
acceleration	9000	EIB_value_acceleration	m s^{-2}
acceleration, angular	9001	EIB_value angular Acceleration	rad s^{-2}
activation energy	9002	EIB_value_activation Energy	J mol^{-1}
activity (radioactive)	9003	EIB_value_activity	s^{-1}
amount of substance	9004	EIB_value_mol	mol
amplitude	9005	EIB_value_amplitude	(unit as appropriate)
angle, radiant	9006	EIB_value_angleRad	rad
angle, degree	9007	EIB_value_angleDeg	°(degrees)
angular momentum	9008	EIB_value_angular Momentum	J s
angular velocity	9009	EIB_value_angular Velocity	rad s^{-1}
area	9010	EIB_value_area	m^2
capacitance	9011	EIB_value_capacitance	F
charge density (surface)	9012	EIB_value_charge DensitySurface	C m^{-2}
charge density (volume)	9013	EIB_value_charge DensityVolume	C m^{-3}
compressibility	9014	EIB_value_ compressibility	m^2 N^{-1}
conductance	9015	EIB_value_ conductance	S = Ω^{-1}
conductivity, electrical	9016	EIB_value_ electrical Conductivity	S m^{-1}
density	9017	EIB_value_density	kg m^{-3}
electric charge	9018	EIB_value_electric Charge	C
electric current	9019	EIB_value_electric Current	A

UNIT of MEASUREMENT	CODE	SYMBOL	UNITS
electric current density	9020	EIB_value_electric CurrentDensity	A m^{-2}
electric dipole moment	9021	EIB_value_electric DipoleMoment	C m
electric displacement	9022	EIB_value_electric Displacement	C m^{-2}
electric field strength	9023	EIB_value_electric FieldStrength	V m^{-1}
electric flux	9024	EIB_value_electric Flux	C
electric flux density	9025	EIB_value_electric FluxDensity	C m^{-2}
electric polarization	9026	EIB_value_electric Polarization	C m^{-2}
electric potential	9027	EIB_value_electric Potential	V
electric potential difference	9028	EIB_value_electric PotentialDifference	V
electromagnetic moment	9029	EIB_value_ electromagnetic Moment	A m^{2}
electromotive force	9030	EIB_value_ electromotiveForce	V
energy	9031	EIB_value_energy	J
force	9032	EIB_value_force	N
frequency	9033	EIB_value_frequency	Hz = s^{-1}
frequency, angular (pulsatance)	9034	EIB_value_angular Frequency	rad s^{-1}
heat capacity	9035	EIB_value heat Capacity	J K^{-1}
heat flow rate	9036	EIB_value_heatFlow Rate	W
heat, quantity of	9037	EIB_value_heat Quantity	J
impedance	9038	EIB_value_ impedance	Ω
length	9039	EIB_value_length	m
light, quantity of	9040	EIB_value_light Quantity	J or lm s
luminance	9041	EIB_value_luminance	cd m^{-2}
luminous flux	9042	EIB_value_luminous Flux	lm
luminous intensity	9043	EIB_value_luminous Intensity	cd
magnetic field strength	9044	EIB_value_magnetic FieldStrength	A m^{-1}
magnetic flux	9045	EIB_value_magnetic Flux	Wb
magnetic flux density	9046	EIB_value_magnetic FluxDensity	T
magnetic moment	9047	EIB_value_magnetic Moment	A m^{2}
magnetic polarization	9048	EIB_value_magnetic Polarization	T
magnetization	9049	EIB_value_ magnetization	A m^{-1}
magnetomotive force	9050	EIB_value_ magnetomotiveForce	A
mass	9051	EIB_value_mass	kg
mass flux	9052	EIB_value_massFlux	kg s^{-1}

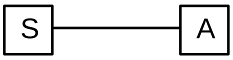
UNIT of MEASUREMENT	CODE	SYMBOL	UNITS
momentum	9053	EIB_value_ momentum	N s^{-1}
phase angle, radiant	9054	EIB_value_phase AngleRad	rad
phase angle, degrees	9055	EIB_value_phase AngleDeg	° (degrees)
power	9056	EIB_value_power	W
power factor (cos Φ)	9057	EIB_value_power Factor	
pressure	9058	EIB_value_pressure	Pa (= N m^{-2})
reactance	9059	EIB_value_reactance	Ω
resistance	9060	EIB_value_resistance	Ω
resistivity	9061	EIB_value_resistivity	Ωm
self inductance	9062	EIB_value_self Inductance	H
solid angle	9063	EIB_value_solidAngle	sr
sound intensity	9064	EIB_value_sound Intensity	W m^{-2}
speed	9065	EIB_value_speed	m s^{-1}
stress	9066	EIB_value_stress	Pa (= N m^{-2})
surface tension	9067	EIB_value_surface Tension	N m^{-1}
temperature, common	9068	EIB_value_common temperature	°C
temperature (absolute)	9069	EIB_value_absolute temperature	K
temperature difference	9070	EIB_value_ temperatureDifference	K
thermal capacity	9071	EIB_value_ thermal Capacity	J K^{-1}
thermal conductivity	9072	EIB_value_thermal Conductivity	W m^{-1} K^{-1}
thermoelectric power	9073	EIB_value_ thermoelectricPower	V K^{-1}
time[3]	9074	EIB_value_time	s
torque	9075	EIB_value_torque	N m
volume	9076	EIB_value_volume	m^3
volume flux	9077	EIB_value_volume Flux	m^3 s^{-1}
weight	9078	EIB_value_weight	N
work	9079	EIB_value_work	J

Note: The EIB_value_time shall not be used for transmitting the time of day. For the time of day EIS 3 format shall be used.

Other units of measurement are under consideration.

[3] For proper usage see note!

2.6.11 EIB-Function EIS 10: 16-bit Counter Value



2.6.11.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Counter 16-bit" is used to transmit values representing 16-bit counter values via the bus. The EIB-function EIS 11, which is described in the following paragraph uses 32 bits.

2.6.11.2 Value Size

Counter Type	Code	Symbol	Size
unsigned counter value	10000	EIB_value_ucount	16-bit unsigned integer
signed counter value	10001	EIB_value_count	16-bit signed integer

2.6.11.3 Definition

The negative values of the signed counter values are encoded in the two's complement.

The counter values are encoded as follows:

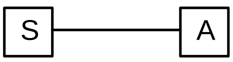
Signed counter value:

Bit	15	14	0
	Sign	binary number	
Byte	1		2

Unsigned counter value:

Bit	15	0
	binary number	
Byte	1	2

2.6.12 EIB-Function EIS 11: 32-bit Counter Value



2.6.12.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Counter 32-bit" is used to transmit values representing 32-bit counter values via the bus. The EIB-function EIS 10, which is described in the previous chapter uses 16 bits.

2.6.12.2 Value Size

Counter Type	Code	Symbol	Size
unsigned long counter value	11000	EIB_value_ulcount	32-bit unsigned integer
signed long counter value	11001	EIB_value_lcount	32-bit signed integer

2.6.12.3 Definition

The negative values of the signed counter values are encoded in the two’s complement.

The counter values are encoded as follows:

Signed long counter value:

Bit	31	30	0
	Sign	binary number	

Byte	1	2	3	4

Unsigned long counter value:

Bit	31	0
	binary number	

Byte	1	2	3	4

2.6.13 EIB-Function EIS 12: Access

2.6.13.1 Description and Purpose of the EIB-Function

The EIB-function "Access" is used in access gratification procedures. It regulate the following subjects:

1. Central recording and/or time and motion study of access control actions;
2. to trigger single objects by certain access data (e.g. light, heating, alarm functions, etc.);
3. to display access actions in relation to the used access device at tableaus, PC-Screen, etc.

2.6.13.2 Value Size

4 bytes.

Function	Code	Symbol	Size
Access	12.000	EIB_Access	4 bytes

Please refer to the definition for the strucure eand the encoding.

2.6.13.3 Definition

1. **Byte 1 (MSB)**

7	6	5	4	3	2	1	0
B53	B52	B51	B50	B43	B42	B41	B40

B53...B50 Digit 6 of access identification code; range 0...9

Most significant digit of access identification code.

B43...B40 Digit 5 of access identification code; range 0...9

2. **Byte 2**

7	6	5	4	3	2	1	0
B33	B32	B31	B30	B23	B22	B21	B20

B33...B30 Digit 4 of access identification code; range 0...9

B23...B20 Digit 5 of access identification code; range 0...9

3. **Byte 3**

7	6	5	4	3	2	1	0
B13	B12	B11	B10	B03	B02	B01	B00

B13...B10 Digit 2 of access identification code; range 0...9

B03...B00 Digit 1 of access identification code; range 0...9.

Least significant digit of access identification code;

4. Byte 4 (LSB)

7	6	5	4	3	2	1	0
E	P	D	C	N3	N2	N1	N0

E: Detection error
1 = error detected

P: Permission
0 = not accepted
1 = accepted

D: Read direction (e.g. of badge)
0 = left to right
1 = right to left
If not used (e.g. electronic key) set to zero.

C: Access information encrypted
0 = no
1 = yes

N3...N0: Index of access identification code

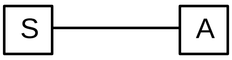
2.6.13.4 Remarks

- **Byte 1 up to Byte 3**

These are the carrier of the access information code. Only a card or key number should be used. System number, version number, country code, etc are not necessary. Ciphered access information code should be possible in principle. If 24 bits are not necessary, the most significant positions shall be set to zero.

- **Byte 4**
  - the first bit in Byte 4 is used to inform that a detection error has occurred (reading of access information code was not successful).
  - Bit P informs about the access decision which is made by the controlling device.
  - Bit D represents special information concerning the source or reader.
  - Bit C gives information whether the access information code in Byte 1 up to
  - Byte 3 is clear (normal) or encrypted.
  - Bit N3 .. N0 are for future use.

2.6.14 EIB-Function EIS 13: EIB-ASCII-Char



2.6.14.1 Description and Purpose of the EIB-Function

The EIB-function "EIB-ASCII-Char" is used to transmit a single textual character from a sensor.

2.6.14.2 Value Size

Function	Code	Symbol	Size
ASCII-Character	13.000	EIB-ASCII-Char	8 bit

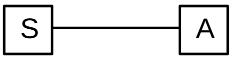
2.6.14.3 Definition

Length:	8 bits (characters to be sent as individual bytes - not grouped to strings in one communication object)
Format:	MSB LSB 0AAAAAAA
Value range:	[0…127]

Symbol:	Code:	AAAAAAA
EIB_ASCII_Char	13.00	(see table below)

<table>
<tr><td>MSN</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>LSN</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>0</td><td>NUL</td><td>DLE</td><td></td><td>0</td><td>@</td><td>P</td><td>'</td><td>p</td></tr>
<tr><td>1</td><td>SOH</td><td>DC1</td><td>!</td><td>1</td><td>A</td><td>Q</td><td>a</td><td>q</td></tr>
<tr><td>2</td><td>STX</td><td>DC2</td><td>"</td><td>2</td><td>B</td><td>R</td><td>b</td><td>r</td></tr>
<tr><td>3</td><td>ETX</td><td>DC3</td><td>#</td><td>3</td><td>C</td><td>S</td><td>c</td><td>s</td></tr>
<tr><td>4</td><td>EOT</td><td>DC4</td><td>$</td><td>4</td><td>D</td><td>T</td><td>d</td><td>t</td></tr>
<tr><td>5</td><td>ENQ</td><td>NAK</td><td>%</td><td>5</td><td>E</td><td>U</td><td>e</td><td>u</td></tr>
<tr><td>6</td><td>ACK</td><td>SYN</td><td>&</td><td>6</td><td>F</td><td>V</td><td>f</td><td>v</td></tr>
<tr><td>7</td><td>BEL</td><td>ETB</td><td>'</td><td>7</td><td>G</td><td>W</td><td>g</td><td>w</td></tr>
<tr><td>8</td><td>BS</td><td>CAN</td><td>(</td><td>8</td><td>H</td><td>X</td><td>h</td><td>x</td></tr>
<tr><td>9</td><td>HT</td><td>EM</td><td>)</td><td>9</td><td>I</td><td>Y</td><td>i</td><td>y</td></tr>
<tr><td>A</td><td>LF</td><td>SUB</td><td>*</td><td>:</td><td>J</td><td>Z</td><td>j</td><td>z</td></tr>
<tr><td>B</td><td>VT</td><td>ESC</td><td>+</td><td>;</td><td>K</td><td>[</td><td>k</td><td>{</td></tr>
<tr><td>C</td><td>FF</td><td>FS</td><td>'</td><td><</td><td>L</td><td>v</td><td>l</td><td>|</td></tr>
<tr><td>D</td><td>CR</td><td>GS</td><td>-</td><td>=</td><td>M</td><td>]</td><td>m</td><td>}</td></tr>
<tr><td>E</td><td>SO</td><td>RS</td><td>.</td><td>></td><td>N</td><td>^</td><td>n</td><td>~</td></tr>
<tr><td>F</td><td>SI</td><td>US</td><td>/</td><td>?</td><td>O</td><td>_</td><td>o</td><td>DEL</td></tr>
</table>

2.6.15 EIB-Function EIS 14: 8-bit Counter



2.6.15.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Counter 8-bit" is used to transmit values representing 8-bit counter values via the bus. The EIB-function EIS10, EIS11 and EIS 14 are used as an EIB counting function. The choice of one of this three EIB-function "EIB_Counter" depends on the value range the counter shall contain.

2.6.15.2 Value Size

Counter Type	Code	Symbol	Size
unsigned counter value	14.000	EIB_value_ucount	8-bit unsigned char
signed counter value	14.001	EIB_value_count	8-bit signed char

2.6.15.3 Definition

The negative values of the signed counter values are encoded in the two’s complement.

The counter values are encoded as follows:

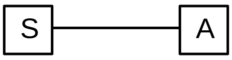
Signed counter value:

Bit	7	6 … 0
	Sign	binary number

Unsigned counter value:

Bit	7 … 0
	binary number

2.6.16 EIB-Function EIS 15: Character String



2.6.16.1 Description and Purpose of the EIB-Function

The EIB-function "EIB_Character_String" is used to transmit values representing character strings.

2.6.16.2 Value Size

Function	Code	Symbol	Size
Character string	15.000	EIB_character_string	MAXDATA

2.6.16.3 Definition

The string is transmitted in a field of length MAXDATA (14 Bytes). The contents of each of the bytes shall comply with the format as defined by the EIS type 13 ‘ASCII character’. If the string to be transmitted is less than the full 14 bytes long, unused trailing bytes in the MAXDATA character string shall be set to NULL.

2.6.16.4 Example

‘EIB is OK’ is encoded as follows : 45 49 42 20 69 73 20 4F 4B 00 00 00 00 00.

2.6.16.5 Warning

When MAXDATA values are received by communication objects with a smaller data type (e.g. DATA10), the corresponding User RAM of other communication objects may be overwritten by the BCU system software. This is the case when using BCU 1 and BCU 2.

2.6.17 Partial Implementation of EIS

Partial implementation of EIS is **only** permitted in communication objects with a communication way OUT (in general sensor functions). Consequently, a sensor application program consisting of communication objects only capable of transmitting EIS 1 on telegrams can be declared as EIS and can be certified. This for instance also applies to all EIS consisting of a number of subfunctions, of which some have been omitted to realise a sensor function.

However, partial implementation of EIS 1 to EIS 8 in objects with a communication way in is neither acceptable nor can it be declared as a non-EIB standardised communication object. Application programs implementing functional communication objects for which a corresponding EIS exists but which are declared as non-EIB will be rejected for certification.

2.6.18 Implementation of Group Addresses Ex-Factory

It is strongly recommended to deliver devices without any group addresses.

If a company sees the need for implementation of a group address or group addresses, it shall be selected within the range F000H to FFFDH.

The following addresses are selected, application oriented, in detail:

Group address	Reserved for
$F000_H$	light switching on/off
$F001_H$	dimming more/less brightness
$F002_H$	dimming set value
$F003_H$	dimming read status
$F004_H$	shutter control up/down
$F005_H$	shutter control lamellas open/closed
$F006_H$	shutter control wind sensor
$F007_H$	heating control on/off
$F008_H$	binary I/O priority control
$F009_H$	light scenario programming
$F00A_H$	light scenario programming enable
$F00B_H$	light scenarios A/B
$F00C_H$	light scenarios C/D
$F00D_H$	inverter - receiving object
$F00E_H$	inverter - transmitting object
$F00F_H$	room temperature
$F010_H$	light sensor - light value
$F011_H$	light scenario A/B programming method II
$F012_H$	light scenario C/D programming method II
$F013_H$	dimming read status (8-bit value)
$F014_H$	(info-display) - alarm reset
$F015_H$	(info-display) - display-update
$F016_H$	counter - transmitting object
$F017_H$	counter - receiving object
$F018_H$	DC/AC-converter - transmitting object
$F019_H$	DC/AC-converter - receiving object

Group address	Reserved for
$F01A_H$	heating-valve - set value
$F01B_H$	temperature, regulation ratio cooling in % - 1 byte
$F01C_H$	temperature, current setpoint, 2 byte
$F01D_H$	temperature, setpoint relocation in % - 1 byte
$F01E_H$	temperature, setpoint relocation in K - 2 byte
$F01F_H$	temperature, status - 1 byte
$F020_H$	temperature, standby - 1 bit
$F021_H$	temperature, comfort/presence - 1 bit
$F022_H$	temperature, night reduction - 1 bit
$F023_H$	temperature, anti-freeze/heat-protection - 1 bit
$F024_H$	temperature, basic setpoint - 2 byte
$F025_H$	temperature, transfer heating/cooling - 1 bit
$F026_H$	temperature, dew point alarm - 1 bit
$F027_H$	master clock - time - 3 bytes according to EIS 3
$F028_H$	master clock - date - 3 bytes according to EIS 4
$F029_H$	fire alarm - 1 bit according to EIS 1
$F02A_H$	reserved for future extensions
. . .	

If a company selects one or more group addresses for implementation ex factory, it shall provide information of this in the product related documentation, e.g. data sheets and product description. In addition a hint shall be given that situations in which some 50/100 devices will switch on simultaneously shall be avoided.

3. Interworking Standard for EIB Objects

In Chapter 3/4/1 "User Layer" of this Handbook, the general structure for EIB objects is given. When implemented, the data type, code and type of a property, shall comply with the coding listed in the following tables.

Property Data Type	Type Code	Type-Length
PT_CONTROL	00h	1 Read /10 Write
PT_CHAR	01h	1
PT_UNSIGNED_CHAR	02h	1
PT_INT	03h	2
PT_UNSIGNED_INT	04h	2
PT_EIB_FLOAT	05h	2
PT_DATE	06h	3
PT_TIME	07h	3
PT_LONG	08h	4
PT_UNSIGNED_LONG	09h	4
PT_FLOAT	0Ah	4
PT_DOUBLE	0Bh	8
PT_CHAR_BLOCK	0Ch	10
PT_POLL_GROUP_SETTINGS	0Dh	3
PT_SHORT_CHAR_BLOCK	0Eh	5

Fig. 3/7/1-4: Data Types of Properties

Property-IDs	Property_Id
Reserved	0
PID_OBJECT_TYPE	1
PID_OBJECT_NAME	2
PID_SEMAPHOR	3
PID_GROUP_OBJECT_REFERENCE	4
PID_LOAD_STATE_CONTROL	5
PID_RUN_STATE_CONTROL	6
PID_TABLE_REFERENCE	7
PID_SERVICE_CONTROL	8
PID_FIRMWARE_REVISION	9
PID_SERVICES_SUPPORTED	10
PID_SERIAL_NUMBER	11
PID_MANUFACTURER_ID	12
PID_PROGRAM_VERSION	13
PID_DEVICE_CONTROL	14
PID_ORDER_INFO	15
PID_PEI_TYPE	16
PID_PORT_CONFIGURATION	17
PID_POLL_GROUP_SETTINGS	18
Reserved for global Property Ids	-50
Reserved for object type specific property ids	51-200
Reserved for application specific property ids	200-255

Fig. 3/7/1-5: Codes for Property Ids

Objecttypes	Type_Id
Device Object	0
Addresstable Object	1
Associationtable Object	2
Applicationprogramm Object	3
.	
Reserved for global Object types	-50000
Reserved for application specific Objecttypes	50001-65535

Fig. 3/7/1-6: Codes for Object types

A detailed description is to be defined.

4. Interworking Standard for Polling Information

4.1 General Structure of Polling Byte

bit 7							bit 1
sys							

Bit 7: System code flag 0 = application
1 = system

4.2 System Codes for Polling Information

bit 7							bit 1
1							

FE_{HEX} BAU not responding, set by polling master

FF_{HEX} Application not running

4.3 General Polling Interworking Standard for Applications

bit 7							bit 1
0	E	E					

Bit 5, 6 Application Alarm Class

00	no alarm	
01	Application warning	
10	Application detected error	(i.e. application program running, but device not working properly; e.g. sensor defect)
11	Application Alarm	the properly working device has detected an alarm (e.g. fire alarm)

Bit 4..0 This bits are application specific standard. These bits are to be set to 0 until further polling formats are defined.