EIBA Handbook Series

Release 3.0

Volume 1:
Primer

Part 2: Introduction to the System

19.04.1999

Table of Contents

Page left blank intentionally.

1. EIB: User's View

1.1 Introduction to the EIB Technology

1.1.1 General Purpose of EIB

The European Installation Bus (hereafter referred to as "the Installation Bus" or in short as "the Bus") is designed as a management system in the field of electrical installation for load switching, environmental control and security, for different types of buildings. The Installation Bus can be installed in large buildings such as business premises, schools, hospitals, factories and administration premises as well as in domestic residences. Its purpose is to ensure the monitoring and control of functions and processes such as lighting, window blinds, heating, ventilation, air-conditioning, load management, signaling, monitoring and alarms.

The EIB system allows the bus devices to draw their power supply from the communication medium, like Twisted Pair or Powerline (230 V mains). Other devices may, additionally, require power supply from the mains or other sources, as in the Radio Frequency and Infra-red media. Fig. 1/2-1 draws some usage examples.

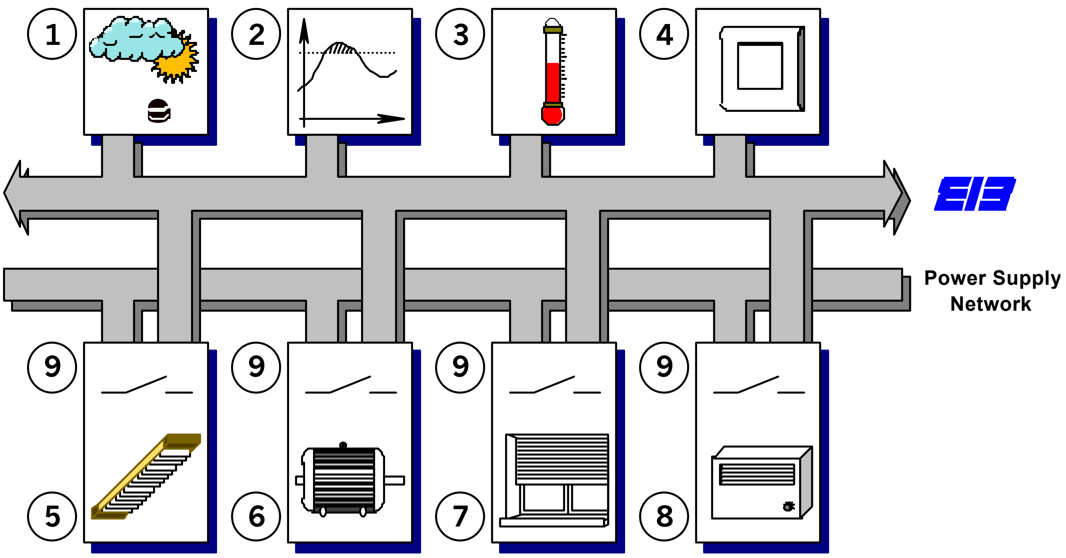


Fig. 1/2-1: Bus and mains networks

① brightness sensor	② threshold detection	③ sensor
④ monitoring	⑤ lighting	⑥ motor control
⑦ window blinds	⑧ heating	⑨ 230V switching

1.1.2 Main Features of the Bus

The Installation Bus is designed to provide distributed technical control for management and surveillance of buildings. Therefore it provides a serial data transmission between the devices connected to the bus. It also operates as a compatible, flexible low-cost system supporting the above applications.

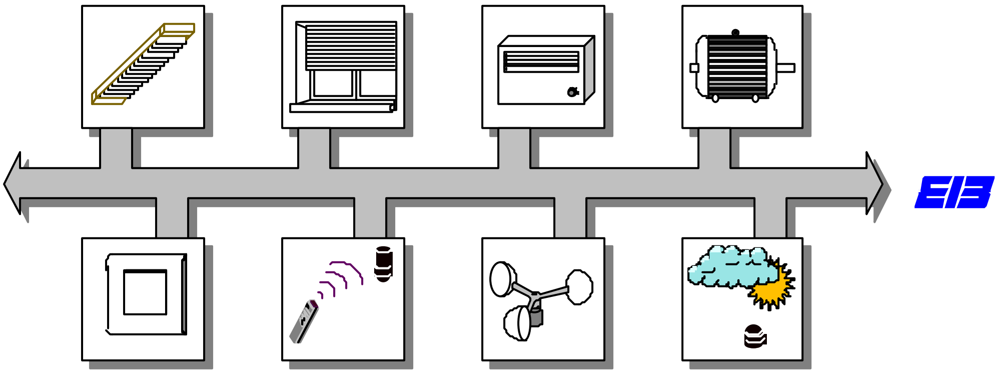


Fig. 1/2-2: Decentralized BUS organization

The Bus system is usually implemented as a decentralized system (see Fig. 1/2-2) but nevertheless it still allows, whenever it is required, centralized application implementations. Decentralized management is implemented within the devices whether they are transmitters or receivers, they communicate directly to each other without recourse to hierarchy or network supervisory device. This type of management makes the system highly flexible.

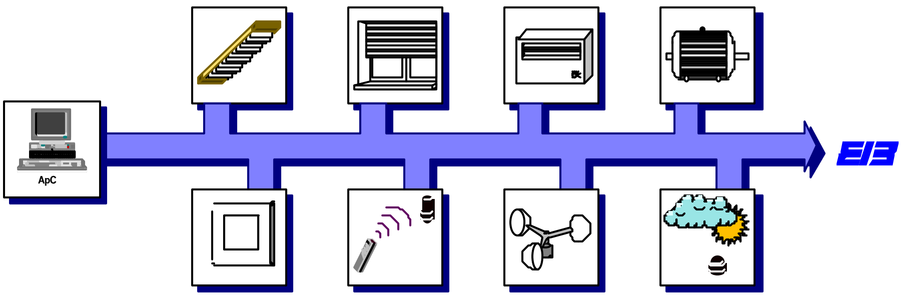


Fig. 1/2-3: Centralized BUS Organization

The application nevertheless allows for a centralized management mode (see Fig. 1/2-3). An Application Controller (ApC) may be positioned on anywhere on the bus.

1.2 The Media

The EIB protocol is today supported by several media, like Twisted Pair, Powerline, Radio Frequency and Infra-Red. It is of course always possible to connect gateways to other media.

1.2.1 Physical Topology

The physical topology can be regarded as the descriptions of the paths over which the communication signals can be transported.

In some media the physical transmission is not bound to any electrical signal carrier.

Example: Radio Frequency, Infra-Red

In other media, the data signals follow the paths of some electrical wiring.

Example: Twisted Pair, Powerline, Optical Fiber

1.2.2 Twisted Pair

The electrical segments can have an arbitrary topology (i.e. linear, star, tree, loop or combinations of them) consisting of individual wiring sections as long as the electrical requirements (resistive and capacitive length) are not exceeded. Examples of such topologies of electrical segments are shown in Fig. 1/2-4.

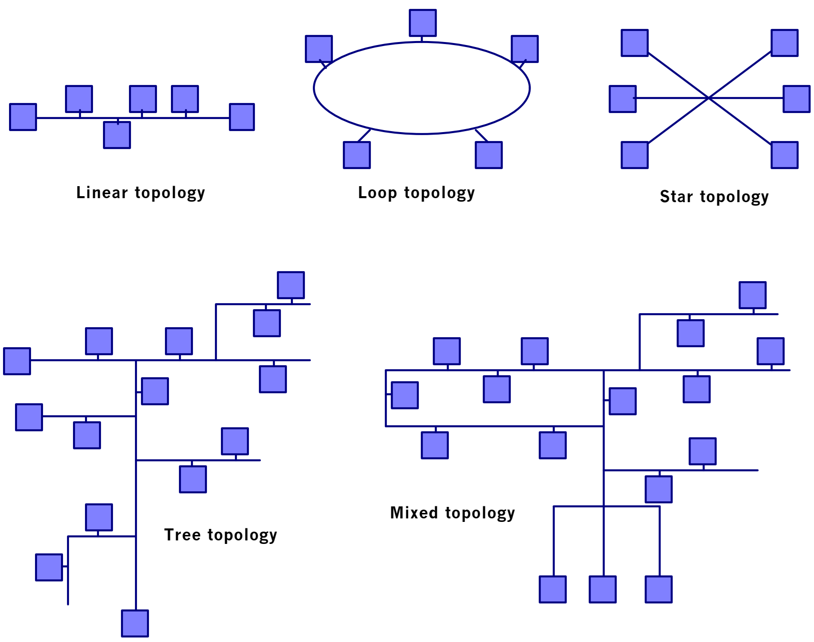


Fig. 1/2-4: Electrical Segment Topology

Terminating resistors are not required.

1.2.2.1 Twisted Pair Installation Topology

The bus needs 1 twisted pair to work. When standard control cables with 2 pairs are used, one is dedicated to signal transmission, the second one may, for instance, be used for complementary power feeding services. But in one zone or one building, it shall be used in one way only.

Installation topology is similar to the installation of the mains power supply distribution (see Fig. 1/2-5). It is adapted to a home or a building structure by using a tree topology. But other physical (cable) topologies are also possible.

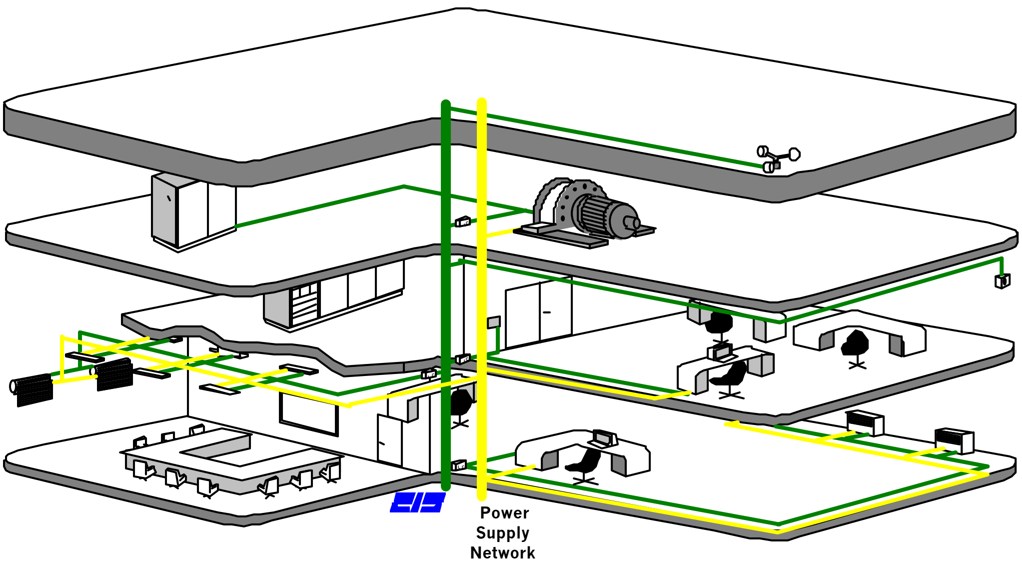


Fig. 1/2-5: Bus Cabling

1.2.2.2 Twisted Pair Electrical Segments

- Up to **64** bus devices may then be connected to each of these lines, allowing a total of 64.000 components to be connected.
- The total cable length shall not exceed **1000 m** per electrical segment.
- The maximum length allowed is **700 m** between two devices and **350m** between a power supply unit and a device.

1.2.2.3 Twisted Pair Logical Segments

In certain cases the connection of more than 64 devices to the same line may be required. The system allows two segments to be connected via a bridge, mostly named "repeater", (see Fig. 1/2-6). The connection capacity of the line may thus be doubled. In principle, a line may include up to 4 electrical segments connected together via repeaters, thus taking the capacity of the line to 256 devices. However, more than one electrical segment shall only be used for extension of existing installation but not for a new (initial) installation. A maximum of 6 Line Controllers (i.e. Line Couplers, Backbone Couplers and Repeaters) are allowed in one transmission path.

The logical segments themselves are connected together by line couplers (LC) via a single logical segment. A maximum of 16 logical segments is allowed.

Up to 15 zones can be federated by using the Bus itself. This can be also achieved by higher-level bus systems like ISDN or Profibus, requiring appropriated gateways.

1.2.2.4 EIB Twisted Pair Bus Devices

The maximum number of devices that can be connected to the bus without any repeater is 13105, when using only 12 lines, and 16129 if using the full address range of 15 lines. When repeaters are used, these numbers become 49201 and 61249 respectively.

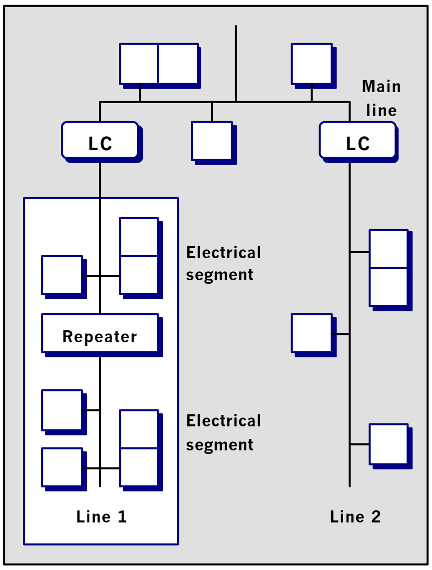


Fig. 1/2-6: Bus Extension

1.2.2.5 Transmission Characteristics

- Transmission: balanced, baseband, asynchronous.
- Transmission speed rate: 9600 bps.
- The collision avoidance algorithm: CSMA/CA

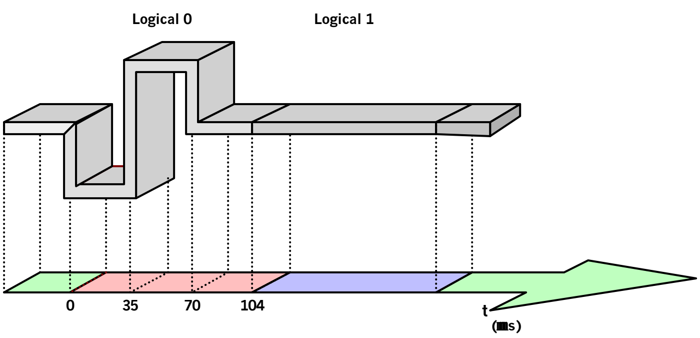


Fig. 1/2-7: Bit coding

1.3 Data Exchange & Interworking

1.3.1 Data Exchange

The communication between a sensor (e.g. a switch) and an actuator (e.g. a lamp) is a sequence of operations (see Fig. 1/2-8). In our case, using the EIB protocol, a switch - being uniquely defined by its physical address - can communicate to lamps using group addressing.

The group addressing is based on the exchange of data coded with common rules between communication objects (mailboxes). A communication object is only able to transmit telegram on a single group address. On the opposite side, a communication object can be subscriber to several group addresses, allowing it to receive telegram from different emitters

That means all EIB Bus devices subscribers to the right group address (i.e. our lamp) will receive the command message from the switch.

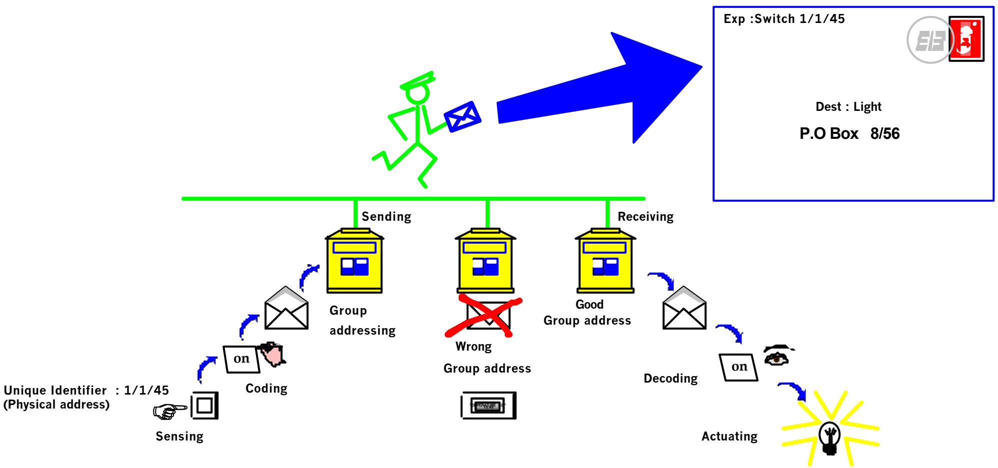


Fig. 1/2-8: Communication Synopsis

1.3.2 Interworking

The EIB protocol does not communicate to communicate. The aim of communication is the interworking between sensors and actuators.

The interworking pyramid, shown on Fig. 1/2-9, defines the different interworking degrees. It starts with the data format used and ends with the application functionality. It can be compared to a mail exchange where the communication object is the mailbox and the functionality the complete written order.

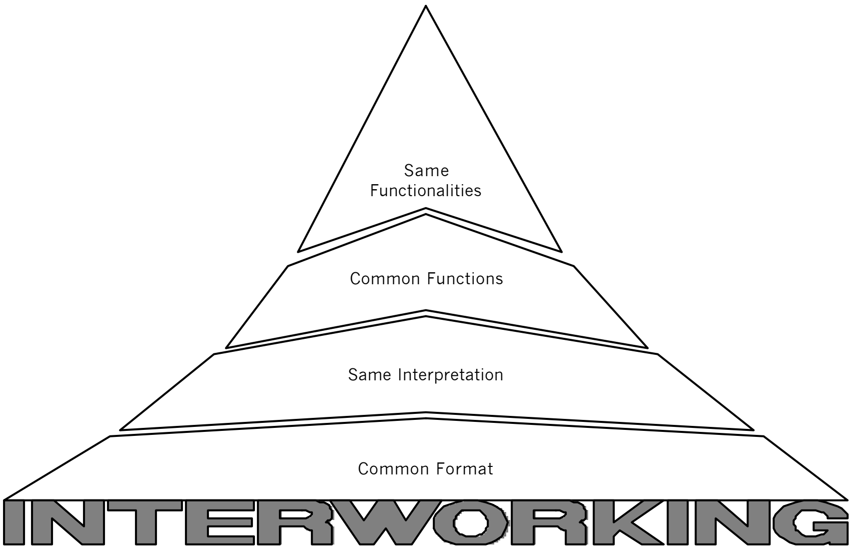


Fig. 1/2-9: Interworking Pyramid

- Minimum is to exchange data with a common format (same envelope). This is the basis of the communication, but nevertheless it doesn't guarantee at all the interworking.
- Minimum of interworking is guaranteed if exchanged variables are understood with the same interpretation by other products (same vocabulary).
- A step further is share common functions to keep compatibility in Input/Output data (same grammar rules).
- The top of interworking is achieved by sharing common functionality's (same phraseology) to allow you to have the same well-defined behavior for different products.

The **E**IB **I**nterworking **S**tandards (**EIS**) answer the problem and assume consistency between application processing.

1.4 The EIB Protocol

1.4.1 Structure of Data Packet

The information exchange between two devices is achieved by transmission of data packets. Each data packet must be acknowledged. For every medium, the message frame looks similar like in Fig. 1/2-10.

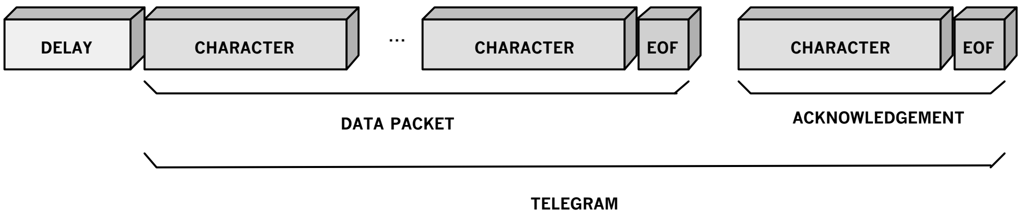


Fig. 1/2-10: Message frame

Some media will precede or follow this message by some medium specific sequences, characteristic for its medium access control or error correction mechanisms.

The data packet (see Fig. 1/2-11) contains the following fields:

- control field
- source address field
- destination address field
- length
- LSDU (Link Service Data Unit) -i.e. info to be transferred-
- check byte

In the case for example of a failure detection message or any other urgent message, the EIB system allows a transmission priority to be assigned to the transmission of the data packets. Alarm messages may have priority over all other messages sent in normal operation mode. Retransmitted data packets have also higher priority than normal packets.

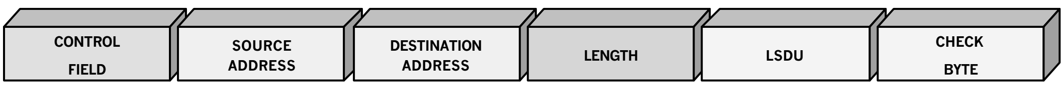


Fig. 1/2-11: Data Packet Fields

1.4.2 Addressing Mode

Management of EIB Bus devices connected to the Installation Bus can be addressed using two modes:

1) physical addressing (⇨ system operation)

2) group addressing (⇨ normal operation).

Every bus device is identified by a unique physical address, e.g. as described in Fig. 1/2-13. Two EIB Bus devices should not have the same physical address. The physical address consists of a zone, line and EIB Bus device number; it corresponds to the device as a whole. The source address field always contains the physical address. The physical address is only used as destination address for initialization, programming and diagnostic operations (connection oriented transmission). This corresponds to a system access mode.

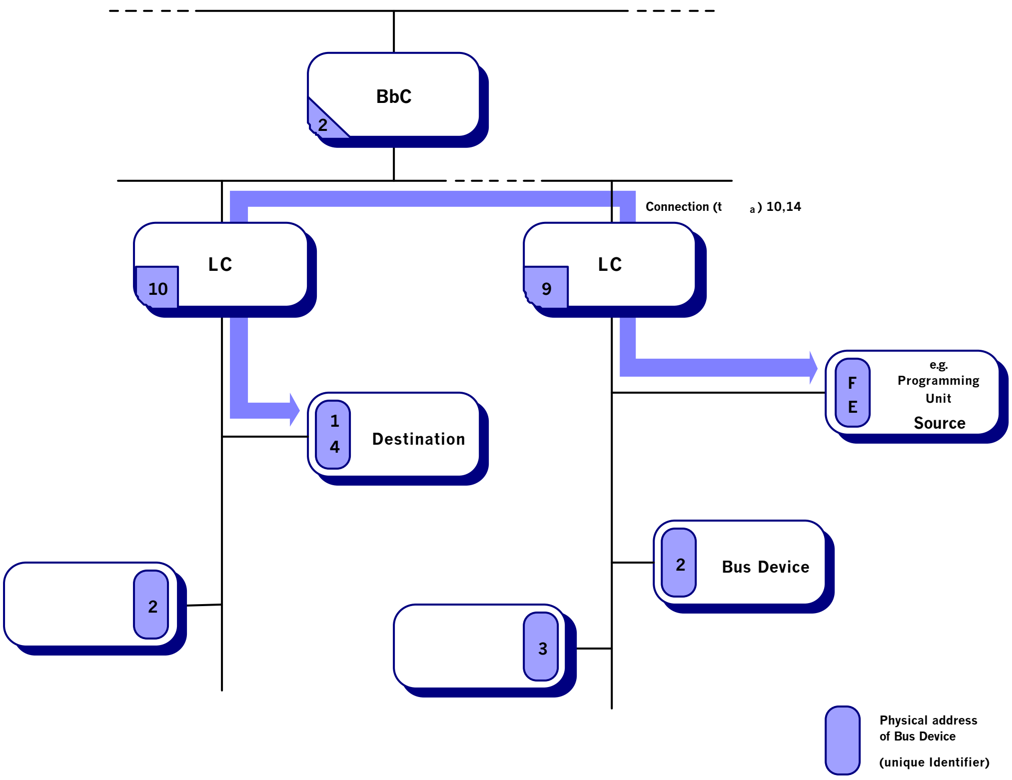


Fig. 1/2-12: Connection oriented Transmission (Physical Addressing)

Example for address used by data packet sent by the programming unit:

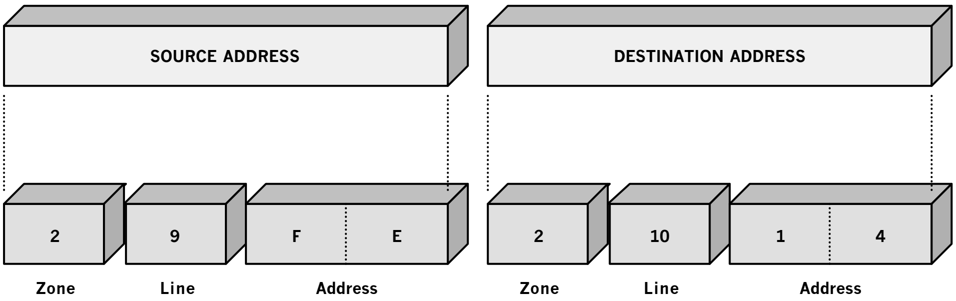


Fig. 1/2-13: Address Field for Physical Addressing

Group addressing (see Fig. 1/2-14) corresponds to the normal operation mode. Functions of EIB Bus devices belonging to the same group, may be controlled by only one message sent by a "source" EIB Bus device. Functions however, may belong to several groups and may be activated independently by every EIB Bus device of the group.

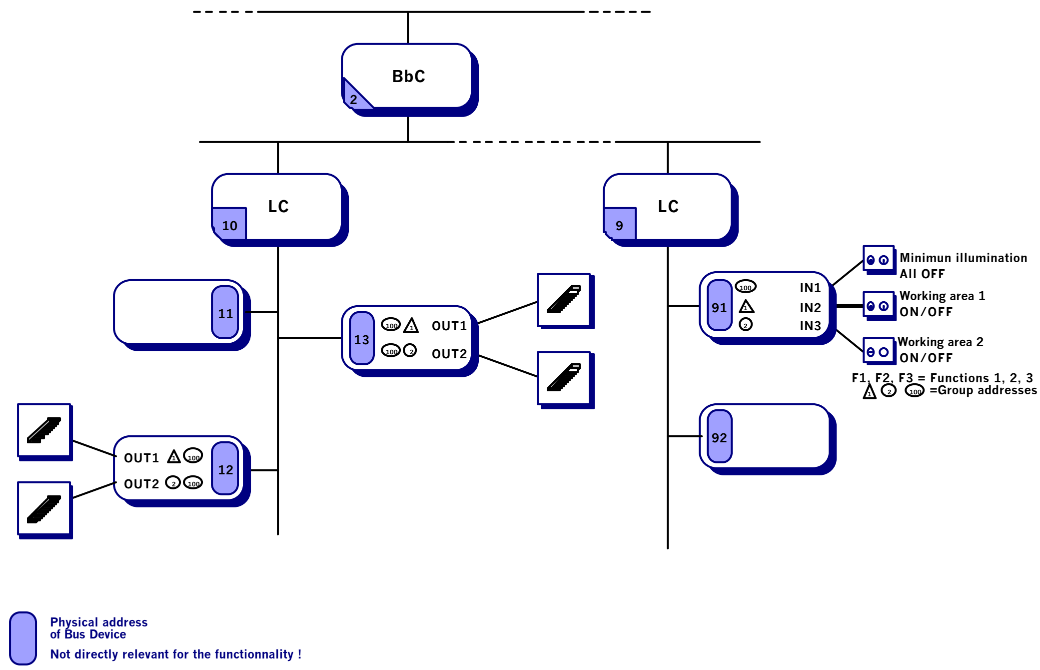


Fig. 1/2-14: Group Addressing

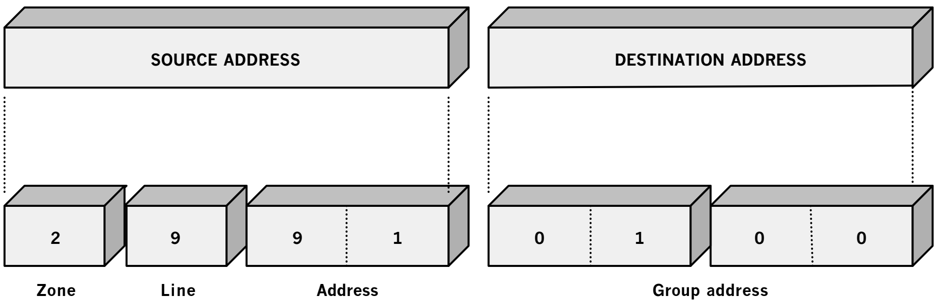


Fig. 1/2-15: Address Field for Group Addressing

The group addressing is a logical link between bus devices. A sensor can only transmit on one group address and an actuator can receive several. The group addressing gives flexibility by means it allows to add a bus device in a very simple way, just by connecting it to the correct group address.

1.5 The EIB Components

EIB devices are divided into three types according to their use:

- basic components, such as power supply unit (PSU), choke, signal filter...
- system components, which support the basic operation of the system such as Bus Coupling Unit (BCU), Line Coupler (LC), Phase Coupler, Repeater, ...
- EIB devices which are dedicated to applications such as sensors, actuators, IR-decoders, display panels. These types of devices are connected to EIB by a Bus Coupling Unit or similar interface.

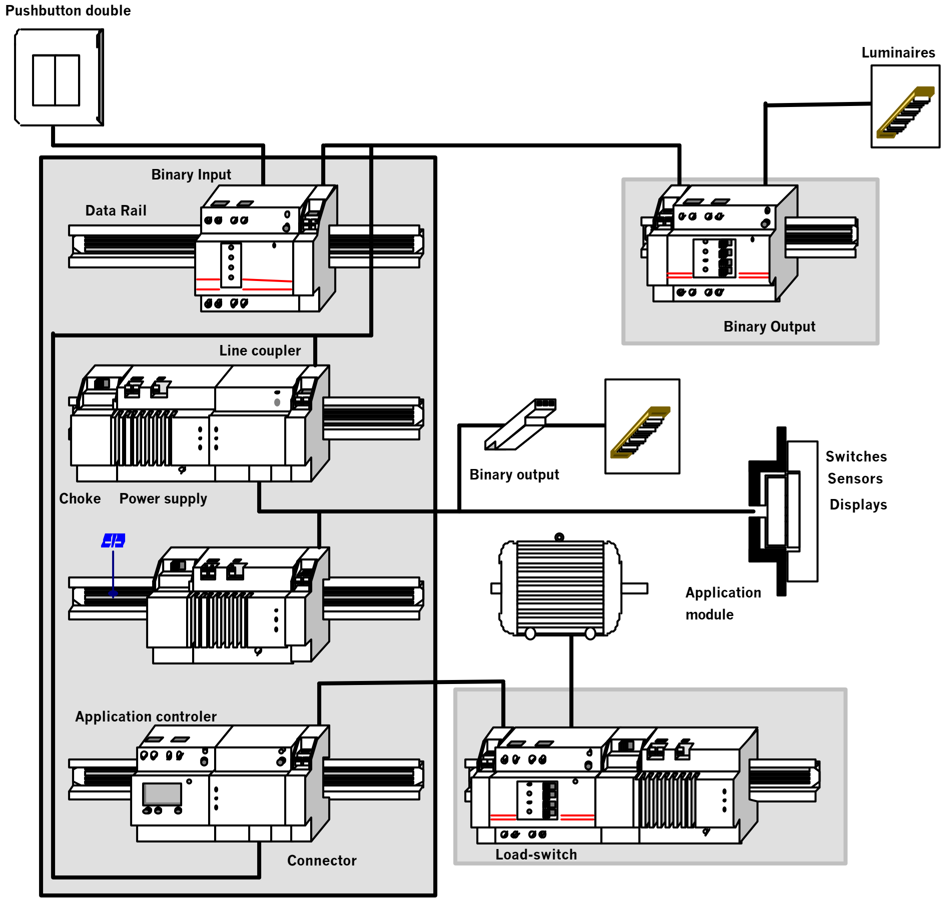


Fig. 1/2-16: Installation Example

1.5.1 Basic Components Definition

Power Supply Unit (PSU): Provides power for feeding of EIB Bus devices (Safety Extra Low Voltage (SELV), 30 V DC nominal).

Choke: Provides the coupling of the Power Supply Unit to the data bus line.

Data rail: Mounted support with four tracks to distribute the bus onto DIN rail.

Data rail connector: Provides the connection between the bus cable and the data rail.

1.5.2 System Components Definition

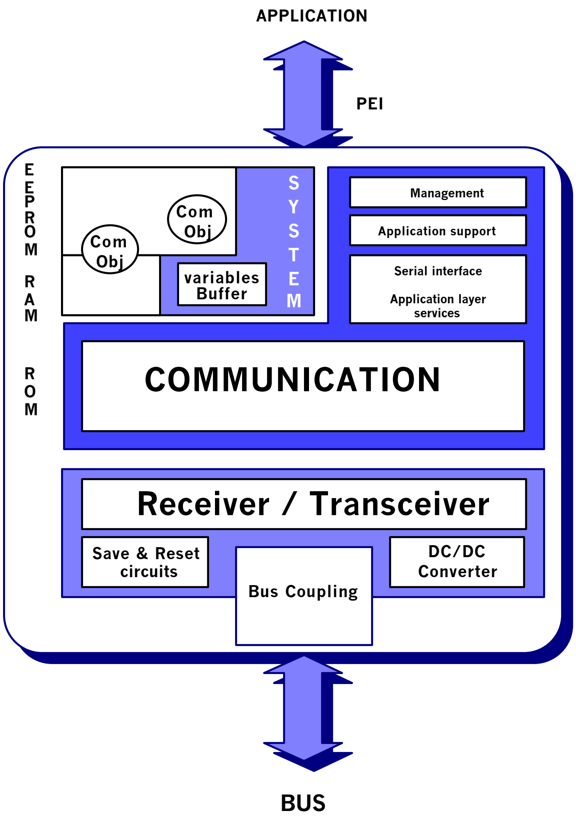


Fig. 1/2-17: Bus Coupling Unit Schematic

Bus Coupling Unit (BCU): The BCU is available as an EIB standard product or the functionality may be integrated directly in the product. As it is shown on Fig. 1/2-17 the BCU is composed of:

- a Transceiver :

  A module which provides:

  - bus coupling of the device by sending data signals onto the medium and decoding received signals
  - optional a DC power conversion for the BCU, for the receiver and transmitter function, for output signal generation,...

- a Communication Controller:

  A microprocessor which provides:

  - necessary communication features
  - optimal application routing
  - Physical External Interface (PEI) support
  - Operating system
  - Space for a program.

The Line Coupler (LC): The Line Coupler is a system components for Twisted Pair. It uses the same basic functions as the repeater, but it connects the line to which it belongs to the main line. The line coupler ensures data packets routing and buffering with overflow management going from a line to the main line and vice versa. The power feeding of the LC comes from the line it belongs to. The LC provides galvanic separation.

The Repeater: The repeater as well is a Twisted Pair system component. Its function is to regenerate the electrical network signals and to separate the bus access. The repeater makes it possible to connect electrical segments together and create large extended lines without back feeding effects from one electrical segment to the other. By use of repeaters more than 64 devices per line can physically be connected and more than 1000m overall wire length is allowed.

1.5.3 EIB Bus Device Definition

EIB Bus devices are generally built up from two parts: the Bus Coupling Unit and the application module. The Bus Coupling Unit is a decentralized bus manager in each device and provides electrical features as well as data coupling to the Bus, in order to allow the separation of application hardware and software from the Bus communication system.

From the installation point of view the EIB Bus devices can be divided into 4 groups:

1. *Rail-mounted* EIB Bus devices (for applications like load switching, analog input, binary input, IR-decoder) for controlling of appliances such as brightness sensors, anemometers, humidity sensors, temperature sensors, ...
2. *Flush mounted*: where the Bus Coupling Unit is mounted *in* the wall for monitoring purposes and the application module snapped onto it outside the wall (like push-button, sensor, IR-decoder, set-point control, display panel, ..).
3. *Surface mounted* EIB Bus devices: where bith the Bus Coupling Unit and the application module are mounted *outside* the wall.
4. *Device mounted*: for incorporation into appliances such as heaters, lamps, etc. ...

1.5.4 Gateways to External Networks

An Installation Bus System may be connected via gateways to external networks. This connection may be done either at the backbone, the main line itself or the any other line. Some connection (telephone, radio, power line, etc. ...) must be in conformance with the relevant national regulations.

The gateways connect the Installation Bus network to networks where the communication layers are different from the Bus network, especially when there is an address translation function built in which depends on the other network. Application specific functions may be implemented in the gateways as well.

Examples for such gateways are:

- data over voice
- analog phone connection
- ISDN connection
- field bus connection
- connection to mainframe computer

1.6 The EIB Installation Software

The EIB Tool Software - ETS - runs under Windows™ 3.1 (or higher), it allows the user to fully prepare and configure projects for EIB installations using technical descriptions of certified EIB products (see Certification section), supplied by the EIB Manufacturers.

With the Commissioning&Test module, the user can download the prepared project into the system, and perform some diagnostics checks. It may be installed e.g. on a laptop PC.

1.7 An EIB-Based Lighting Control System

This chapter presents an example of how to design a control-command network and how EIB technology easily answers the problem.

1. Project Definition

A simple lighting control system for an office building illustrates how an EIB-based system can be used. Unlike centralized systems, an EIB-based system can be installed cost effectively one floor at a time. This example involves the two major aspects of the EIB technology: the project design & configuration and the on-site installation.

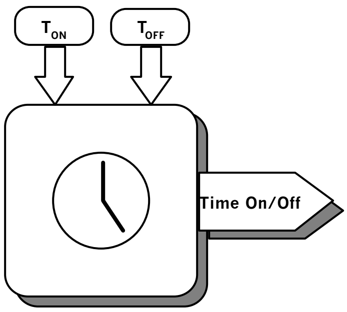
A very simple functional scenario could be as follows:

Between 7:30 AM and 6:00 PM the time-switch permits lighting to be on.

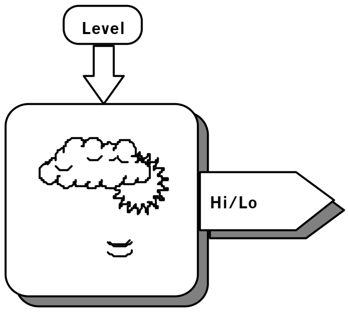
In each office and in the corridor photocells in combination with detectors command lighting according to the daylight level if employees are inside the room.

To answer the problem 4 products will be used:

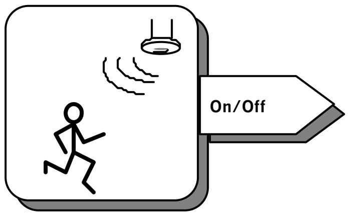
A **time switch** to send a bi-state message using the EIB Interworking Standard EIS 1 (1 bit variable). "Time On" allows lighting; "Time Off" turns off all luminaries. The T_{ON} en T_{OFF} values are set by using parameters.



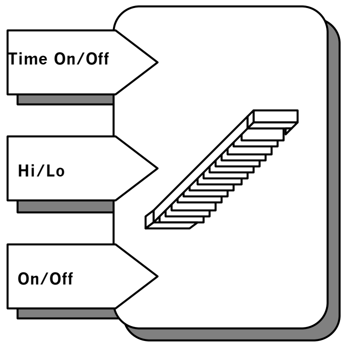
A **light sensor** to evaluate the outside light level. It sends a Hi/Lo message using the EIB Interworking Standard EIS 1 (1 bit variable). The light level is set by using parameters.



A **presence detector** to detect employees. It sends an On/Off message using the EIB Interworking Standard EIS 1 (1 bit variable).



A **luminaire controller** receiving commands and functional conditions from the different sensors.



2. Addressing

After the various products have been selected from the EIB product database of ETS, the next step is to define the connections between them. The EIB products are linked via Communication Objects (ComObj). As the ComObj is defined as a "transmitter of variable" it is characterized by the format and the type of the variable (EIS type) it has to transmit.

The connection is done by using a group address. Once the group address is assigned to a ComObj, the group address is associated to the variable type. Then, ETS is able to verify every time this group address is to be connected to a new ComObj, if the EIS type matches.

By using the graphical representation of ETS, the planner just has to select a group address and connect it to all the ComObj he wants to link all together.

Bus device		**Connection**					
		Time On/Off	**Light level**	**Presence Office N°1**	**Presence Office N° 2**	**Presence Office N° 3**	**Presence orridor**
Name	**ComObj**	**1/1**	**1/2**	**2/1**	**3/1**	**4/1**	**5/1**
Time switch	Time On/Off	S					
Light sensor	Hi/Lo		S				
Presence detector N°1	On/Off			S			
Presence detector N°2	On/Off				S		
Presence detector N°3	On/Off					S	
Presence detector corridor	On/Off						S
Luminaire controller N° 1	Time On/Off	R					
	Hi/Lo		R				
	On/Off			R			
Luminaire controller N° 2	Time On/Off	R					
	Hi/Lo		R				
	On/Off				R		
Luminaire controller N° 3	Time On/Off	R					
	Hi/Lo		R				
	On/Off					R	
Luminaire controller corridor	Time On/Off	R					
	Hi/Lo		R				
	On/Off						R

S: Sender R: Receiver

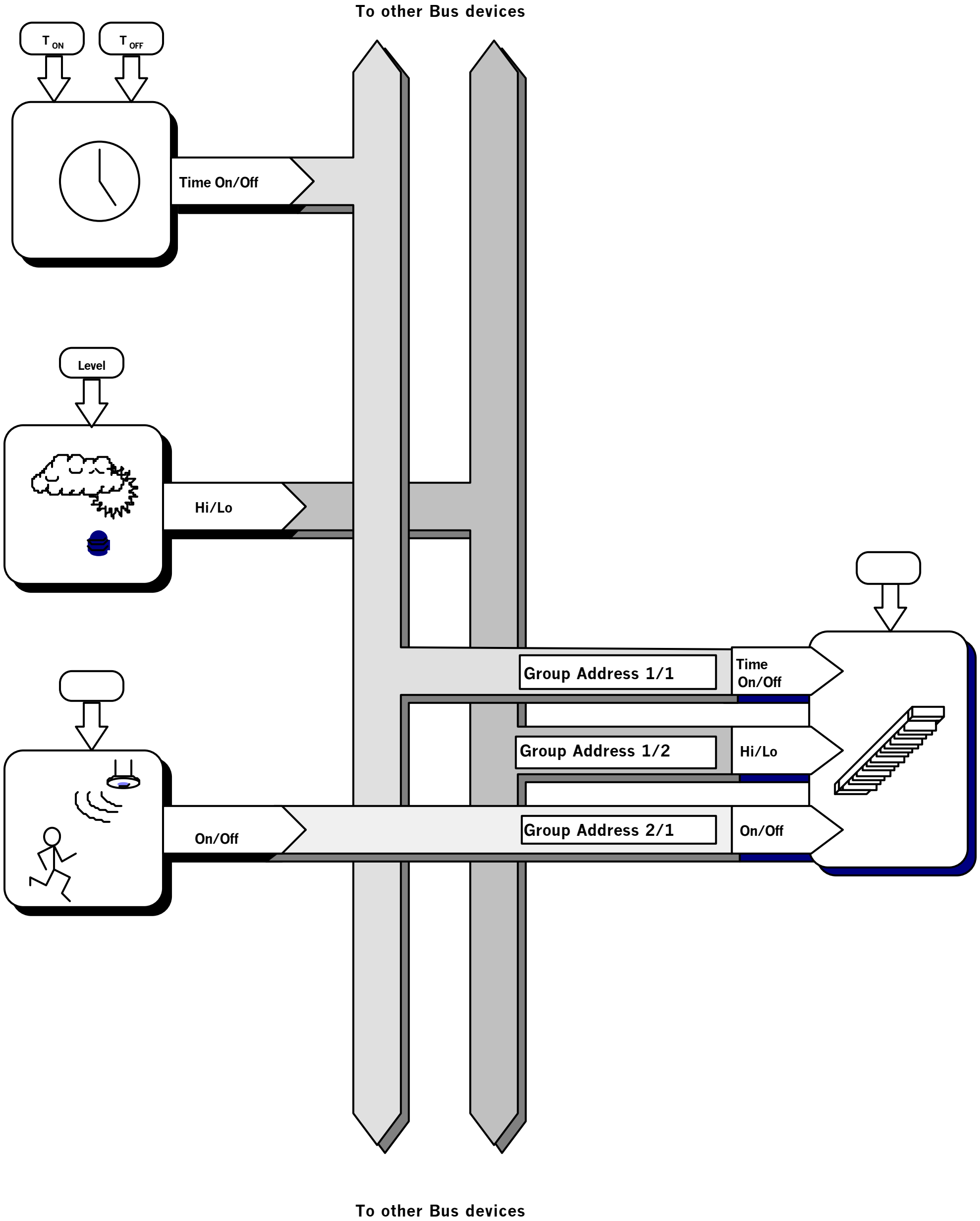


Fig. 1/2-18: Office N°2 Link Mapping

3. Programming

Once the addressing is done, the planner is able to set the parameters of each application. In the design section of ETS, he opens the parameter window of the application selected and sets the parameters to the right value.

For the time switch, he sets T_{ON} parameter to 7:30 value and the en T_{OFF} parameter to the 18:00 value.

For the light sensor, he selects the 2000 lux level from a table of possible values: (500-1000-2000-5000-10000-15000-20000).

The application programming with the design section of ETS is complete and the contractor has to download programmed applications to the EIB Bus devices using ETS download facility.

4. Downloading

The first step in downloading is the physical address setting. On ETS the contractor accesses the physical address loading function. He selects the EIB Bus device application and presses the service button on the BCU of the EIB Bus device to be configured. The physical address is automatically downloaded by ETS. This procedure must be carried out for all EIB Bus devices on the system.

When all of the EIB Bus devices have been configured, the contractor, via the application downloading function automatically installs, in one run, all of the applications into the EIB Bus devices

The EIB installation is now ready to work.

2. EIB: Developer's View

2.1 Preface

EIB concentrates unequivocally on home and/or building management. This focus permits it to deal with all tasks and challenges within this domain thoroughly and efficiently.

The European Installation Bus (EIB) is an open, comprehensive system which covers all aspects of Building Automation. It is managed by the neutral EIB Association.

Though standardized *Bus Access Unit* (BAU) building blocks are available from several vendors, EIB is in the 0^{th} approximation a *specification*, not an implementation (like a chip or a transceiver). This means EIB is *open*: EIB may be implemented by anyone, on any chip or processor platform chosen - both as proprietary implementation for individual products, as well as for OEM BAU’s. Conformity tests are defined, and EIB Certification is open to all members of the Association.

Why “in 0^{th} approximation”? Because EIB embeds the protocol in an encompassing Home and Building Electronics *System*, with standardized system components (such as the BAU’s), network management and interworking standards, with a vendor-neutral tools and programming interfaces for PC’s, training for electrical contractors, certifications schemes etc.

2.2 Network Topology

EIB is a fully peer-to-peer network, which accommodates up to 65’536 devices. The logical topology allows 256 devices on one *line*. As shown in Fig. 1/2-19, 15 lines may be grouped together with a *main line* into an *area*. An entire domain is formed by 15 areas together with a *backbone line*. On open media, nearby domains are logically separated with a 16-bit SystemID. Without the addresses reserved for couplers, (255 x 16) x 15 + 255 = 61’455 end devices may be joined by an EIB network. Installation restrictions may depend on implementation (medium, transceiver types, power supply capacity) and environmental (electromagnetic noise, …) factors. Installation and product guidelines should be taken into account.

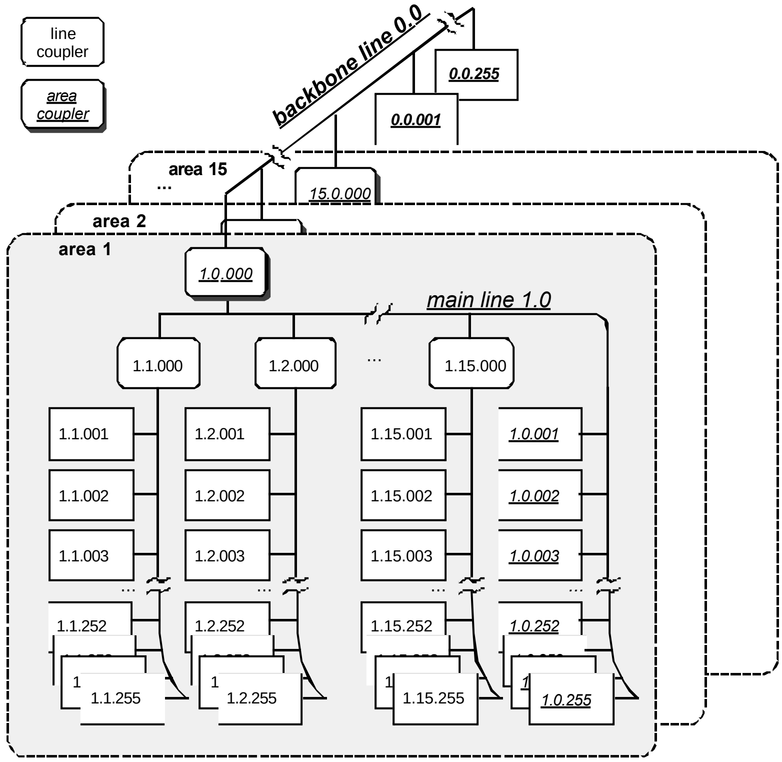


Fig. 1/2-19 The Logical Topology of EIB

Couplers connect lines or segments, e.g. within the Twisted Pair (TP) medium, or different media; their functionality may be (some combination of) repeater, bridge, router, package filter (for traffic optimization), firewall protection etc. EIB defines various standard coupler profiles.

2.3 Media

As indicated, EIB Medium Access Control is highly optimized for each medium individually. Available implementations further optimize for a combination of transceiver performance and cost. EIB.IR (*Infrared)* is currently being developed.

2.3.1 EIB.*TP* - Twisted Pair

On EIB TP (Twisted Pair), bit-level collision detection with dominant logical 0 ensures that in case of collision, the transmission always succeeds for one of the communication partners. The resulting elimination of re-transmissions further enhances the performance of EIB TP. Together with EIB's powerful group addressing, EIB TP1 Collision Avoidance caters for extreme efficiency with reaction times 100 ms for two simultaneous transmissions. Fast polling allows up to 14 devices to be polled for 1 byte status-information within 50 ms. A physical TP segment may be up to 1 000 m long.

2.3.2 EIB.*PL* - Powerline

EIB PL (Powerline) uses a novel Spread Frequency Shift Keying modulation technique. With a corresponding numerical matched filter, the available BAU's guarantee adequate communication for group addressing to work reliably on PL. Medium access is controlled via a preamble sequence, with randomized re-transmission slots.

Maximum distance between 2 devices (without repeater): 600 m. (Communication is influenced by electromagnetic pollution conditions in the installation.)

2.3.3 EIB.*RF* - Radio Frequency

EIB RF (Radio Frequency) lines are physically separated by a different carrier frequency. In free field conditions, the transmission distance is about 300 m. Retransmission ensures that large volumes can also be covered inside the building. Retransmitter functionality is optimally distributed among the installed devices by the system itself.

2.3.4 EIB.*net* - Automation Networking

The EIB.*net* specification realizes EIB on all media with a logical link layer according to ISO/IEC 802-2, including Ethernet and Arcnet. Not limited to high-speed backbones, EIB.*net* also allows management or automation level devices to be directly connected.

An enhanced specification catering for routing based on the Internet Protocol (IP) is being reviewed. In this way, EIB.net allows transparent usage of existing LAN infrastructure, and is intrinsically Internet and Intranet enabled.

2.4 The EIB OSI Communication Protocol

Fig. 1/2-20 shows how the EIB communication stack is structured according to the OSI 7-layer model. This is also reflected in the frame structure shown in Fig. 1/2-21. The physical layer and link layer obviously depend on the characteristics of the physical medium. For medium access control, EIB prescribes Carrier Sense Multiple Access (CSMA) with Optimized Collision Avoidance. The precise mechanism may be highly optimized for the particular medium. The Destination Address Flag (DAF) distinguishes between Group and Device Oriented telegrams.

Through the Network Protocol Control Information (NPCI), the Network Layer controls the hop count; for devices other than routers or bridges, it is trivial. The Transport Layer manages logical communication relationships, which can be:

1. one-to-many connectionless (“group” multicast)
2. one-to-all connectionless (broadcast)
3. one-to-one connectionless
4. one-to-one connection-oriented

It provides the mapping between addresses and an abstract internal representation, the Communication_Reference_ID (cr_id).

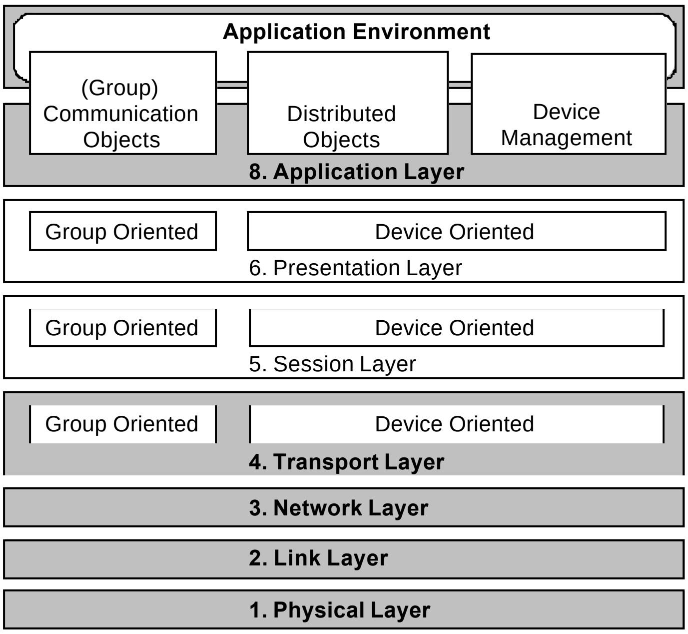


Fig. 1/2-20 EIB Communication according to the OSI Reference Model

All services are mapped transparently across Session and Presentation Layer, which are reserved. Application Layer implements the API for client/server management of EIB networks (see paragraph 2.5 "EIB Network Management and Addressing"). The Group Application Layer deals with the assignment of a group cr_id to a local instance of a Group Communication Object (or shared variable), for receiving (one-to-n) and for sending (one-to-one). For convenience, both Group Communication Objects and Distributed Objects are encapsulated by the EIB User Layer, which takes over the Application Layer nitty-gritty from the application. The User Layer acts also as default management server application.

<table>
<tr><td>octet 0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td colspan="2">6</td><td colspan="2">7</td><td>8</td><td>..</td><td>N - 1</td><td>N ≤ 22</td></tr>
<tr><td>Control Field</td><td colspan="2">Source Address</td><td colspan="2">Destination Address</td><td>DAF;
NPCI;
length</td><td>TP CI</td><td colspan="2">AP CI</td><td>data /AP CI</td><td colspan="3">data</td><td>Check Octet</td></tr>
</table>

Fig. 1/2-21 EIB PDU frame structure (long frames allow N < 255).

EIB Protocol Data Unit (PDU) frames may carry application data formats of up to 14 bytes (extension to 230 is currently under consideration).

In the next section, we will discover the central importance of the dedicated group-oriented facilities of the EIB Operating System.

2.5 EIB Network Management and Addressing

2.5.1 Network Management

To manage network resources (e.g. when configuring an installation), EIB uses a combination of broadcast and point-to-point communication.

Via broadcast (optionally using a device's unique serial number), each device in the installation is assigned a unique Physical Address, which is used from then on for further point-to-point communication.

- A connection (optionally with access authorization) may be built up, for example to download the complete 'applet' binary image of an application program.
- Connectionless access is possible to EIB Distributed Objects through *‹device›.‹object›.‹property›* addressing, as a native EIB management-level mechanism for status visualization and control.
- A dedicated master-slave *fast polling* mode ensures live- and status check of critical subsystems.

2.5.2 Group Addressing for Run-time Efficiency

EIB supports full multicast ("group") addressing. *Full* means that:

1. EIB is not limited to grouping devices: each device may publish several variables (known as "(Group) Communication Objects") individually, which can be grouped independently from one another into network-wide shared variables. As a bonus, properties of Distributed Objects may be published as shared variables as well.
2. As explained above in the description of the group-oriented EIB communication stack, a shared variable can be fully read/write bi-directional. In this way, all devices can also send unsolicited multicast frames.
3. EIB makes a 16 bit address space available for these shared variables. Even with the limitation of some implementations to 15 bits, this signifies that one installation may have up to 32k shared variables (or "group addresses"), each with any number of local instances.

The resulting scope and efficiency makes group address communication the preferred "runtime" mode for autonomous EIB field level communication.

In this maybe slightly unexpected way, EIB goes some distance towards reducing the need for redundant automation hierarchy levels (and bandwidth!) through appropriate addressing and device modeling schemes.

2.5.3 Multi-client / Multi-server Management of the OO EIB Network

An EIB installation may be seen as a collection of distributed resources, which can be managed across the network. To this end, each EIB device implements a server which provides control over local resources (including hosting services for external CPU or memory resources accessed via the serial Physical External Interface (PEI, see below). A series of APCI's render these services accessible to remote clients. Through the introduction of EIB Distributed Objects, the network resources actually become Object Oriented (OO).

Management clients typically access the network either for control or for (initial) configuration services. EIBA implements a complete suite of vendor-neutral, standard PC-based configuration tools, which manage loadable applets, as described in 8. Hand-held devices are also available. Network-based (typically DIN-rail mounted) clients permit Interactive Self-configuration (*Easy Installation*) of (sub)-systems.

2.6 Data Formats and Interworking

As we have seen, today's EIB frame may carry data formats of up to 14 bytes. The basic data formats specified by the EIB specification include:

- boolean (1bit)
- (un)signed short (16 bit)
- (un)signed long (32 bit)
- short float (16 bit)
- IEEE float (32 bit)
- date (24 bit)
- time (24 bit)
- control (4 bit) etc.

Identifiers are defined for nearly all physical values like temperature, length, speed, field strength, energy, power, etc.

Type information is used mainly at configuration time: it is not transmitted for better performance and to avoid imposing unnecessary restrictions on the combinations of devices.

Properties with these basic data types are grouped into 'Distributed' Objects, accessible via the network. The EIB Interworking Standards (EIS) specify various specialized objects for all area's of building automation such as lighting (dimming control, …), HVAC applications (room temperature control, boiler temperature control, …), time and event management (schedule handler, event handler, …).

2.7 Hosting and Interfacing Features of the EIB Operating System

Not only does the distributed EIB Operating System (OS) serve remote clients over the network. Of course it also puts its services as a communication server and management server at the disposal of *local* client applications.

2.7.1 Internal Applications

To an internal application, the BAU will moreover provide CPU and memory resources, timers etc.: the application "runs in the BAU". Advanced implementations allow up to three asynchronous application threads.

2.7.1.1 Utility Library API

As part of the User abstraction Layer, EIB standardizes a Utility (or User) Library API, which provides further infrastructure to the application. Included are user timers, debouncing, arithmetic, bit logic, message handling etc. Through the API, the application may also access external application hardware, as explained in the next section.

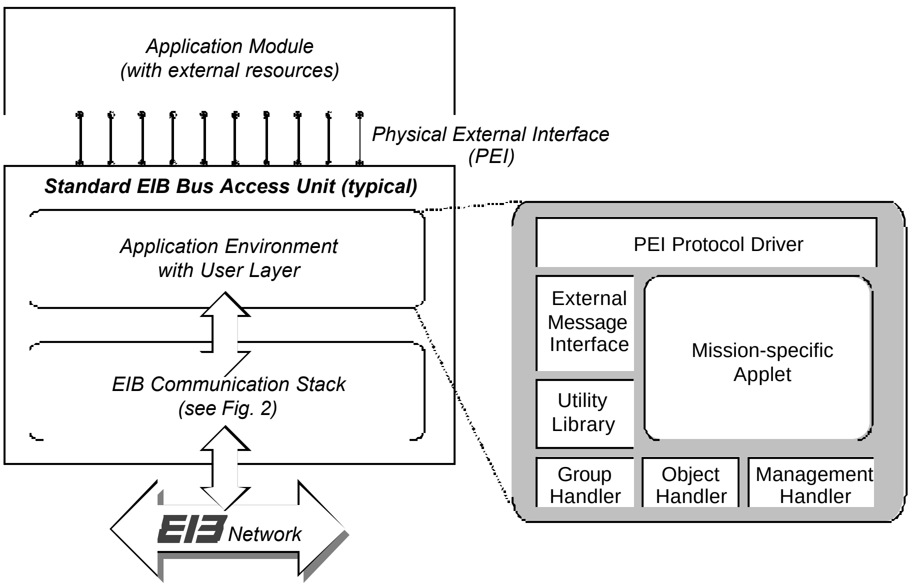


Fig. 1/2-22 PEI, User Layer and Hosting Features of typical EIB BAU

2.7.1.2 Hosting of External Application Hardware

Another unique feature of EIB streamlines the hosting features for application hardware or external resources: he standardized Physical External Interface (PEI). The PEI defines both electromechanical and software services for connecting an external application module to a BAU. Via a type resistor, the application module identifies its capabilities to the application; the BAU can deal with about 20 types (including binary, analog and serial I/O) and provides the application with corresponding services (via the API).

As an extra boon, the combination API / User Layer + PEI allows the combination loadable application + application module to be used *as is* on any physical medium. Particularly for flush-mounted devices, engineering overhead or adjustments to production are eliminated completely, and distribution logistics simplified accordingly.

Though optional, the PEI feature is exploited by certain standardized BAU’s, called Bus Coupling Units, (see below), which would not be possible without it.

2.7.1.3 Message Interface for Access to External Resources

For the serial PEI, EIB defines an *External Message Interface* (EMI). The EMI server allows both a local or remote client to access external CPU or memory resources.

2.7.2 Dual-processor Design, External Applications and Systems

Using the EMI the other way around, an external application can now utilize the local communication stack as EIB communication server. This permits dual-processor design of EIB devices. The EMI server may be implemented optionally by EIB Bus Access Units.

Actually, this feature can be used as a general mechanism for serial interfacing to external systems. Typical examples are the BCU-based EIB RS-232 interfaces, available from several manufacturers.

2.8 Tool Suites and Software Engineering Framework

EIB explicitly encompasses a methodology for Project Engineering, i.e. for linking a series of individual devices into a functioning installation. This is embodied in the two vendor-independent EIB Tool Software (ETS) suites for Windows:

1. ***ETS End-User's Edition***

   A project engineer or electrical contractor can import the Component Description into he ETS Project Edition. All device instances can be customized to the needs of the project and logically linked by assigning Group Addresses.

As a result, the effort is radically shifted away from project engineering.

2. ***ETS Developer's Edition (ETS+)***

   With the ETS Developer's Edition, the manufacturer encapsulates the remotely loadable applets in a series of abstract representations, which hide all implementation details. The resulting Component Description can be exported.

Both ETS-Version are built on top of a framework of software-engineering components for PC/Windows platforms, called the ***EIB Tool Environment*** (ETE). This set of API's forms part of the EIB standard. It is an architecture providing libraries, drivers and components for local- and remote access to the EIB media, the product- and project database and Graphical User Interfaces (GUI) . The ETE implementation is commercially available for 3rd party use, in various information technologies.

With the ***ETE Component Architecture*** (eteC), EIBA is completing an ambitious program for a powerful, network enabled 32-bit component infrastructure for its software tools.

The ***EIB Developer*** in the EIBA Member Company is additionally supported by:

- **Bus Monitor Program ("BusMon")**

  BusMon runs under DOS 4.0 or higher. This uses unformatted hexadecimal representations of BCU and telegram data contents, BusMon is a low-level tool for use by skilled engineers only.

- **Application Software Development**

  Third party solutions for writing application software are also available. Detailed list is available from EIBA.

The ***Certification Co-ordinator***, responsible for the certification process in an EIBA Member Company, is supported by integrated certification tools. These are available from EIBA and are intended for Interworking compliance testing (see Interworking- and Certification sections).

The application software is sent to EIBA in Brussels on a diskette. After testing, EIBA certifies the application software and returns them back to the manufacturer as certified database to be included in the certified database of ETS.

- **EIB Interoperability Test Tool ("EITT")**

  EITT runs under Windows 3.1 or later. It is intended for testing the conformity of application programs to the EIB Interworking Standards. It is a tool for experts and testing laboratories.

- **Bus Load Generator** ("BLG")

  EIB push button with BCU. The BLG cyclically sends a telegram on the bus. The time interval can be adjusted. (Only for testing new BAU's)

2.9 Other System Features

2.9.1 EIB Home Management

EIB's native object oriented management-level features explained above are exploited by the EIB Home Management concept. At it's core is a commercially available PC (Windows) *Home Assistant* API framework for supervision and control of home networks. It extends EIB with an *easy installation* approach for white and brown goods, aimed at the consumer.

2.9.2 Developing Application Software for EIB

EIB is not bound to any particular processor or processor design. For a particular implementation, a plethora of commercially available tools such as assemblers, compilers and emulators can be put to use. These range from shareware to fully fledged environments. Certain EIB system providers offer an *Integrated Development Environment*, which allows development in ANSI C, powerful debugging and come with dedicated EIB programming infrastructure.

In turn, the Developer's Edition of ETS features the necessary utilities to smoothly import the results into ETE.

2.9.3 System Implementations allow Scaleable Access

An EIB solution developer may gain access to the EIB system by using standard EIB Bus Access Units (BAU's) with various levels of integration (scaleability). Alternatively, he/she may go for a proprietary (but compatible) implementation on any microprocessor chip.

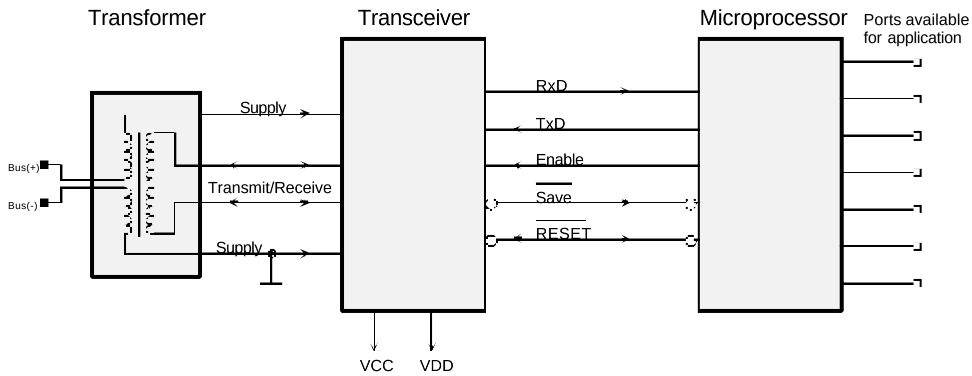
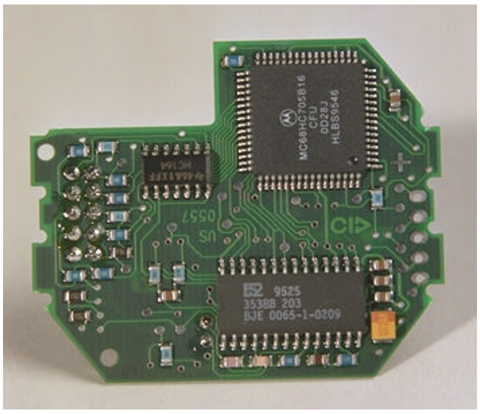


Fig. 1/2-23 Principle Electrical Schema of inductively coupled TP BCU or BIM

The *Bus Coupling Unit* (BCU) is the most complete BAU, with Medium Access, EIB OS firmware, application hosting resources (CPU, RAM, EEPROM, …) and the full PEI and EMI. A BCU is housed in a compact, screened and *ready-to-install* package; construction shapes are aligned to the demands of practical installation, like DIN-rail of flush-mounting. Application-compatible BCU's are available for all EIB media.

EIB *Bus Interface Modules* (BIM) fulfil the same tasks of bus access and application host, as explained in the previous paragraph - but with only the electrical PEI (with EMI), and no EMC shielding or housing. This makes them ideal for tighter integration into the application-specific solution. A high-end C programmable TP BIM is available with 8-32 k EEPROM.



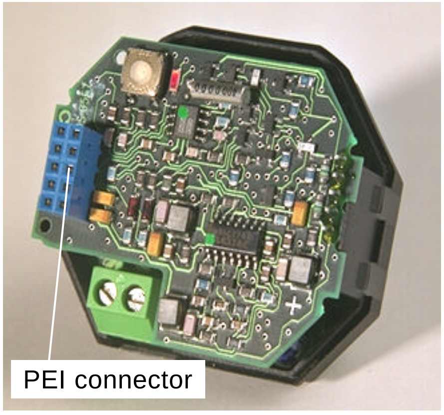


Fig. 1/2-24 The EIB PL BCU for flush-mounting

BAU building blocks may also be obtained as a *chip set*. To facilitate rapid 3rd party implementations of the EIB systems, the full source code can be obtained at non-discriminatory conditions.

2.10 Conformity & Quality

2.10.1 Conformity Assessment Concept

To bear the EIB trademark, a product shall fulfill all the requirements stipulated in the ad hoc sections of the EIBA Handbook.

However first of all some basic question must be answered:

- **Which equipment may receive an EIB Certificate?**

  All products directly connected to the bus such as:

  - **basic components** (cables, connection elements for rail and bus connection, bus power supplies, bus related high-voltage suppression elements,...),

  - **system components**, such as Bus couplers, line couplers, backbone couplers , repeaters, which implement the EIB Protocol,

- **EIB Bus devices** which implement the EIB Protocol through accessing the OSI application layer (level 7) of the available BCU, but also devices with external microcontroller accessing the OSI Transport layer (layer 4) or even Link layer (layer 2) which must also therefore implement the management functions within their application program. Monitoring, software and gateways applications are also considered as EIB Bus devices.

- **Where can the EIBA conformity requirements be found?**

  All relevant requirements, including the EIB functional (i.e. mainly conformity with the EIB protocol) and EIB interworking requirements for a given product or group of products, where necessary, as selected by the type test report form, are stipulated **in the currently valid version of the EIBA Handbook**.

- **Which are the applicable EIBA requirements ?**

  Any equipment, which bears the Trademark, shall comply with the EIBA requirements as laid down in the currently valid edition of the EIBA Handbook. It is not allowed to consider a requirement as optional and not to implement it or implement it differently.

- **When will amendments to the EIBA Handbook become applicable for certification?**

How will certified equipment be treated then ?

The continuous development of the EIBA Handbook requirements and also the modification of a selection of existing requirements of the Handbook in a type test report form, may have an impact on products to be certified or already certified to an earlier set of EIBA requirements.

When amending the requirements EIBA, in accordance with the relevant part of its Internal Regulations, will fix target dates (normally 6 months, but could be less than 6 months in urgent cases, e.g. for functional safety reasons) from which on these amendments shall become applicable.

The EIBA Trademark License Agreement caters for a review of the license conditions including the then applicable conformity requirements every three years.

2.10.2 Quality - General Overview

A high quality level of EIB products is one of the major aims of the EIBA.

The high quality guarantees the functionality offered under the environmental conditions specified, as well as a high level of reliability of the products and the system. Quality serves the availability for the user.

Therefore, an adequate Quality Management System is a prerequisite to provide "QUALITY". EIB products shall have a high reliability. The user will not only see a single product, but the whole EIB-system and its availability.

To meet the high quality level requirements of the EIBA, the manufacturers shall introduce a **Quality Management System (QMS)**, which shall eventually be certified according to ISO 9001 or 9002.

2.11 Spectrum of Available Products

In spring 1999, roughly 50 manufacturers offer 2 500 commercial products, covering application domains such as heating control, energy management, security, time and event management, lighting control. Though compatible, EIB implementations are marketed under various brand names such as instabus, Tebis, i-bus EIB, Powernet, Home Electronic System, Domotik, ImmoCAD etc.