



Konnex Scientific Conference 2005 in Pisa

Implementation of a BACnet – KNX/EIB gateway

Content

- 1 What is BACnet?
- 2 BACnet to KNX/EIB gateway
 - 2.1 Physical build up
 - 2.2 Configuration of KNX/EIB, BACnet and the Gateway
- 3 Overview about ASHRAE 135-2004 Annex H.5: Using BACnet with KNX/EIB
 - 3.1 Object structures
 - 3.2 Mapping rules for BACnet Device Object Type
 - 3.3 Mapping rules for other BACnet Object Types
 - 3.4 Mappings of EIB Functional Blocks
 - 3.5 Defining Proprietary Object Types
 - 3.6 Defining Proprietary Properties

1 What is BACnet?

BACnet (**B**uilding **A**utomation and **C**ontrol **net**works) is an open protocol for communication systems in buildings and is defined by the ASHRAE (**A**merican **S**ociety of **H**eating, **R**efrigeration, and **A**ir **C**onditioning **E**ngineers Inc.) in ASHRAE 135-2004. Since 2003 BACnet is an ISO standard so a worldwide standard. It was developed to replace the proprietary systems in building automation in order to reduce the engineering costs of a project and make the interoperability of systems from different manufacturers easier. BACnet is more located in the automation and management level but can be also in the field level.

BACnet works by dividing the problem of interoperability into three distinct areas, and by defining methods and standards for implementing each (figure 1).

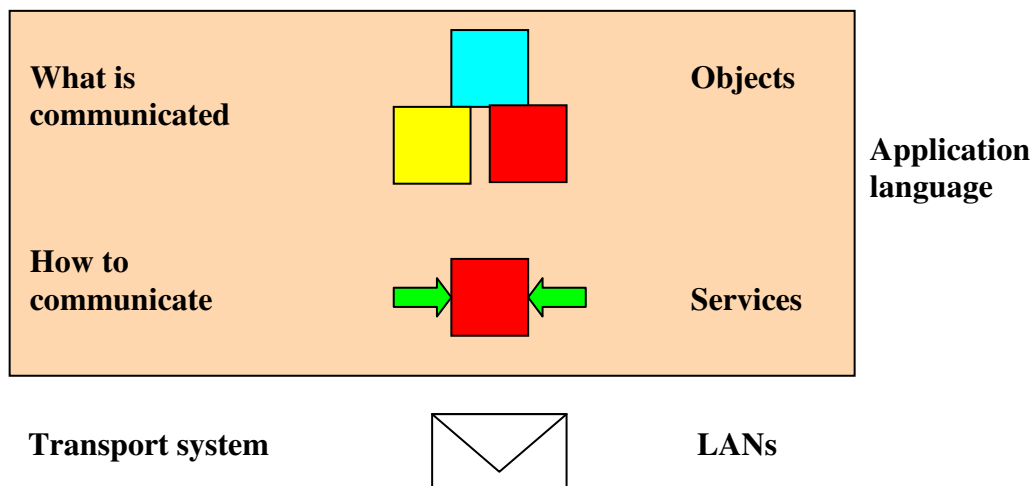


Figure 1: How does BACnet work

- **What is communicated:** BACnet models common building automation and control functions as collections of information called objects. BACnet defines 18 standard objects including analog inputs and outputs, binary inputs and outputs, and schedules.
- **How to Communicate:** BACnet defines methods for exchanging data. These are called services and are used by system suppliers to read and write information between systems.

- Local Area Networks (LANs): BACnet can work over a variety of commonly available network technologies used in our industry including Ethernet, ARCNET, RS-232, RS-485 and the LonTalk data link layers.

2 BACnet to KNX/EIB gateway

2.1 Physical build up

There are two possible ideas of the physical build up. The first idea is that the gateway has only one port. This is an ethernet adapter, connecting to ethernet with the BACnet/IP protocol and the EIBnet/IP protocol running on it (figure 2). The second possibility is that the gateway has two ports. One port is connected to ethernet like in the first idea, but this time there runs only the BACnet/IP protocol. The second port is connected with KNX/EIB twisted pair (figure 3).

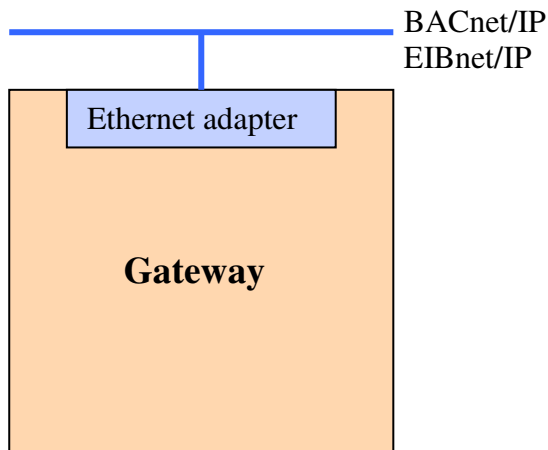


Figure 2: physical build up, idea 1

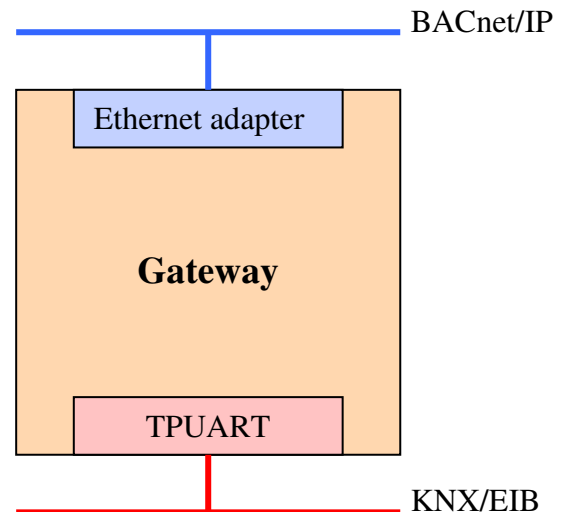


Figure 3: physical build up, idea 2

2.2 Configuration of KNX/EIB, BACnet and the gateway

The configuration of a gateway is always an important point for the acceptance of it. Conventionally, gateways have to be configured on both protocols connecting to it manually

and the gateway itself has to be configured manually (figure 4). This way is always a problem because there may result inconsistent data by typing in the data manually twice.

A more reliable way is to set up only the data for one protocol and generate the data for the second protocol automatically. For example to set up the BACnet side of the gateway, then program the gateway over BACnet/IP and export data from the BACnet side in order to import them on the KNX/EIB side (figure 5). It is also possible to configure the KNX/EIB side of the gateway, then program the gateway over KNX/EIB and import the data from KNX/EIB in the BACnet side.

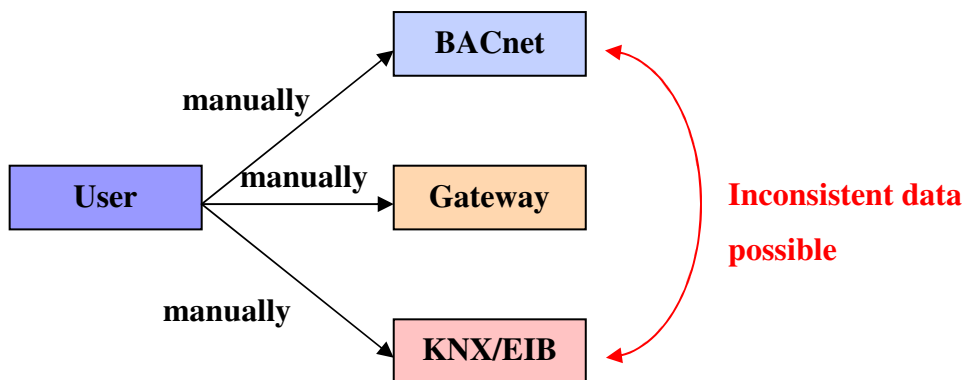


Figure 4: manual configuration of the gateway

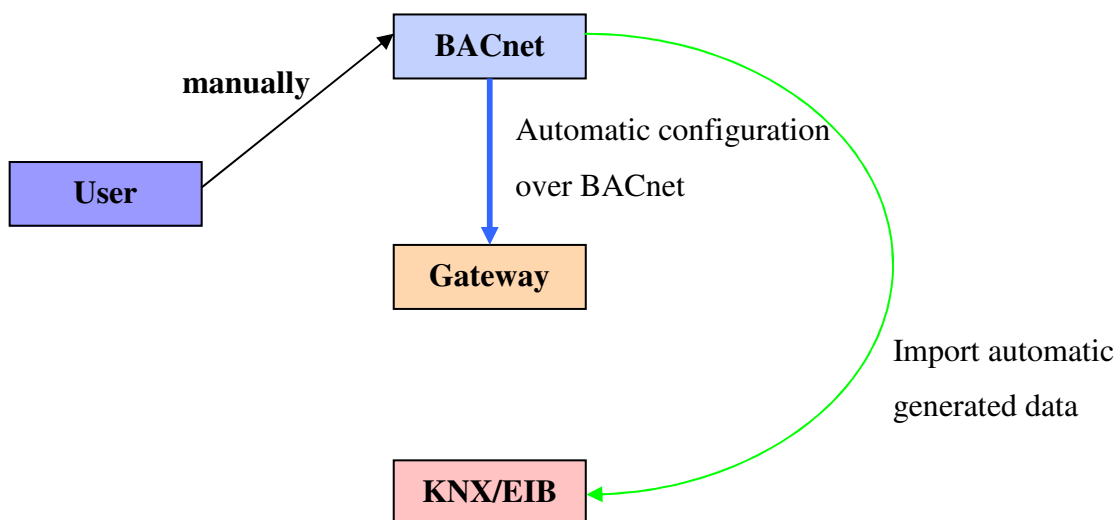


Figure 5: partly automatic configured system with gateway



3 Overview about ASHRAE 135-2004 Annex H.5: Using BACnet with KNX/EIB

3.1 Object structures of KNX/EIB and BACnet

“EIB Functional Blocks not only describe the semantics of a function in the English language, but also define how to access the services associated with that function. This is done on the basis of "Datapoints." The Datapoints are divided into two principal categories, input and output. A Functional Block consists of a non-empty collection of one or more input and output Datapoints. At least one Datapoint is required.

Functional Blocks are contained within Physical Devices. A Physical Device implements at least one Functional Block. It may be regarded as a container for, or collection of, Functional Blocks.”¹

“BACnet's object types define functions in terms of semantics and the services used to access these functions. To accomplish this task, BACnet object types contain properties. An object type consists of a non-empty collection of properties, some of which are mandatory while others may be optional.

BACnet also defines a Device object and a "BACnet Device" contains a collection of instances of object types. Each BACnet Device contains one, and only one, Device object. Typically, each physical device corresponds to a single BACnet Device and therefore contains a single Device object.”¹

“EIB Functional Blocks are comparable to BACnet object types while EIB Datapoints correlate to BACnet properties.”¹

3.2 Mapping rules for BACnet Device Object Type

Annex H.2 describes explicitly the mapping rules for the BACnet Device Object Type, because every BACnet Device must have this object.



3.3 Mapping rules for other BACnet Object Types

Furthermore the mapping rules for other BACnet object types are described including EIB Datapoint Types with the corresponding BACnet Engineering Units.

3.4 Mappings of EIB Functional Blocks

In H.5.4 the mapping of the EIB Functional Blocks Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value und the Dimming Actuator is shown. The mapping of Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value is very easy because there are the same BACnet Object Types and the semantics of these two data structures are identical and the required properties of the BACnet Object Type can be mapped. For example the Analog Output Mapping is shown in figure 6.

Property Identifier	Property Datatype	O/R/W	Mapping
Object_Identifier	BACnetObjectIdentifier	R	As specified in H.5.3.1
Object_Name	CharacterString	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	R	As specified in H.5.3.3
Present_Value	REAL	W	PID_ANALOG_SET.Value
Description	CharacterString	O	As specified in H.5.3.5
Status_Flags	BACnetStatusFlags	R	As specified in H.5.3.6
Event_State	BACnetEventState	R	NORMAL
Reliability	BACnetReliability	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	R	As specified in H.5.3.11
Priority_Array	BACnetPriorityArray	R	As specified in H.5.3.12
Relinquish_Default	REAL	R	As specified in H.5.3.13
Profile_Name	CharacterString	R	"74-EIB_AnalogOutput"

Figure 6: Analog Output Mapping¹

The Dimming Actuator Functional Block is mapped to the standard BACnet Analog Value Object, which is extended by the addition of 4 EIB-specific properties. See figure 7.



Property Identifier	Property Datatype	S/P ¹	O/R/W	Mapping
Object_Identifier	BACnetObjectIdentifier	S	R	As specified in H.5.3.1
Object_Name	CharacterString	S	R	As specified in H.5.3.2
Object_Type	BACnetObjectType	S	R	As specified in H.5.3.3
Description	CharacterString	S	O	As specified in H.5.3.5
Device_Type	CharacterString	S	O	Device's functional description from manufacturer data from ETS
Present_Value	REAL	S	R	PID_ANALOG_PRESENT.Value
Status_Flags	BACnetStatusFlags	S	R	As specified in H.5.3.6
Event_State	BACnetEventState	S	R	NORMAL
Reliability	BACnetReliability	S	O	As specified in H.5.3.8
Out_Of_Service	BOOLEAN	S	R	As specified in H.5.3.9
Units	BACnetEngineeringUnits	S	R	As specified in H.5.3.11
COV_Increment	REAL	S	R	1.0
Profile_Name	CharacterString	S	R	"74-EIB_DimmingActuator"
Target_Value	REAL	P	W	PID_ANALOG_SET.Value
Present_Bin_Value	BACnetBinaryPV	P	R	PID_BOOLEAN_PRESENT.Value
Target_Bin_Value	BACnetBinaryPV	P	W	PID_BOOLEAN_SET.Value
Dimming_Control ²	REAL	P	O/W	PID_CONTROL_SET.Value

¹ S/P = Standard/Proprietary property

² Because BACnet specifies that a writeable property must also be readable, the value returned when the Dimming_Control property is read shall be 0.0

Figure 7: Dimming Actuator Mapping¹

3.5 Defining Proprietary Object Types

“If it is not possible to map an EIB Functional Block to an existing BACnet object type, a new proprietary object type must be defined. Such object types shall be assigned an object type enumeration greater than 127 and contain the Profile_Name property to uniquely identify such object types and to provide a reference to an object-specific profile or description of the object type. At a minimum, proprietary object types must have the Object_Identifier, Object_Name, Object_Type and Profile_Name properties”¹

3.6 Defining Proprietary Properties

“If it is possible to map a Functional Block to a standard BACnet object type but Datapoints exist within the EIB standard that cannot be mapped to existing object properties, or if a completely new proprietary object type with proprietary properties is defined, it will be necessary to define proprietary properties for that object type. Proprietary properties shall be assigned property identifiers greater than 511 and the profile pointed to by the Profile_Name property of the object type shall provide the property name, datatype and conformance code for each such profile-specific property.”¹



¹ ANSI/ASHRAE Standard 135-2004, H.5