System Specifications

KNXnet/IP

Overview

3

8

1

Summary

This document provides the overview over the KNXnet/IP specifications.

Version 01.04.02 is a KNX Approved Standard.

This document is part of the KNX Specifications v2.1.

Document updates

Version	Date	Modifications
1.0 DP	2004.01.07	Final version for Release for Voting
1.1 DV	2005.05.27	Preparation of the Draft for Voting.
1.2 DV	2006.02.13	Preparation of the Draft for Voting.
1.3 AS	2008.07.02	Publication of the Approved Standard.
1.4 AS	2008.09.04	• AN106 "Phasing out TP0" integrated • AN109 "Phasing out PL132" integrated
1.4 AS	2009.06.29	• Preparation for inclusion in the KNX Specifications v2.0: editorial update.
1.4.01 AS	2011.01.04	• Added KNX IP as communication medium to Table 15.
01.04.02	2013.10.28	Editorial updates for the publication of KNX Specifications 2.1.

References

[01] Chapter 3/7/2 “Datapoint Types”

A general reference is made to the RFCs [1] defining the Internet Protocol. These documents can be obtained on the Internet at http://www.ietf.org/rfc.html.

[02] ISO/IEC Directives, Part 3, 1997 – Annex E, Verbal forms of expression of provisions

Filename: 03_08_01 Overview v01.04.02 AS.docx
Version: 01.04.02
Status: Approved Standard
Savedate: 2013.10.28
Number of pages: 18

[1] Request for Comment: Internet Standards defined by the Internet Engineering Task Force (IETF) are first published as RFCs.

Contents

1 Introduction

1.1 Scope

This specification defines the integration of KNX protocol implementations on top of Internet Protocol (IP) networks, called KNXnet/IP. It describes a standard protocol for KNX devices connected to an IP network, called KNXnet/IP devices. The IP network acts as a fast (compared to KNX transmission speed) backbone in KNX installations.

Widespread deployment of data networks using the Internet Protocol (IP) presents an opportunity to expand building control communication beyond the local KNX control bus providing:

- remote configuration,
- remote operation (including control and annunciation),
- fast interface from LAN to KNX and vice versa, and
- WAN connection between KNX systems (where an installed KNX system is at least one Subnetwork).

A KNXnet/IP system contains at least these elements:

- one KNX Subnetwork with up to 64 (255) KNX devices
  OR
  one KNX segment (KNX TP1, KNX RF, KNX PL110),
- an KNX-to-IP network connection device
  (called KNXnet/IP Server),

and typically additional

- software for remote functions residing on e.g. a workstation
  (may be iETS, database application, BACnet Building Management System, browser ...).

Figure 1 shows a typical scenario where a KNXnet/IP Client (e.g. running ETS) accesses multiple KNX installed systems or KNX Subnetworks via an IP network. The KNXnet/IP Client may access one or more KNXnet/IP Servers at a time. For Subnetwork routing server-to-server communication is possible.

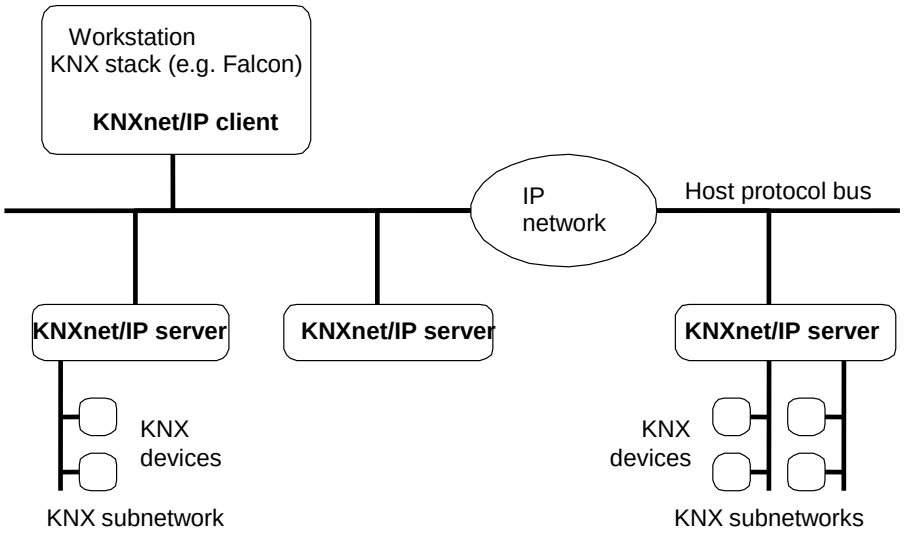

Figure 1 – Device types and configuration examples

1.2 Definitions, acronyms and abbreviations

- The key words "shall", "shall NOT", "SHOULD", "SHOULD NOT", "MAY", “MAY NOT”, “CAN”, and “CANNOT” in this document are to be interpreted as described in ISO/IEC Directives, Part 3, 1997.
- If any KNX protocol is referenced, it is used in the meaning as defined by the KNX Specifications v2.0.
- A KNX node is considered to be any device that implements a KNX protocol stack and fulfils the requirements for certification according to the KNX Specifications.
- A Communication channel (or simply channel) is a logical connection between a KNXnet/IP Client and a KNXnet/IP Server (or, in case of routing, between two or more KNXnet/IP Servers). A communication channel consists of one or more connections on the definition of the host protocol used for KNXnet/IP.
- A KNXnet/IP Server is a KNX device that has physical access to a KNX network and implements the KNXnet/IP Server protocol to communicate with KNXnet/IP Clients or other KNXnet/IP Servers (in case of routing) on an IP network channel. A KNXnet/IP Server is by design always also a KNX node.
- A KNXnet/IP Client is an application that implements the KNXnet/IP Client protocol to get access to a KNX Subnetwork over an IP network channel.
- A subnet is a portion of a network that shares a common address component known as the "subnet address". Different network protocols specify the subnet address in different ways.
- All KNX devices are either a KNX or a KNXnet/IP node.
- ETS is the Engineering Tool Software.
- A KNXnet/IP Router is a dedicated type of KNXnet/IP device that routes KNX protocol packets between KNX Subnetworks.
- The term HPAI represents the Host Protocol Address Information structure. This structure holds the IP host protocol address information and is used to address a KNXnet/IP endpoint on another KNXnet/IP device.
- The TTL (Time To Live) of a multicast UDP/IP datagram is the maximum number of IP routers that may route this datagram. Each IP router the datagram passes decrements the TTL by one; the local host adapter does this as well. When the TTL has reached zero, the router discards the datagram. When sending a datagram from the local host adapter, a TTL of zero means that the datagram never leaves the host. A TTL of one means that the datagram never leaves the local network (it is not routed).
- KNXnet/IP Tunnelling refers to the encapsulation of cEMI (common EMI) frames within Internet Protocol (IP) packets.
- The Internet Control Message Protocol (ICMP) is an extension to the Internet Protocol (IP) defined by RFC [2)] 792. ICMP supports packet containing error, control, and informational messages. The PING command, for example, uses ICMP to test an Internet connection.
- Internet Group Management Protocol (IGMP) is defined in RFC 1112 as the standard for IP multicasting in the Internet. It's used to establish host memberships in particular multicast groups on a single network. The mechanisms of the protocol allow a host to inform its local router - using Host Membership Reports - that it wants to receive messages addressed to a specific multicast group. All hosts conforming to level 2 of the IP multicasting specification require IGMP.

[2)] Request for Comment: Internet Standards defined by the Internet Engineering Task Force (IETF) are firstly published as RFCs.

- An IP channel is a logical connection between two IP host/port endpoints. IP channels are either a guaranteed, reliable TCP (transmission control protocol) or an unreliable point-to-point or multicast (in case of routing) UDP (user datagram protocol) connection.
- A communication channel as defined by the KNXnet/IP Core specification is represented by one or two IP channels.

2 KNXnet/IP documents

2.1 General

This document and subsequent parts define the integration of KNX protocol implementations within the Internet Protocol (IP) named KNXnet/IP or EIBnet/IP for continuity with the CEN pre-standard EIBnet as defined by CEN TC 247. EIBnet was introduced as an expansion of EIB into the information technology realm and was incorporated as a building controls technology in CEN TC 247. KNXnet/IP is the logical successor to EIBnet.

This specification defines a standard protocol, which is implemented within KNX devices, the Engineering Tool Software (ETS) and other implementations to support KNX data exchange over IP networks. In fact, KNXnet/IP provides a general framework that accommodates several specialized “Service Protocols” in a modular and extendible fashion.

The KNXnet/IP specification consists of these parts:

- Part 1, Overview
- Part 2, Core Specification
- Part 3, Device Management
- Part 4, Tunnelling
- Part 5, Routing
- Part 6, Remote Logging
- Part 7, Remote Configuration and Diagnosis
- Part 8, Object Server

Additional parts may be added to the KNXnet/IP specification in the future at which time this Part 1 “Overview” shall be updated.

KNXnet/IP supports different software implementations on top of the protocol. Specifically these software implementations can be Building Management, Facility Management, Energy Management, or simply Data Base and SCADA (Supervision, Control and Data Acquisition) packages.

Most of these packages need to be configured for the specific user application. In order to simplify this process and cut cost for engineering, KNXnet/IP provides simple engineering interfaces, namely a description “language” for the underlying KNX system. This may be done offline, e.g. generated as an ETS export file, or online by a mechanism that self-describes the underlying KNX system (reading data from the system itself).

KNXnet/IP supports

- On-the-fly change-over between Operational modes (configuration, operation);
- Event driven mechanisms;
- Connections with a delay time greater than $t_{KNX\_transfer\_timeout}$ (e.g. network connection via satellite).

2.2 Part 1, Overview

Part 1 “Overview” is this document.

2.3 Part 2, Core specification

Part 2 “Core Specification” defines a standard protocol that is implemented within KNXnet/IP devices and the Engineering Tool Software (ETS) to support KNX data exchange over IP networks.

This specific implementation of the protocol over the Internet Protocol (IP) is called KNXnet/IP.

This specification addresses:

- definition of data packets sent over the IP host protocol network for KNXnet/IP communication
- Discovery and self-description of KNXnet/IP Servers
- Configuring and establishing a communication channel between a KNXnet/IP Client and a KNXnet/IP Server

2.4 Part 3, Device Management

Part 3 “Device Management” defines services for remote configuration and remote management of KNXnet/IP Servers.

2.5 Part 4, Tunnelling

Part 4 “Tunnelling” defines services supporting all ETS functions for download, test, and analysis of KNX devices on KNX networks connected via KNXnet/IP Servers. This includes changes of single KNX Device Object Properties.

Tunnelling assumes that a data transmission round-trip between ETS or any other KNXnet/IP Tunnelling client and KNXnet/IP Servers takes less than $t_{KNX\_transfer\_timeouts}$.

It describes direct communication between ETS and target KNX device with single telegrams exchanged between both (like iETS). This mode also allows for change of Properties in devices.

2.6 Part 5, Routing

Part 5 “Routing” defines services, which route KNX telegrams between KNXnet/IP Servers through the IP network.

2.7 Part 6, Remote Logging

Part 6 “Remote Logging” defines services for off-line monitoring of KNX networks. KNXnet/IP Servers or network connection devices are configured to capture KNX network traffic in memory and send this information to ETS when ETS reconnects to the KNXnet/IP Server.

2.8 Part 7, Remote Configuration and Diagnosis

Part 7 “Remote Configuration and Diagnosis” defines configuration and diagnosis services for networks with data transmission delay times, which are greater than $t_{KNX\_transfer\_timeout}$.

2.9 Part 8, Object Server

Part 8 “Object Server” defines services for remote operation of KNX networks.

Operation is defined as any exchange of data between a KNX installation and one or more software packages for annunciation/alarming, visualization, and control.

With Tunnelling and Routing a KNX packet is wrapped into an IP frame, which is sent to a receiving IP address (single cast or multicast). The receiving software package interprets the KNX telegram and processes it.

With Object Server services the KNXnet/IP network connection device acts towards the KNX network as a KNX device with Datapoints. A software package running on a PC or Workstation can read the configuration of this device that contains information for easy engineering / configuration of that software package. The KNXnet/IP network connection device sends true IP telegrams, which send “enriched” data of the Datapoints.

3 Mandatory and optional implementation of IP protocols

3.1 General

KNXnet/IP uses existing IP protocols where possible unless their use implies an undue burden with regards to memory and implementation requirements for the intended service.

The following table shows mandatory (M) and optional (O) implementation of IP protocols by KNXnet/IP service types. Although this table refers to the KNXnet/IP Server it also indicates, which IP protocols shall be implemented by the KNXnet/IP Client. Any non-applicable IP protocol is marked as (na).

Table 1 – KNXnet/IP service types and IP protocols

IP protocol	Service Type						
	Core	Device Management	Tunnelling	Routing	Remote Logging	Remote Config.	Object Server
ARP	M	M	M	M	M	M	M
RARP	O	O	O	O	O	O	O
Support of fixed IP address	M	M	M	M	M	M	M
BootP (Client)[3]	M	M	M	M	M	M	M
DHCP (Client) [3]	M	M	M	M	M	M	M
UDP	M	M	M	M	O	M	M
TCP	O	O	O	na	M	O	O
ICMP	M	M	M	M	M	M	M
IGMP	M	M	na	M	na	na	na

Other Internet protocols like NTP (network time protocol), FTP (file transfer protocol), HTTP (hypertext transfer protocol), SMTP (simple message transfer protocol), DNS (domain name system), and SNMP (simple network management protocol) MAY be used but are not within the scope of the KNXnet/IP protocol.

3.2 Minimum KNXnet/IP device requirements

KNXnet/IP service types as defined in this specification require the implementation of a minimal set of IP protocols for interworking.

KNXnet/IP Servers shall implement these IP protocols: ARP, BootP, UDP, ICMP and IGMP. Other IP protocols may be required for specific services.

3.3 Network environment

Because KNXnet/IP Servers use IP this specification does not require any specific medium carrying the IP datagrams.

[3] BootP / DHCP: Either one shall be implemented by a KNXnet/IP device.

It is recommended to use a medium that carries at least twice the bitrate of all KNXnet/IP Routers connected to this medium. For a point-to-point, e.g. PPP, connection this would be at least 19 200 bit/s, for a network with up to 400 KNXnet/IP Servers this would be at least 8 Mbit/s.

10BaseT is RECOMMENDED as a minimum for KNXnet/IP Servers using Ethernet as physical medium.

3.4 Addressing

KNXnet/IP Servers are typically connected to one KNX Subnetwork and to an IP network. Hence, KNXnet/IP Servers have one distinct address for each network they are connected to: one KNX Individual Address and one IP address.

Additionally, KNXnet/IP Servers are assigned to a KNXnet/IP project-installation using the same IP multicast address for all KNXnet/IP Servers in a KNXnet/IP project-installation.

3.5 KNXnet/IP Device Classes

A KNXnet/IP device can be a software implementation running on a PC. Hence ETS or any other software implementing KNXnet/IP services is viewed as a KNXnet/IP device.

The definition of KNXnet/IP Device Classes ensures interoperability between KNXnet/IP devices as well as a minimum set of supported KNXnet/IP services for a given KNXnet/IP Device Class.

Table 2 lists mandatory and optional service types for KNXnet/IP Device Classes.

Device Class A encompasses configuration and system maintenance tools. Except for Object Server services all other KNXnet/IP services shall be implemented. ETS is a member of this device class.

Device Class B defines the minimum set of KNXnet/IP services for KNXnet/IP Routers.

Device Class C defines the minimum set of KNXnet/IP services for any other KNXnet/IP device. Building, energy, and facilities management systems are members of this KNXnet/IP Device Class.

Table 2 – KNXnet/IP Device Classes

KNXnet/IP Device Class	Service Type						
	Core	Device Management	Tunnelling	Routing	Remote Logging	Remote Config.	Object Server
A (Configuration and System Maintenance Tools)	M	M	M	M	M	M	O
B (KNXnet/IP Router)	M	M	M	M	O	O	O
C (any other KNXnet/IP device)	M	M	O	O	O	O	O

4 Security considerations

4.1 Introduction

For KNX, security is a minor concern, as any breach of security requires local access to the network. In the case of KNX TP1 or KNX PL110 networks this requires even physical access to the network wires, which in nearly all cases is impossible as the wires are inside a building or buried underground.

Hence, security aspects are less of a concern for KNX field level media.

The intention of KNXnet/IP is to provide a means of using existing data networking technology, i.e. the Internet Protocol, for fast communication between different KNX Subnetworks in the same or different locations.

Unless the data network is completely isolated and only used for KNXnet/IP a number of potential threats need to be considered.

4.2 Categories of threats

4.2.1 Goal

This clause briefly enumerates various types of security threats and mentions typical countermeasures.

4.2.2 Passive attacks

Passive attacks involve a violation of privacy or leak of information to an unauthorized party that “wire-taps” or listens in on the network. Passive attacks are difficult to detect, since they do not modify the network traffic in any way. The primary defence against passive attacks is isolation of network traffic (one cannot eavesdrop or analyse traffic that one cannot see).

- Eavesdropping – interception and examination of packet contents by unauthorized third parties. In addition to network isolation, this threat can be countered by encrypting packets.
- Traffic analysis – interception and analysis of traffic characteristics without examining the contents of individual packets. This threat exists even if packets are encrypted, and it is very difficult to counter except by isolation of network traffic.

4.2.3 Active attacks

Active attacks involve some type of modification to network traffic: adding, deleting, or delaying packets, or changing the contents of packets in some way.

- Masquerade – unauthorized entities forging traffic to make it appear to originate from a legitimate source. This threat can be countered by authentication mechanisms.
- Access control violation – a user accesses a resource that it does not have proper authorization to use. This threat can be countered by message integrity, authentication, and authorization mechanisms (the access control policies must also be secured).
- Modification – intercepting and changing the contents of packets on the network. This threat can be countered by message integrity mechanisms.
- Deletion – preventing packets from arriving at their intended destination (for example, by maliciously configuring a router to drop packets or by deliberately interfering with packet transmissions on the wire). This threat can be very difficult to counter, especially if the attacker can jam packet transmissions on the physical network.
- Delay – delaying arrival of packets at their intended destination. This attack can be achieved by compromising or overloading a router or bridge or by performing a time-limited attack on the packet transmission (thereby causing retransmission delays). This threat is also very difficult to counter.

- Replay – intercepting packets and later resending them on the network. This threat can be countered by authentication, message integrity, and timestamps or message counters.
- Denial of service – this type of attack typically involves injecting a flood of useless traffic on the network to consume resources such as bandwidth or processing time so that the resources are unavailable to service legitimate traffic. This threat is difficult to counter, especially if the attack traffic cannot be easily distinguished from legitimate traffic.

4.3 Countermeasures

This is a list of countermeasures for KNXnet/IP.

(a) Typically, building control data is only exchanged in a closed or tightly surveillanced network. Use KNXnet/IP in an Intranet or over Virtual Private Network only.

(b) Filtering KNXnet/IP datagrams from the network requires network analysis tools and expertise. The content of a KNXnet/IP message is not self-descriptive but requires semantic knowledge residing in ETS. Access to ETS is limited to authorized personnel only.

(c) Use authentication when opening point-to-point connections e.g. for tunnelling or remote logging.

(d) Do not publish default KNXnet/IP IP multicast address.

(e) Use sequence counter on all point-to-point connections e.g. for tunnelling or remote logging.

(f) Use UDP/IP or TCP/IP port numbers, which are filtered by a firewall.

Threat....	is countered by measure(s)
Eavesdropping	(a), (b), (d)
Traffic analysis	(a), (d)
Masquerade	(a), (c), (d)
Access control violation	(a), (c), (d)
Modification	(a), (e)
Deletion	(a)
Delay	(a)
Replay	(c), (e)
Denial of service	(a)

4.4 Conclusion

Most of the listed threats are countered by staying within a network, which tightly supervises and restricts network access from outside the network.

Using authentication when establishing point-to-point connections is an additional means to this end.

It is quite unlikely that legitimate users of a network would have the means to intercept, decipher, and then tamper with the KNXnet/IP without excessive study of the KNX Specifications.

Thus the remaining security threat is considered to be very low and does not justify mandating encryption, which would require considerable computing resources.

5 Addendum A: list of codes

5.1 Goal

This addendum to the KNXnet/IP Overview provides a complete listing of all codes used by KNXnet/IP.

This list shall be updated when codes are added, revised, or deleted from subsequent KNXnet/IP documents.

5.2 Common constants

Table 3 – Common KNXnet/IP constants

Constant name	Value	V.[4)]	Description
KNXNETIP_VERSION_10	10h	1	Identifier for KNXnet/IP protocol version 1.0
HEADER_SIZE_10	06h	1	Constant size of KNXnet/IP header as defined in protocol version 1.0

5.3 KNXnet/IP services

5.3.1 Service type number ranges

0200h … 020Fh KNXnet/IP Core

0310h … 031Fh KNXnet/IP Device Management

0420h … 042Fh KNXnet/IP Tunnelling

0530h … 053Fh KNXnet/IP Routing

0600h … 06FFh KNXnet/IP Remote Logging

0740h … 07FFh KNXnet/IP Remote Configuration and Diagnosis

0800h … 08FFh KNXnet/IP Object Server

5.3.2 Core KNXnet/IP services

Table 4 – KNXnet/IP Core service type identifiers

Service name	Code	V.	Description
SEARCH_REQUEST	0201h	1	Sent by KNXnet/IP Client to search available KNXnet/IP Servers.
SEARCH_RESPONSE	0202h	1	Sent by KNXnet/IP Server when receiving a KNXnet/IP SEARCH_REQUEST.
DESCRIPTION_REQUEST	0203h	1	Sent by KNXnet/IP Client to a KNXnet/IP Server to retrieve information about capabilities and supported services.

4) Protocol version since which the constant, error code or service type is defined and may therefore be supported. As this is version 1 of the specification, this value is 1 for all services defined.

Table 4 – KNXnet/IP Core service type identifiers (continued)

Service name	Code	V.	Description
DESCRIPTION_RESPONSE	0204h	1	Sent by KNXnet/IP Server in response to a DESCRIPTION_REQUEST to provide information about the server implementation.
CONNECT_REQUEST	0205h	1	Sent by KNXnet/IP Client for establishing a communication channel to a KNXnet/IP Server.
CONNECT_RESPONSE	0206h	1	Sent by KNXnet/IP Server as answer to CONNECT_REQUEST telegram.
CONNECTIONSTATE_REQUEST	0207h	1	Sent by KNXnet/IP Client for requesting the connection state of an established connection to a KNXnet/IP Server.
CONNECTIONSTATE_RESPONSE	0208h	1	Sent by KNXnet/IP Server when receiving a CONNECTIONSTATE_REQUEST for an established connection.
DISCONNECT_REQUEST	0209h	1	Sent by KNXnet/IP device, typically the KNXnet/IP Client, to terminate an established connection.
DISCONNECT_RESPONSE	020Ah	1	Sent by KNXnet/IP device, typically the KNXnet/IP Server, in response to a DISCONNECT_REQUEST.

5.3.3 Device Management services

Table 5 – KNXnet/IP Device Management service type identifiers

Service name	Code	V.	Description
DEVICE_CONFIGURATION_REQUEST	0310h	1	Reads/Writes KNXnet/IP device configuration data (Interface Object Properties)
DEVICE_CONFIGURATION_ACK	0311h	1	Sent by a KNXnet/IP device to confirm the reception of the DEVICE_CONFIGURATION_REQUEST.

5.3.4 Tunnelling services

Table 6 – Tunnelling KNXnet/IP service type identifiers

Service name	Code	V.	Description
TUNNELING_REQUEST	0420h	1	Used for sending and receiving single KNX telegrams between KNXnet/IP Client and - Server.
TUNNELING_ACK	0421h	1	Sent by a KNXnet/IP device to confirm the reception of the TUNNELING_REQUEST.

5.3.5 Routing services

Table 7 – KNXnet/IP Routing service type identifier

Service name	Code	V.	Description
ROUTING_INDICATION	0530h	1	Used for sending KNX telegrams over IP networks. This service is unconfirmed.
ROUTING_LOST_MESSAGE	0531h	1	Used for indication of lost KNXnet/IP Routing messages. This service is unconfirmed.

5.4 Connection types

Table 8 – Connection types

Connection type	Value	V.	Description
DEVICE_MGMT_CONNECTION	03h	1	Data connection used to configure a KNXnet/IP device.
TUNNEL_CONNECTION	04h	1	Data connection used to forward KNX telegrams between two KNXnet/IP devices.
REMLOG_CONNECTION	06h	1	Data connection used for configuration and data transfer with a remote logging server.
REMCONF_CONNECTION	07h	1	Data connection used for data transfer with a remote configuration server.
OBJSVR_CONNECTION	08h	1	Data connection used for configuration and data transfer with an Object Server in a KNXnet/IP device.

5.5 Error codes

5.5.1 Common error codes

Table 9 – Common KNXnet/IP error codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	Operation successful
E_HOST_PROTOCOL_TYPE	01h	1	The requested host protocol is not supported by the KNXnet/IP device.
E_VERSION_NOT_SUPPORTED	02h	1	The requested protocol version is not supported by the KNXnet/IP device.
E_SEQUENCE_NUMBER	04h	1	The received sequence number is out of order.

5.5.2 CONNECT RESPONSE status codes

Table 10 – Common CONNECT_RESPONSE status codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The connection is established successfully.
E_CONNECTION_TYPE	22h	1	The KNXnet/IP Server device does not support the requested connection type.
E_CONNECTION_OPTION	23h	1	The KNXnet/IP Server device does not support one or more requested connection options.
E_NO_MORE_CONNECTIONS	24h	1	The KNXnet/IP Server device cannot accept the new data connection because its maximum amount of concurrent connections is already used.

5.5.3 CONNECTIONSTATE_RESPONSE status codes

Table 11 – CONNECTIONSTATE_RESPONSE status codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The connection state is normal.
E_CONNECTION_ID	21h	1	The KNXnet/IP Server device cannot find an active data connection with the specified ID.
E_DATA_CONNECTION	26h	1	The KNXnet/IP Server device detects an error concerning the data connection with the specified ID.
E_KNX_CONNECTION	27h	1	The KNXnet/IP Server device detects an error concerning the KNX connection with the specified ID.

5.5.4 Tunnelling CONNECT_ACK error codes

Table 12 – Tunnelling CONNECT_ACK error codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The message is received successfully.
E_TUNNELING_LAYER	29h	1	The KNXnet/IP Server device does not support the requested KNXnet/IP Tunnelling layer.

5.5.5 Device Management DEVICE_CONFIGURATION_ACK status codes

Table 13 – Device Management DEVICE_CONFIGURATION_ACK status codes

Error constant	Value	V.	Description
E_NO_ERROR	00h	1	The message is received successfully.

5.6 Description Information Block (DIB)

This table lists the valid description type codes.

Table 14 – Description type codes

Description type	Value	V.	Description
DEVICE_INFO	01h	1	Device information e.g. KNX medium.
SUPP_SVC_FAMILIES	02h	1	Service families supported by the device.
Reserved	03h – FDh	1	Reserved for future use.
MFR_DATA	FEh	1	DIB structure for further data defined by device manufacturer.
Not used	FFh	1	Not used.

Table 15 – KNX medium codes

KNX medium	Value	V.	Description
reserved	01h	1	reserved
TP1 (KNX TP)	02h	1	KNX TP1 = KNX TP
PL110	04h	1	KNX PL110
reserved	08h	1	reserved
RF	10h	1	KNX RF
KNX IP	20h	1	KNX IP

These values shall be identical to the encoding of DPT_Media as specified in [01]; exactly one single bit shall be set.

5.7 Host protocol codes

This table lists the valid host protocol codes.

Table 16 – Host protocol codes for IP network

Constant name	Value	V.	Description
IPV4_UDP	01h	1	Identifies an Internet Protocol version 4 address and port number for UDP communication.
IPV4_TCP	02h	1	Identifies an Internet Protocol version 4 address and port number for TCP communication.

5.8 Timeout constants

This table lists the valid timeout constants.

Table 17 – Timeout constants

Constant name	Value	V.	Description
CONNECT_REQUEST_TIMEOUT	10 s	1	KNXnet/IP Client shall wait for 10 seconds for a CONNECT_RESPONSE frame from KNXnet/IP Server.
CONNECTIONSTATE_REQUEST_-TIMEOUT	10 s	1	KNXnet/IP Client shall wait for 10 seconds for a CONNECTIONSTATE_RESPONSE frame from KNXnet/IP Server.
DEVICE_CONFIGURATION_-REQUEST_TIMEOUT	10 s	1	KNXnet/IP Client shall wait for 10 seconds for a DEVICE_CONFIGURATION_RESPONSE frame from KNXnet/IP Server.
TUNNELING_REQUEST_TIMEOUT	1 s	1	KNXnet/IP Client shall wait for 1 second for a TUNNELING_ACK response on a TUNNELING_REQUEST frame from KNXnet/IP Server.
CONNECTION_ALIVE_TIME	120 s	1	If the KNXnet/IP Server does not receive a heartbeat request within 120 seconds of the last correctly received message frame, the server shall terminate the connection by sending a DISCONNECT_REQUEST to the client's control endpoint.

5.9 Internet Protocol constants

This table lists the Internet Protocol relevant values.

Table 18 – KNXnet/IP Internet Protocol constants

Description	Value	V.
KNXnet/IP Port Number	3671	1
KNXnet/IP System Setup Multicast Address	224.0.23.12	1