System Specifications

KNXnet/IP

Remote Diagnosis and Configuration

3

8

7

Summary

This document defines a standard protocol that is implemented within KNX devices and the Engineering Tool Software (ETS) to support KNX data exchange for remote configuration and diagnosis over IP networks.

Version 01.01.02 is a KNX Approved Standard.

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

Document updates

Version	Date	Modifications
1.0.00	2010.06.14	• **AN123 "KNXnet/IP Remote Configuration and Diagnosis"** integrated. • Publication of the Approved Standard v1.0.00.
01.01.00	2013.07.16	• Explicit indication of the DIBs to transfer in 4.4.2.
01.01.01	2013.07.18	• Correction also of clause 2.3.
01.01.02	2013.10.28	Editorial updates for the publication of KNX Specifications 2.1.

References

[01] Chapter 3/6/3 "External Message Interface"
[02] Chapter 3/8/1 "KNXnet/IP Overview"
[03] Chapter 3/8/2 "KNXnet/IP Core"
[04] Chapter 3/8/2 "KNXnet/IP Management"
[05] Volume 6 "Profiles"

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

Filename: 03_08_07 KNXnetIP Remote Configuration and Diagnosis v01.01.02 AS.docx
Version: 01.01.02
Status: Approved Standard
Savedate: 2013.10.28
Number of pages: 16

Contents

1 General

1.1 Scope

This specification defines the integration of KNX protocol implementations on top of Internet Protocol (IP) networks, called KNXnet/IP. It specifies 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.

An overview of KNXnet/IP is presented in [02].

This Chapter 3/8/7 “Remote Diagnosis and Configuration” of the KNXnet/IP specification provides services for remote configuration and diagnosis of a KNX installation.

This specification addresses

- the definition of data packets for remote diagnosis via KNXnet/IP communication, and
- the definition of data packets for remote configuration via KNXnet/IP communication.

This document defines a standard protocol that is implemented within KNX devices and the Engineering Tool Software (ETS) to support KNX data exchange for remote configuration and diagnosis over IP networks.

1.2 Definitions, acronyms and abbreviations

Refer to [02] for a list of definitions for the KNXnet/IP specification.

Tables listing implementation requirements use the abbreviations as specified in clause 1.4 in [05].

2 Remote Diagnosis of KNXnet/IP devices

2.1 Introduction

KNXnet/IP devices must support KNXnet/IP Core services including device discovery.

KNXnet/IP devices may receive their IP address via ETS configuration or automatically via DHCP or BootP services. In the latter case or if the network setup is unknown the KNXnet/IP Core Device Discovery may not work or may not deliver enough information to allow for establishing a Tunnelling or other connection with the KNXnet/IP device.

As a device may have an IP address that is not reachable via unicast datagrams by the configuration tool the remote diagnosis and configuration datagrams are used with multicast addressing. Broadcast addressing may be used if multicast addressing does not provide results in a specific network configuration. As the datagrams are transmitted via multicast or optionally via broadcast all KNXnet/IP devices receive the remote diagnosis services in parallel. A Selector is defined to allow for selecting all devices or a specific device via MAC address or Programming Mode.

2.2 REMOTE_DIAGNOSTIC_REQUEST

The REMOTE_DIAGNOSTIC_REQUEST datagram shall be transmitted using multicast or optionally via broadcast. A device that fits the selector shall respond with a REMOTE_DIAGNOSTIC_RESPONSE datagram.

2.3 REMOTE_DIAGNOSTIC_RESPONSE

The REMOTE_DIAGNOSTIC_RESPONSE datagram shall be the response to a REMOTE_-DIAGNOSTIC_REQUEST datagram or to a REMOTE_BASIC_CONFIGURATION_REQUEST datagram. The response shall use the target address of the „discovery endpoint" of the HPAI in the request. The response may contain any number of DIBs. A diagnostic tool analyses only those DIBs that it recognizes. All other DIBs are discarded. The device shall send the DIBS that it supports from Table 1. (The values of the DIBs are defined in [03].)

2.4 REMOTE_BASIC_CONFIGURATION_REQUEST

The REMOTE_BASIC_CONFIGURATION_REQUEST datagram shall be transmitted via multicast or optionally via broadcast. A device that fits the selector shall accept the configuration received with a REMOTE_DIAGNOSTIC_RESPONSE datagram. If a Device Information Block contains write-protected data then that data shall not be overwritten with the data in the DIBs of the configuration request. The configuration request shall only contain DIBs that shall be configured. This service shall be acknowledged with a REMOTE_DIAGNOSTIC_RESPONSE datagram.

2.5 REMOTE_RESET_REQUEST

The REMOTE_RESET_REQUEST datagram shall be transmitted using multicast or optionally via broadcast. A device that fits the selector shall accept the reset command without sending an acknowledgement. It should restart immediately or with a reset to factory default settings before.

3 Configuration and Management

General device management and configuration of KNXnet/IP devices is described in [04].

KNXnet/IP Remote Diagnosis and Configuration does not require any configuration beyond the general device management.

4 Data packet structures

4.1 Introduction

All KNXnet/IP data packets, or frames, shall have a common header, consisting of the protocol version, length information, and the KNXnet/IP service type identifier.

4.2 Common constants

Refer to [02] for a list of valid KNXnet/IP common constants.

4.3 Common error codes

Refer to [02] for a list of valid KNXnet/IP common error codes.

4.4 Remote diagnosis and configuration services

4.4.1 REMOTE_DIAGNOSTIC_REQUEST

The REMOTE_DIAGNOSTIC_REQUEST datagram shall be transmitted using multicast or optionally via broadcast. A device that fits the selector shall respond with a REMOTE_DIAGNOSTIC_RESPONSE datagram.

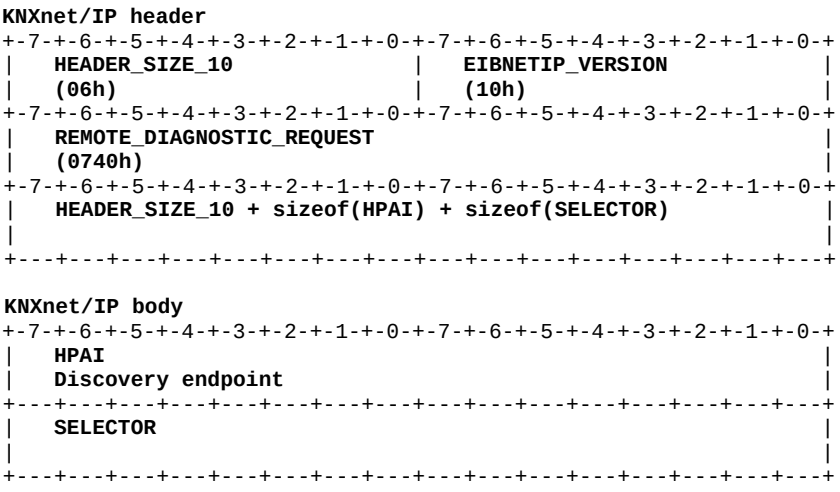


Figure 1 – REMOTE_DIAGNOSTIC_REQUEST frame binary format

4.4.2 REMOTE_DIAGNOSTIC_RESPONSE

The REMOTE_DIAGNOSTIC_RESPONSE datagram shall be the response to a REMOTE_-DIAGNOSTIC_REQUEST datagram or a REMOTE_BASIC_CONFIGURATION_REQUEST datagram. The response shall use the target address of the „discovery endpoint” of the HPAI in the request. The response may contain any number of DIBs. A diagnostic tool analyses only those DIBs that it recognizes. All other DIBs are discarded. The device shall send the DIBS that it supports from Table 1. (The values of the DIBs are defined in [03].)

Table 1 - Description type codes

Description type	Description
IP_CONFIG	IP configuration
IP_CUR_CONFIG	current configuration
KNX_ADDRESSES	KNX addresses

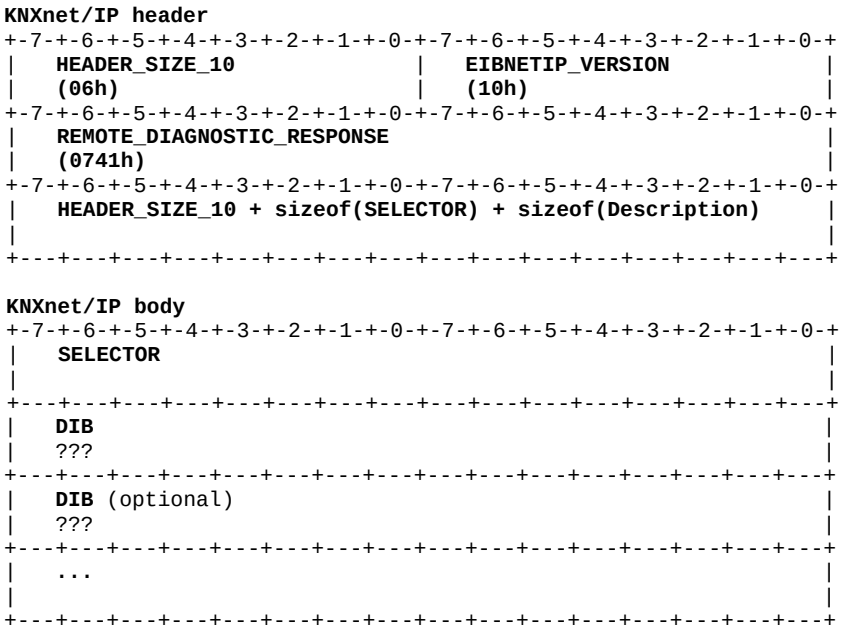


Figure 2 – REMOTE_DIAGNOSTIC_RESPONSE frame binary format

4.4.3 REMOTE_BASIC_CONFIGURATION_REQUEST

The REMOTE_BASIC_CONFIGURATION_REQUEST datagram shall be transmitted via multicast or optionally via broadcast. A device that fits the selector shall accept the configuration received with a REMOTE_DIAGNOSTIC_RESPONSE datagram. If a Device Information Block contains write-protected data then that data shall not be overwritten with the data in the DIBs of the configuration request. The configuration request shall only contain DIBs that shall be configured. This service shall be acknowledged with a REMOTE_DIAGNOSTIC_RESPONSE datagram.

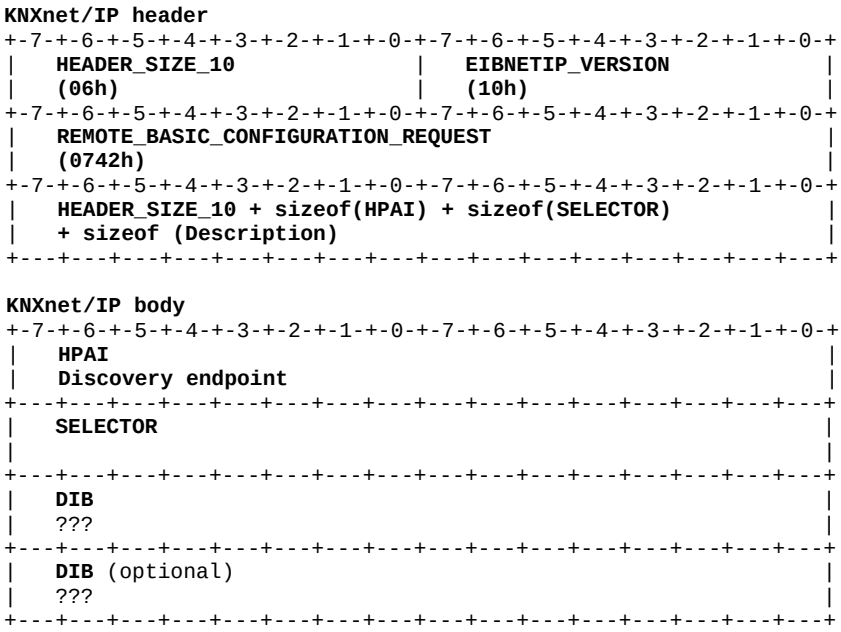


Figure 3 – REMOTE_BASIC_CONFIGURATION_REQUEST frame binary format

4.4.4 REMOTE_RESET_REQUEST

The REMOTE_RESET_REQUEST datagram shall be transmitted via multicast or optionally via broadcast. A device that fits the selector shall accept the reset command without sending an acknowledgement.

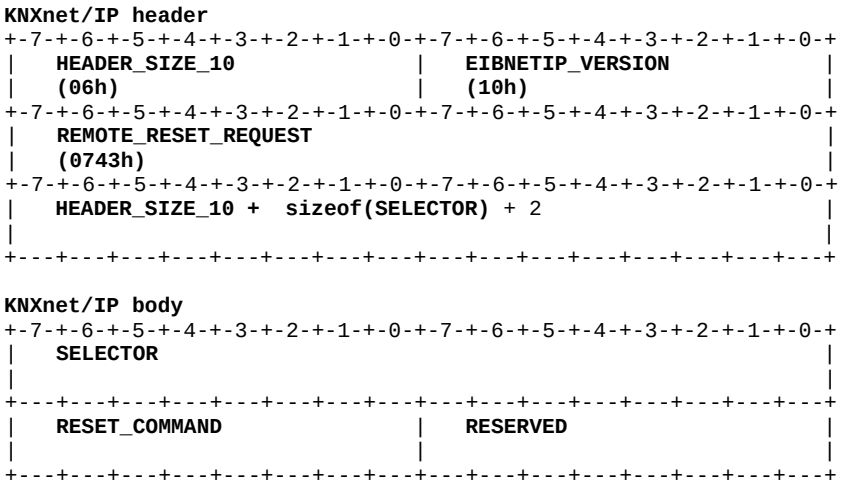


Figure 4 – REMOTE_RESET_REQUEST frame binary format

4.5 Description Information Block (DIB)

4.5.1 Introduction

The Description Information Block (DIB) shall be a set of data accessed via remote diagnosis and configuration services. While the Core services for device discovery and device description only allow reading DIBs, the Remote Diagnosis and Configuration DIBs allow reading data from the DIBs and writing data to them.

4.5.2 DIB description type codes

Description Information Blocks (DIB) are defined in [03] clause 7.5.4. Table 1 in that clause lists the description type codes.

KNXnet/IP Remote Diagnosis and Configuration shall use the following Description type codes:

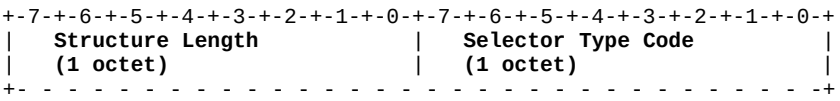
- IP_CONFIG, and
- IP_CUR_CONFIG, and
- KNX_ADDRESSES.

4.6 SELECTOR

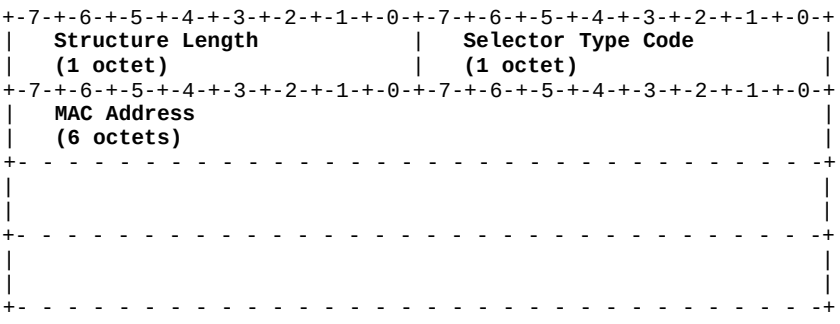
As the datagrams are transmitted via multicast all KNXnet/IP devices receive the remote diagnosis services in parallel. A Selector is defined to allow for selecting a specific device via MAC address or one or more devices in Programming Mode.

Description type	Value	Description
PrgMode Selector	01h	selection of devices in Programming Mode
MAC Selector	02h	selection of a device via MAC address

4.6.1 PrgMode Selector



4.6.2 MAC Selector



4.7 RESET COMMAND

The reset command shall be picked from an enumeration.

The reset shall be executed immediately after receiving the command.

Description type	Value	Description
Restart	01h	The device is restarted.
Master Reset	02h	The device is reset to factory default settings and then restarted.

5 Binary examples of KNXnet/IP frames

5.1 REMOTE_DIAGNOSTIC_REQUEST

			- - - - KNXnet/IP header - - - -
1	**06h**		header size
2	**10h**		protocol version
3	**07h**	\	
4	**40h**	>	service type identifier 0740h
5	**00h**	\	
6	**10h**	>	total length, 16 octets
		/	
			- - - - HPAI - - - -
7	**08h**		structure length of HPAI
8	**01h**		host protocol code, e.g. 01h, for UDP over IPv4
9	**E0h**	\	
10	**00h**	|	
11	**17h**	>	IP multicast address e.g. 224.0.23.12 (System Routing Multicast Address)
12	**0Ch**	/	
13	**0Eh**	\	
14	**57h**	>	port number of control endpoint, 3671
		/	
			- - - - SELECTOR - - - -
15	**02h**		structure length of SELECTOR
16	**01h**		Programming Mode Selector

Figure 5 – REMOTE_DIAGNOSTIC_REQUEST frame binary format: example

5.2 REMOTE_DIAGNOSTIC_RESPONSE

	+-------------------------------+	**- - - - KNXnet/IP header - - - -**
1	| **06h** |	header size
	+-------------------------------+	
2	| **10h** |	protocol version
	+-------------------------------+	
3	| **07h** |	\
	+- - - - - - - - - - - - - - -+	> service type identifier 0741h
4	| **41h** |	/
	+-------------------------------+	
5	| **00h** |	\
	+- - - - - - - - - - - - - - -+	> total length, 58 octets
6	| **3Ah** |	/
	+-------------------------------+	**- - - - SELECTOR - - - -**
7	| **02h** |	structure length of SELECTOR
	+-------------------------------+	
8	| **01h** |	Programming Mode Selector
	+-------------------------------+	**- - - - DIB IP Config - - - -**
9	| **10h** |	structure length of DIB IP Config
	+-------------------------------+	
10	| **03h** |	Description Type Code
	+-------------------------------+	
11	| **C0h** |	\
	+- - - - - - - - - - - - - - -+	|
12	| **A8h** |	|
	+- - - - - - - - - - - - - - -+	> IP address
13	| **02h** |	| e.g. 192.168.2.12
	+- - - - - - - - - - - - - - -+	|
14	| **0Ch** |	/
	+-------------------------------+	
15	| **FFh** |	\
	+- - - - - - - - - - - - - - -+	|
16	| **FFh** |	|
	+- - - - - - - - - - - - - - -+	> subnet mask
17	| **FFh** |	| e.g. 255.255.255.0
	+- - - - - - - - - - - - - - -+	|
18	| **00h** |	/
	+-------------------------------+	
19	| **C0h** |	\
	+- - - - - - - - - - - - - - -+	|
20	| **A8h** |	|
	+- - - - - - - - - - - - - - -+	> default gateway IP address
21	| **02h** |	| e.g. 192.168.2.1
	+- - - - - - - - - - - - - - -+	|
22	| **01h** |	/
	+-------------------------------+	
23	| **02h** |	IP capabilities (e.g. DHCP)
	+-------------------------------+	
24	| **01h** |	IP assignment method (e.g. manually)
	+-------------------------------+	**- - - - DIB IP Current Config - - - -**
25	| **14h** |	structure length of DIB IP Current Config
	+-------------------------------+	
26	| **04h** |	Description Type Code
	+-------------------------------+	
27	| **C0h** |	\
	+- - - - - - - - - - - - - - -+	|
28	| **A8h** |	|
	+- - - - - - - - - - - - - - -+	> IP address
29	| **02h** |	| e.g. 192.168.2.12
	+- - - - - - - - - - - - - - -+	|
30	| **0Ch** |	/
	+-------------------------------+	
31	| **FFh** |	\
	+- - - - - - - - - - - - - - -+	|
32	| **FFh** |	|
	+- - - - - - - - - - - - - - -+	> subnet mask
33	| **FFh** |	| e.g. 255.255.255.0
	+- - - - - - - - - - - - - - -+	|
34	| **00h** |	/
	+-------------------------------+	
35	| **C0h** |	\
	+- - - - - - - - - - - - - - -+	|
36	| **A8h** |	|
	+- - - - - - - - - - - - - - -+	> default gateway IP address
37	| **02h** |	| e.g. 192.168.2.1
	+- - - - - - - - - - - - - - -+	|

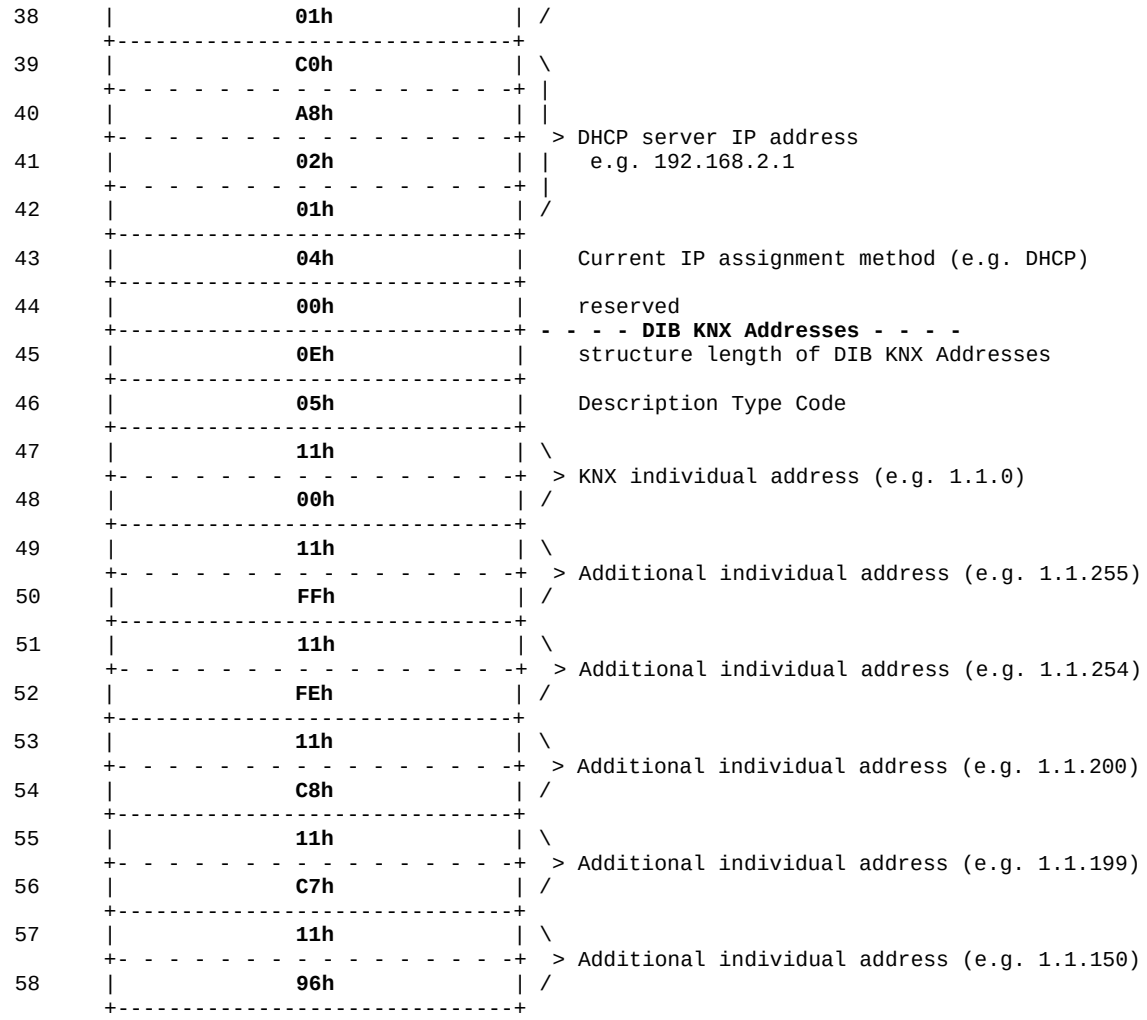


Figure 6 – REMOTE_DIAGNOSTIC_RESPONSE frame binary format: example

5.3 REMOTE_BASIC_CONFIGURATION_REQUEST

			- - - - KNXnet/IP header - - - -
1	**06h**		header size
2	**10h**		protocol version
3	**07h**	\	
4	**42h**	/	> service type identifier 0742h
5	**00h**	\	
6	**20h**	/	> total length, 32 octets
			- - - - HPAI - - - -
7	**08h**		structure length of HPAI
8	**01h**		host protocol code, e.g. 01h, for UDP over IPv4
9	**E0h**	\	
10	**00h**	|	
11	**17h**	|	> IP multicast address e.g. 224.0.23.12 (System Routing Multicast Address)
12	**0Ch**	/	
13	**0Eh**	\	
14	**57h**	/	> port number of control endpoint, 3671
			- - - - SELECTOR - - - -
15	**02h**		structure length of SELECTOR
16	**01h**		Programming Mode Selector
			- - - - DIB IP Config - - - -
17	**10h**		structure length of DIB IP Config
18	**03h**		Description Type Code
19	**C0h**	\	
20	**A8h**	|	
21	**03h**	|	> IP address e.g. 192.168.3.12
22	**0Ch**	/	
23	**FFh**	\	
24	**FFh**	|	
25	**FFh**	|	> subnet mask e.g. 255.255.255.0
26	**00h**	/	
27	**C0h**	\	
28	**A8h**	|	
29	**03h**	|	> default gateway IP address e.g. 192.168.3.1
30	**01h**	/	
31	**00h**		IP capabilities (not writable → 00h)
32	**01h**		IP assignment method (e.g. manually)

Figure 7 – REMOTE_BASIC_CONFIGURATION_REQUEST frame binary format: example

5.4 REMOTE_RESET_REQUEST

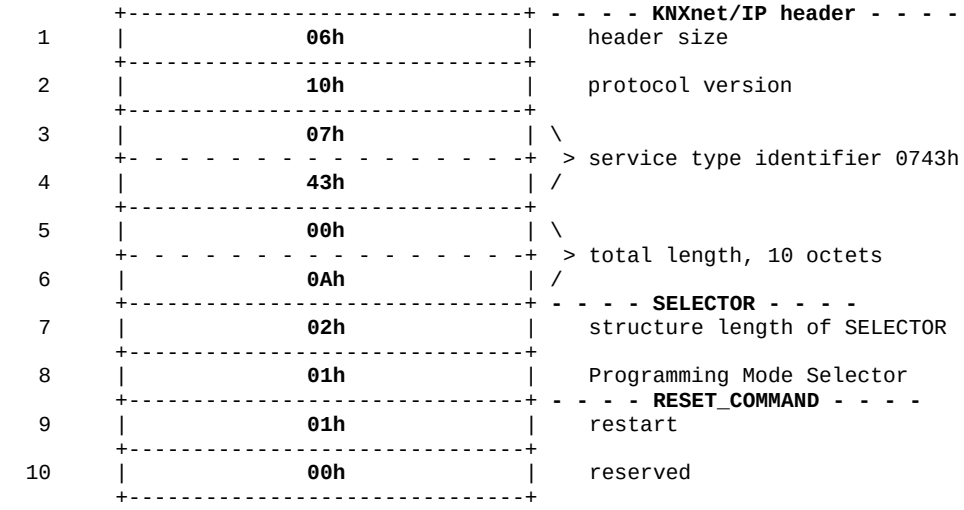


Figure 8 – REMOTE_RESET_REQUEST frame binary format: example

6 Certification

6.1 Introduction

This clause provides information on the test procedures and requirements of the certification process.

6.2 Support matrix

Service name	sent from ... to ...	implementation is
REMOTE_DIAGNOSTIC_REQUEST	Client → Server	M
REMOTE_DIAGNOSTIC_RESPONSE	Server → Client	M
REMOTE_BASIC_CONFIGURATION_REQUEST	Client → Server	M
REMOTE_RESET_REQUEST	Client → Server	M

Legend: “M” = Mandatory, “O” = Optional, “n.a.” = not applicable