Application Note 136/10 v01

Title:	**A_NetworkParameter_Read for PID_OBJECT_TYPE in Coupler**	
Status:		**Date:**
	Draft Proposal	2010.07.07
Transitional period:	Immediate effect after Final Voting.	
Date:	2010.07.07	

Subject: Specification of the reaction of the Coupler on reception of an A_NetworkParameter_Read-PDU with parameter_type = Device Object + PID_OBJECT_TYPE

Documents

Modified

[1] Chapter 3/5/1 “Resources” v1.2 of 2009.06.24.
[2] Volume 6 “Profiles” v1.9.00 of 2009.10.22

Referred

[3] Chapter 3/3/2 “Data Link Layer General” v1.2 of 2009.06.29
[4] Chapter 3/3/3 “Network Layer” v1.1 AS of 2008.12.22
[5] Chapter 3/3/7 “Application Layer” v1.1 AS of 2008.12.21
[6] Part 9/3 “Couplers” v1.1 of 2009.06.19

Document updates

Version	Date	Modifications
KSG466-01	2010.06.15	• Document creation.
KSG466-02	2010.07.07	• Update according the feedback of the KSG meeting of 2010.06.29-30. - Router Object “not allowed” for other Profiles than the Coupler Profiles.
AN136 v01	2010.07.07	• Creation of the Draft Proposal.

Contents

1 Purpose, motivation and scope

This clause is not intended for integration in the KNX Specifications.

To check for the presence of a Coupler in a Subnetwork, the service A_NetworkParameter_-Read can be used, with the service parameter parameter_type encoded as follows: object_type = Router Object = 6 and PID = PID_OBJECT_TYPE = 1. However, it is nowhere in the KNX Specifications formally specified how the Coupler shall react on this.

As a reminder, the A_NetworkParameter_Read-PDU is in [5] specified as in Figure 1.

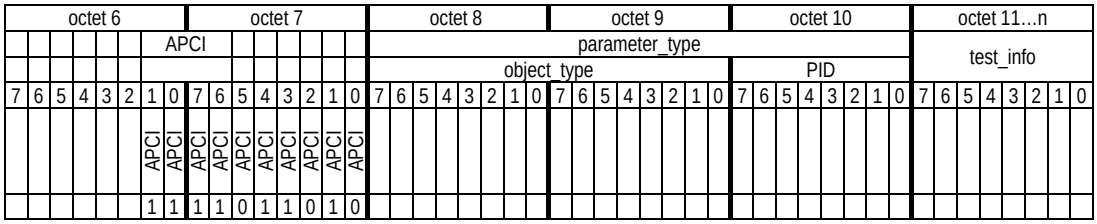

Figure 1 - A_NetworkParameter_Read-PDU (example)

The A_NetworkParameter_Response-PDU is specified as in Figure 2.

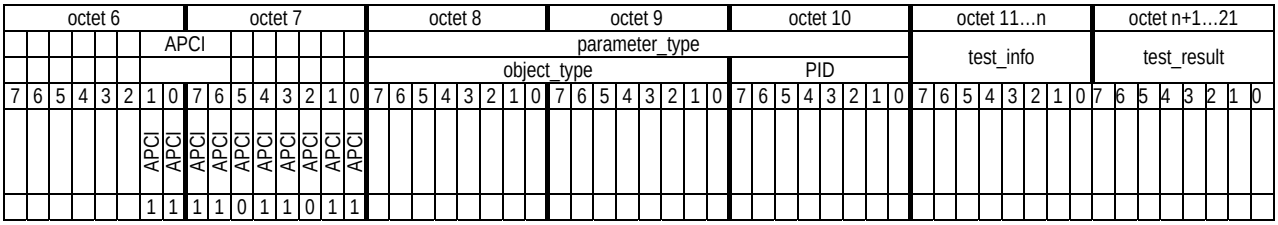

Figure 2 - A_NetworkParameter_Response-PDU (example)

The A_NetworkParameter_Read-service has been designed to prevent from the many new network management operations requiring as many new APCIs. This goal is reached, but it requires that for each use of A_NetworkParameter_Read, thus for each Object Type and for each Property, the reaction is to be specified if needed.

This is specified in the following for PID_OBJECT_TYPE in the Router Object.

For the use of PID_OBJECT_TYPE in other Object Types and for other Properties this specification does not apply.

2 Specification

2.1 Terms and definitions

This document does not introduce neither modify any terms or definitions

2.2 Stack and communication

This document does not introduce neither modify any stack or communication mechanisms.

2.3 Resource definition or used Resources

This clause shall be inserted as new clause 4.4.2 in [1]. The existing clause 4.4.2 in [1] shall be shifted down.

2.3.1 PID_OBJECT_TYPE (PID = 1)

2.3.1.1 General requirements

Please refer to clause 4.2.1 in [1] for the general requirements of PID_OBJECT_TYPE.

This Property shall contain Object Type of the Interface Object. The Router Object shall have Object Type 0006h.

2.3.1.2 Usage by the Management Server (device)

PID_OBJECT_TYPE in the Router Object shall support the Data Property services as specified in clause 3.4.3 of [5] without restrictions.

The Coupler shall additionally support the NM_NetworkParameter_Read_R as follows.

On reception of an A_NetworkParameter_Read-PDU with Object Type = Router Object and PID = PID_OBJECT_TYPE and test_info = 00h, the Coupler shall respond with an A_NetworkParameter_Response-PDU with the same service parameter values and with test_result equal to the Property Value, this is, the Interface Object Type.

This response shall be sent in broadcast communication mode.

The Coupler is in this case addressed itself. Therefore, its handling of the hop_count_type shall be as specified in [4] clause 2.4.1 “State Machine of Network Layer for normal devices”, this is, when handling a service indication primitive with hop_count_type equal to 7, respectively different from 7, it shall in the response primitive use the hop_count_type 7 respectively “Network Layer Parameter”.

The response shall be transmitted on the Coupler side (primary or secondary) on which the original request has arrived.

On reception of an A_NetworkParameter_Read-PDU with Object Type = Router Object and PID = PID_OBJECT_TYPE and test_info ≠ 00h, the Coupler shall not react.

Error and exception handling

- The above applies for any Coupler regardless of the Coupler model (see clause 6.3 in [3]: KNX Router, KNX TP1 Bridge or KNX TP1 Repeater). The Management Client thus needs to interpret the Individual Addresses of the responses to conclude on the use of the present Couplers.

2.3.1.3 Usage by the Management Client

The Management Client may read PID_OBJECT_TYPE with A_NetworkParameter_Read with hop_count value 0 –to find the nearest Couplers – as well as with value 7.

2.4 Management Procedures

This document does not introduce neither modify any Management Procedures.

2.5 Configuration Procedures

This document does not introduce neither modify any Configuration Procedures.

2.6 Usage and context

This clause is not intended for integration in the KNX Specifications.

This mechanism shall be used to find the presence of a Coupler in a Subnetwork.

2.7 Profile definition

In the Profiles ([2]), it is already specified in clause A.3.13 that for masks 0912h and 091Ah, the A_NetworkParameter_Read shall be supported for PID_OBJECT_TYPE in the Router Object, so no change is needed.

If a KNX feature is not mandatory for any Profile, it is typically marked as optional. If however any random KNX device would implement a Router Object with this functionality for PID_OBJECT_TYPE, this would falsify the expected outcome. Therefore, the Router Object is made "not allowed" in Volume 6 for other Profiles than mask 0912h and 091Ah.

Clause 8.2.3.4 in [2] shall be updated as follows. (Original text is in black. Insertions are in green. Changed contents are in red.)

	Type 1
…	O
…	O
9. Router Object	X
10. LTE Address Filter Table Object	O
11. …	

A_NetworkParameter_Read for PID_OBJECT_TYPE in Coupler

Clause A.2.4 in [2] shall be updated as follows. (Original text is in black. Insertions are in green. Changed contents is in red.)

S-Mode Profiles → End-devices → Interface Objects

Interface Object	**System 2**	mask 0020h	mask 0021h	mask 0025h	**System 300**	mask 0300h	mask 2300h	**System 7**	mask 0701h	mask 0705h	**System B**	mask 07B0h	mask 17B0h	mask 2010h
0 Device Object	M	M	M	M	M	M	M	O	M	M	M	M	M	M
1 Addresstable Object	M	M	M	M	M	M	M	O	M	M	M	M	M	O
2 Association Table Object	M	M	M	M	M	M	M	O	M	M	M	M	M	O
3 Applicationprogram Object	M	M	M	M	M	M	M	O	M	M	M	M	M	O
6 Router Object	X	X	X	X	X	X	X	X	X	X	X	X	X	X
9 Group Object Table Object	O	O	O	O	M	M	M	O	O	O	M	M	M	O

Clause A.3.11in [2] shall be updated as follows. (Original text is in black. Insertions are in green. Changed contents is in red.)

S-Mode Profiles → Interface and Couplers → Interface Objects

Interface Object	mask 0912h	mask 1900h	mask 091Ah	RF retransmitter	RF media coupler	**cEMI server**	cEMI server on TP	cEMI server RF asynchr.	cEMI server asynch. & BiBat
0 Device Object	M	O	M	?	?	O	M	M	M
1 Addresstable Object	O	O	O	?	?	O	O	O	O
2 Associationtable Object	O	O	O	?	?	O	O	O	O
3 Applicationprogram Object	O	O	O	?	?	O	O	O	O
4 Interfaceprogram Object	O	O	O	?	?	O	O	O	O
6 Router Object	M	O	M	X	X	X	X	X	X
7 LTE Address Routing Table Object	M	O	M	?	?	O	O	O	O
8 cEMI Server Object	O	O	O	?	?	M	M	M	M
9 Group Object Table Object	O	O	O	?	?	O	O	O	O
10 Polling Master	O	O	O	?	?	O	O	O	O
11 KNXnet/IP Parameter Object [1]	O	O	M	O	O	O	O	O	O

1) The KNXnet/IP Parameter Object is mandatory if there is a KNXnet/IP interface.

2.8 Identifiers and discovery

This clause is not intended for integration in the KNX Specifications.

This document does not introduce neither modify any new Identifiers. The identifiers for the Router Object (Object Type = 6) and the property Object Type (PID = 1) exist already.

The support of this feature cannot be discovered: it is mandatory for the masks 0912h and 091Ah.

3 Impact and dependencies

3.1 System specification (“Handbook”) dependencies

This clause is not intended for integration in the KNX Specifications.

Indications for integration in the KNX Specifications are given at the appropriate locations in the above. Only the Management Server specification for PID_OBJECT_TYPE in the Router Object in [1] needs to be extended.

3.2 Configuration interworking

This clause is not intended for integration in the KNX Specifications.

This document does not modify or introduce any Configuration interworking: it only formalizes the already existing behaviour of the existing implementations of mask 0912h and 091Ah.

3.3 Runtime Interworking

This clause is not intended for integration in the KNX Specifications.

This document focuses on network configuration functionality and is not related to runtime Interworking.

3.4 Registration and certification

This clause is not intended for integration in the KNX Specifications.

The tests for this functionality are already specified in [6] clause 4.7.5.2.

3.5 Integration and common tool impact

It is unclear in this version of the document whether ETS makes use of the reading of PID_OBJECT_TYPE of the Router Object in the Couplers. This could be helpful in resolving problems with the topology or with the local Individual Address. This is being checked.

3.6 Risks and compatibility issues

Risks have been considered and have been resolved.