Application Note 160/13 v01

Title: **RF S-Mode Device Profiles**

Status: **Date:**

Draft Proposal 2013.05.07

Transitional period: Immediate effect after Final Voting.

Subject: Specification of Resources and Device Profiles for end devices, interfaces and Media Couplers of KNX RF devices for S-Mode configuration.

Documents

Modified

[01] [01] Chapter 3/1/2 “Glossary”
[02] [02] Chapter 3/2/5 “Radio Frequency” v1.06.01 of 2012.04.25
[03] [03] Chapter 3/5/1 “Resources”
[04] [04] Chapter 3/7/3 “Standard Identifier Tables”
[05] [05] Volume 6 “Profiles”

Referred

[06] [06] Chapter 3/3/1 “Physical Layer General”
[07] [07] Chapter 3/3/2 “Data Link Layer General”
[08] [08] Chapter 3/3/3 “Network Layer”
[09] [09] Chapter 3/3/4 “Transport Layer”
[10] [10] Chapter 3/3/7 “Application Layer” v1.3.00 AS of 2010.10.22
[11] [11] Chapter 3/4/1 “Application Interface Layer””
[12] [12] Chapter 3/5/2 “Management Procedures”
[13] [13] Chapter 3/6/3 “External Message Interface”
[14] [14] Chapter 8/2/5 “RF Physical and Link Layer Tests”
[15] [15] Chapter 8/3/3 “Network Layer Tests”
[16] [16] Chapter 8/3/4 “Transport Layer Tests”
[17] [17] Chapter 8/3/7 “Application (Interface) Layer Testing – Network Management Server/Client Testing”
[18] [18] Part 8/7 “Interworking and Functionality Tests”
[19] [19] AN134 “Flexible E-Mode Channels”
[20] [20] AN151 “cEMI AddInfo for KNX RF Multi and new Properties”
[21] [21] AN158 “KNX Data Security”
[22] KSG475-01 “Using Domain Address for Group Telegrams on RF”
[23] [23] KSG501 “Coupler Model 2.0”
[24] [24] KSG502 “System aspects of the RF integration in ETS”
[25] [25] KSG528 “System 7 Resources”

Document updates

Version	Date	Modifications
KSG495-23	2013.04.04	- Included common stack for Profile “DoA based RF Retransmitter”.

Version	Date	Modifications
KSG495-24	2013.04.25	• Update according feedback of KSG meeting of 2013.04.19. - Usage of SN or DoA for group communication indicated in the Profiles. - DoA 0 is not a valid DoA and shall not be assigned by ETS. - Programming Mode times out after 4 minutes. - NM_SubnetworkDevices_Scan2 added (“Deep Scan”) - PID_TABLE_REFERENCE is mandatory for the Group Object Association Table and the Group Object Table for mask 2705h and 27B0h. - Details of the Profiles corrected. - Included further comments from KSG of 2013.04.19. - Added PID_RF_DIAG_PROBE
AN160 v01	2013.05.07	• Creation of the Draft Proposal.

Contents

1 Purpose, motivation and scope (informative) 4
1.1 Introduction 4
1.2 Overview 4
1.3 Motivation 4
1.3.1 Why shall the support of KNX RF in ETS be based on a new KNX RF Device Profile and why is not any of the existing Device Profiles for KNX RF used? 4
1.3.2 Why are unidirectional RF devices not supported? 5
1.4 Scope 5
1.4.1 KNX RF systems 5
1.4.2 KNX TP1/RF Media Coupler 5
1.4.3 E-Mode Profiles and combinations with KNX RF S-Mode Profiles 5
1.4.4 Existing KNX RF devices 5
1.4.5 Other topics and documents related to KNX RF S-Mode devices 6
1.5 Main principle 6
1.5.1 No longer supported 6
2 Specification (normative) 7
2.1 Terms and definitions 7
2.2 Stack and communication 7
2.2.1 Communication Medium RF 7
2.3 Resource definition or used Resources 11
2.3.1 DoA Realisation Type 2 – DoA on 6 octets 11
2.3.2 RF Medium Object (Object Type 19) 11
2.3.3 cEMI Server Object 20
2.3.4 Programming Mode 20
2.4 Management Procedures 21
2.4.1 Introduction 21

2.4.2 NM_SubnetworkDevices_Scan2 21
2.4.3 NMP_LinkBudget_Measure 23
2.5 Configuration Procedures 25
2.5.1 Network Configuration Procedures 25
2.5.2 Device Configuration Procedures 26
2.6 External Message Interface 26
2.7 Profile definition 26
2.7.1 Profile: mask 2705h 26
2.7.2 Profile: mask 27B0h 53
2.7.3 Profile: Domain Address based RF Retransmitter 79
2.7.4 KNX RF USB Interface 1 81
2.7.5 Not more supported 81
2.8 RF Multi device handling by ETS 81
2.9 Identifiers and discovery 81

3 Impact and dependencies 82
3.1 System specification (“Handbook”) dependencies 82
3.2 On Push-Button and link services 82
3.2.1 Introduction 82
3.2.2 E-Mode Channel Object – PID_OBJECTLINK (PID = 63) 82
3.3 Configuration interworking 85
3.4 Run-time Interworking 85
3.5 Registration and certification 85
3.6 Integration and common tool impact 85
3.6.1 Programming Mode 85
3.6.2 Domain Address management 85
3.6.3 Interfaces 87
3.6.4 Domain Address based KNX RF Retransmitter 88
3.6.5 Configuration Procedures for mask 27B0h 88
3.6.6 Common requirements for all KNX RF end device Profiles and – interfaces 88
3.6.7 ETS Database 89
3.6.8 Integration of RF Media Coupler 89
3.6.9 Integration of RF interface 89
3.6.10 Integration of RF devices 89
3.6.11 End device or TP1/RF Media Coupler with Retransmitter function 89
3.6.12 Diagnostics 90
3.7 Risks and compatibility issues 90

1 Purpose, motivation and scope (informative)

1.1 Introduction

This document specifies KNX device Profiles for S-Mode Configuration of RF devices.

This proposal defines a solution for an easy integration of KNX RF in ETS. It keeps the threshold low for existing ETS user by not introducing new installer procedures.

1.2 Overview

The KNX RF Communication Medium defines four systems, as specified in Table 1 “Guide for compliance” in [02].

Table 1 – KNX RF Systems (informative copy)

System	Physical Layer	Data Link Layer
KNX RF Ready	clause 5.1 in [02]	clause 6.1 and clause 6.2 in [02]
KNX RF Multi	clause 5.2 in [02]	clause 6.1 and clause 6.6 in [02]
KNX RF BiBat	clause 5.1 in [02]	clause 6.1, clause 6.2 and clause 6.3 in [02]
KNX RF BiBat 2	clause 5.3 in [02]	clause 6.1, clause 6.4 and clause 6.5 in [02]

The configuration of RF devices is until today (March 2013) done by one of the following.

- E-Mode Push Button Configuration (PB SEC)
- E-Mode Controller Mode
- S-Mode configuration via plug-ins in the ETS software based on Ctrl-Mode Management Procedures.

All these 3 Configuration Modes base on E-Mode Channels and base on Link services (see clause 3.4.4 in [10]).

1.3 Motivation

1.3.1 Why shall the support of KNX RF in ETS be based on a new KNX RF Device Profile and why is not any of the existing Device Profiles for KNX RF used?

The support of the existing KNX RF devices and the E-Mode Management – and Configuration Procedures is not seen as an optimal solution: ETS is not able to natively commissioning these devices. It would require a large effort by ETS to support the following.

- ETS has to handle the KNX Serial Numbers of all devices.
- The modification of one Group Address may require the update of several devices.
- A new KNX Profiles would need to be established, with Configuration Procedures using Link Services and – Properties, assignment of the Domain Address.
- ETS would need to access in some way all involved devices, to obtain their KNX Serial Numbers, before it would be able to calculate and download any link.

1.3.2 Why are unidirectional RF devices not supported?

State-of-the-art RF solutions are no more unidirectional and thus it is possible to set up some configuration in the devices. Device can be bidirectional during configuration even if in runtime they are sending only.

1.4 Scope

1.4.1 KNX RF systems

The goal of this document is to specify S-Mode Device Profiles for the KNX RF Communication Medium. This specified in [02] and exhibits four systems, as Table 1. From these, the scope of this document is limited to the system “KNX RF Ready”.

The system “KNX RF Multi” is not within the scope of this document.

- It exhibits the feature Fast Acknowledgement of which it is not specified how it shall be handled by the KNX TP1/RF Media Coupler and how it shall be configured by ETS.
- It also exhibits the feature of multiple RF Channels, so-called Slow – and Fast Channels, of which it is not specified how the sending shall be configured by ETS and how the runtime behaviour shall be by the KNX TP1/RF Media Coupler and by ETS.

KNX RF unidirectional systems are not in the scope of this document either.

The support of KNX RF S-Mode devices will therefore be done in two steps..

1. STEP 1: limited to KNX RF Ready.
   If ETS configures a KNX RF Multi device, it will configure it to behave like a KNX RF Ready device without any user action. (See PID_RF_MULTI_TYPE in 2.3.2.3.)
2. STEP 2: extend to KNX RF Multi.
   In this step, all above mentioned features of KNX RF Multi will be supported.

1.4.2 KNX TP1/RF Media Coupler

This document does not specify the standard KNX TP1/RF Media Coupler. This is specified in the document [23].

1.4.3 E-Mode Profiles and combinations with KNX RF S-Mode Profiles

As the scope of this document is limited to S-Mode, specifications for combined Profile of S-Mode and Ctrl-Mode that may require additional mechanism to protect or lock during linking and E-Mode Management Procedures are not part of this document.

1.4.4 Existing KNX RF devices

Existing KNX RF devices, based on other Profiles, not specified in this document, can be supported through plug-ins in the ETS software.

1.4.5 Other topics and documents related to KNX RF S-Mode devices

Topic:	**KNX TP1/RF Media Coupler**
Description:	A standard KNX TP1/RF Media Coupler is specified. This based on the Coupler Model 2.0.
Status:	See [23].

Topic:	**Services adapted and introduced on the occasion of KNX RF S-Mode devices**
Description:	Mainly for the KNX RF Network Configuration Procedures (discovery), some AL-services need to be modified and need to be introduced.
Status:	See [24].

Topic:	Clarification of AET indication in cEMI
Description:	Verification of consistent encoding of broadcast/system broadcast flags in RFn cEMI and PL110.
Status:	Resolved in KSG meeting of 2012.09.25: no contradictions.

1.5 Main principle

Domain Address for group communication on KNX RF

As KNX RF Telegrams are defined to natively carry only a KNX Extended Frame, the principle is to use the Domain Address to close the RF open medium. Within this domain, all the KNX system is working like on the KNX TP1 medium. This allows that group communication is used as on KNX TP1.

As for the KNX RF S-Mode devices the RF DoA and the Group Addresses are configured by ETS, it is guaranteed the combination is unique.

NOTE 1 This is not the case on PB-Mode, in which each devices uses the same pre-defined Group Addresses.

In this way, there is no need any more of additional association and KNX Serial Number table to ensure Group Communication on RF.

1.5.1 No longer supported

The following features are not supported by the RF S-Mode devices.

- This document specifies S-Mode Profiles for RF bidirectional devices. As the S-Mode Configuration Procedures are used, it is not necessary that PID_OBJECTLINK is supported. The reaction to an A_FunctionPropertyCommand-service shall be the standard error handling: the MaS shall reply with an A_FunctionPropertyState_Response-PDU without the field return_code (this is, the returned PDU shall not contain the field return_code) and without the field data (this is, the returned PDU shall not contain the field data).

2 Specification (normative)

2.1 Terms and definitions

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

This document does not introduce – neither modify any terms or definitions or abbreviations.

2.2 Stack and communication

2.2.1 Communication Medium RF

2.2.1.1 Extended Group Address

In Chapter 3/2/5 "Communication Medium RF" ([02]) clause 6.1.1.1, the following original text shall be replaced.

"The RF frame shall contain the KNX Serial Number of the sender for the following communication modes:

- *point-to-multipoint, connectionless (multicast) and*
- *point-to-system, connectionless (system broadcast).*

This shall be indicated by the value 0 of the field AddrExtensionType in the second block of the RF frame. Multicast datagrams received with the wrong value of the AddrExtensionType shall be discarded by the receiving Data Link Layer instance."

This shall be replaced by the following.

The KNX RF Frame shall contain the KNX Serial Number - or the RF Domain Address of the sender according Table 2. The sender shall indicate the used value (KNX Serial Number or RF Domain Address) with the value of the field Address Extension Type (AET) as also indicated in Table 2.

Table 2 – Use of KNX Serial Number or RF Domain Address

communication mode	KNX Serial Number or RF Domain Address	
	KNX Serial Number of the sender	RF Domain Address of the sender
point-to-system, connectionless (system broadcast)	yes	no
point-to-*domain*, connectionless (broadcast)	no	yes
point-to-multipoint, connectionless (multicast)	yes [a]	yes [a]
point-to-point, connectionless	no	yes
point-to-point, connection-oriented	no	yes
Address Extension Type (=AddrExtensionType) (AET)	0	1

[a] Please refer to the Profile specifications for which type of KNX RF device shall use KNX Serial Number or DoA.

2.2.1.2 RF Domain Address

In Chapter 3/2/5 "Communication Medium RF" ([02]) clause 6.1.1.4, the following original text shall be replaced.

"The RF Domain Address shall be a 6 octet number. The RF Domain Address in an RF installation shall always be identical to the KNX Serial Number of one of the devices in the installation. This shall guarantee that the RF Domain Address is a unique number.

The RF frame shall contain the RF Domain Address for the following communication modes:

- *point-to-point, connectionless,*
- *point-to-point, connection-oriented and*
- *point-to-all-points, connectionless (broadcast).*

This shall be indicated by the value 1 of the field AddrExtensionType in the second block of the RF Frame. Point to point connectionless and point to point connection oriented Telegrams received with the wrong value of the AddrExtensionType shall be discarded by the receiving Data Link Layer instance.

For other communication modes, the KNX Serial Number shall be used.

This shall be replaced by the following.

The RF Domain Address shall be a 6 octet number**.** It shall be guaranteed during the Configuration procedures that the RF Domain Address is a unique number.

The sender shall include the RF Domain Address in the RF Frame according Table 2. This shall be indicated by the value 1 of the field AddrExtensionType in the second block of the RF Frame.

The receiver shall discard the RF Frame of the field AddrExtensionType does not match with the used communication mode as given in Table 2.

2.2.1.3 RF Domain Address based RF Retransmitter

The existing specification of the KNX RF Retransmitter in the KNX RF Specification bases explicitly on the KNX Serial Number. In order for the newly intended RF Retransmitters to be KNX compliant, it is thus needed to be specified that these can base on the RF Domain Address as well. [20130327aa]

The existing clause 6.2.2.2 in [02] shall become clause 6.2.2.2.1 and be renamed to "KNX Serial Number based Retransmitter". The below text shall become a new clause, 6.2.2.2.2 and be named "RF Domain Address based RF Retransmitter".

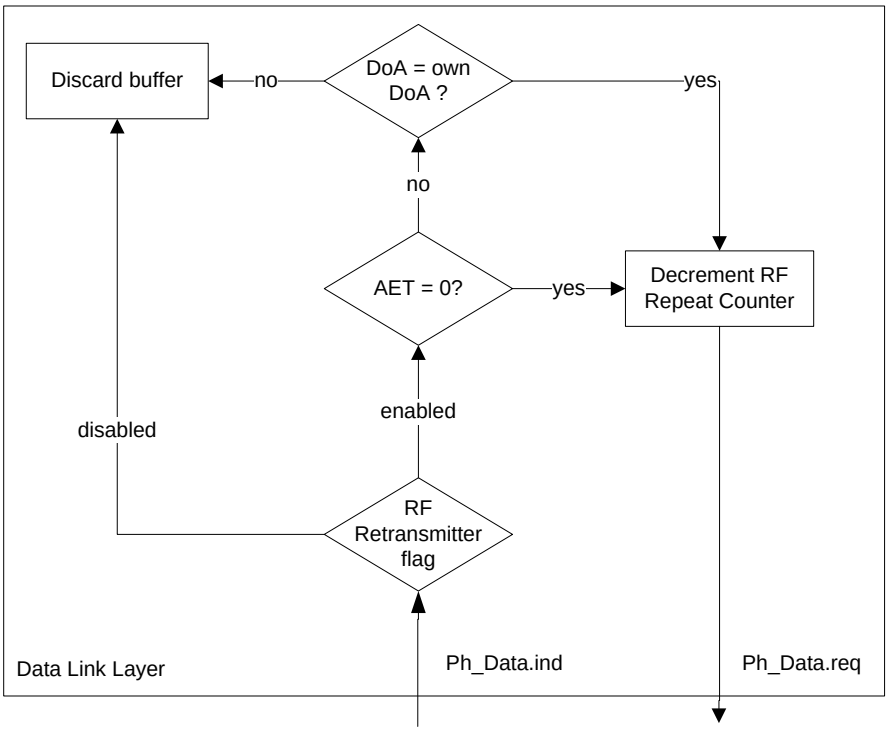


Figure 1 – RF Domain Address based RF Retransmitter (basic flowchart)

The RF Retransmitter shall retransmit all RF Telegrams with Extended Group Addresses, this is, with AET = 0 (as long as there is no filter for SN). This is, it shall retransmit the following.

- All Telegrams in point-to-system, connectionless communication mode (system broadcast).
- All Telegrams in point-to-multipoint, connectionless (multicast) communication mode using AET = 0.

If the RF Retransmitter has a configured RF Domain Address it shall additionally retransmit all RF Telegrams with AET = 1, if the contained RF DoA equals the own DoA of the device. This is, within the RF Domain:

- All Telegrams in point-to-multipoint, connectionless (multicast) communication mode using the RF DoA of the RF Retransmitter. This is normal group communication.
- All Telegrams in point-to-domain, connectionless (broadcast) communication mode.
- All Telegrams in point-to-point connectionless or – connection-oriented communication mode if the Destination Address does not equal the own Individual Address.

NOTE 2 This concerns exclusively the retransmission of Telegrams and does not concern the possible handling of Telegrams addressed to the RF device (system broadcast, broadcast, point-to-point – or group communication).

2.2.1.4 Semi-directional devices and bidirectional mode

This clause is intended to be added to Chapter 3/2/5 "Communication Medium RF" as new clause 6.7. By having this outside the specification of clause 6.2 "KNX RF Data Link Layer for KNX RF Ready and BiBat systems", this behaviour becomes in theory also available in the future for KNX RF Multi devices.

Semi-directional KNX RF devices are bidirectional RF devices that however at runtime only send RF Telegrams and do not received RF Telegrams.

EXAMPLE 1 Devices that are powered by battery or by solar cells, for example in a window contact

Bidirectional Mode

Mainly in order to be configurable by the S-Mode MaC, to support the Network – and Device Management Procedures, semi-directional devices shall have a Bidirectional Mode as specified in Table 3.

Table 3 – Definition of Bidirectional Mode

Bidirectional Mode	Requirements
disabled	The semi-directional device does listen to the KNX RF Medium to receive any KNX RF Telegram.
enabled	The semi-directional device shall listen to the KNX RF Medium and shall receive any proper KNX RF Telegram addressed to it, as a full bidirectional device.

The Bidirectional Mode may be limited in time and shall be started and maintained according the below indications.

Semi-directional devices shall exhibit full bidirectional communication on the default call channel (F1).

Starting the bidirectional mode

The bidirectional mode shall be started and the – time-out timer shall be initiated according the following.

- When the Programming Mode is activated in the device.
  The bidirectional mode time-out timer shall in this case be initiated with a value of 1 minute.
  Inactivation of the Programming Mode shall have no effect on the bidirectional mode.
- When the device has restarted.
  The bidirectional mode time-out timer shall in this case be initiated with a value of 10 seconds.
- Bidirectional mode may optionally be activated by implementation specific means.

Retriggering the bidirectional mode

The MaS shall retrigger the bidirectional time-out timer on any of the following conditions.

- If the Property *Bidirectional Mode Time-out* (PID_RF_BIDIR_TIMEOUT; see 2.3.2.8) is written. If the value contained in the command differs from 0, then the MaS shall retrigger the bidirectional mode time-out timer with the given value.
- If the MaS is addressed by any message in point-to-point connectionless or – connection-oriented communication mode. The MaS shall in this case retrigger the bidirectional mode time-out timer with the value last set in PID_RF_BIDIR_TIMEOUT.

NOTE 3 This shall be done regardless of the Source Address of this point-to-point Telegram.

Inactivating the bidirectional mode

The MaS shall inactivate the bidirectional mode on any of the following conditions.

- If the bidirectional mode time-out timer expires.
- If the Property *Bidirectional Mode Time-out* (PID_RF_BIDIR_TIMEOUT; see 2.3.2.8) is written with the value 0.

2.3 Resource definition or used Resources

2.3.1 DoA Realisation Type 2 – DoA on 6 octets

Default DoA 000000000000h can't be sent by any RF interface from ETS, because this default value is used on the cEMI to indicate that the interface has to insert his own DoA in the RF Telegram.
in clause 3.2.4 "DoA Realisation Type 2 – DoA on 6 octets" the following sentence shall be replaced.

> *If unconfigured (ex-factory) the Domain Address is void and shall have the default value 000000000000h.*

This shall be replaced by the following.

If unconfigured (ex-factory) the Domain Address is void and shall have the default value FFFFFFFFFFFFh.

The value 000000000000h is not a valid DoA. The MaC shall not assign this value to any MaS.

2.3.2 RF Medium Object (Object Type 19)

2.3.2.1 Overview

This clause only gives an overview and is not intended for integration in the KNX Specifications. The Properties are specified in the below closes. The indications about the mandatory and optional Properties are given for the Profiles specified further in this paper.

- *mask 2705h* *clause 2.7.1.6.7*
- *mask 27B0h* *clause 2.7.2.6.8*
- *DoA based RF Retransmitter* *clause 2.7.3.4*
- *RF USB Interface 1* *clause 2.7.4*

The KNX RF Medium Object shall provide access to the following KNX RF features.

Property Identifier (PID)	Property Data Type	Value
1 PID_OBJECT_TYPE	PDT_UNSIGNED_INT	KNX RF Medium Object: Object Type = 19
51 PID_RF_MULTI_TYPE	PDT_GENERIC_01	RF ready and Multi capabilities
56 PID_RF_DOMAIN_ADDRESS	PDT_GENERIC_06	RF Domain Address
57 PID_RF_RETRANSMITTER	PDT_BINARY_INFORMATION	RF Retransmitter Flag
58 PID_RF_FILTERING_MODE_SUPPORT	PDT_BITSET8 (alt.: PDT_GENERIC_01)	RF filtering on KNX Serial Number or RF DoA
59 PID_RF_FILTERING_MODE_SELECT	PDT_ENUM8 (alt: PDT_UNSIGNED_CHAR)	RF Filtering Mode Selection
60 PID_RF_BIDIR_TIMEOUT	PDT_FUNCTION	Time-out for bidirectional communication.
61 PID_RF_DIAG_SA_FILTER_TABLE[]	PDT_GENERIC_03[]	Filter Table for Source Addresses for diagnostics of Link Budget.
62 PID_RF_DIAG_BUDGET_TABLE[]	PDT_GENERIC_04[]	Result of Link Budget diagnostics.
63 PID_RF_DIAG_PROBE	PDT_FUNCTION	Trigger and response for Link Budget diagnostic help telegrams.

2.3.2.2 PID_OBJECT_TYPE (PID: 1)

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

- **Property name:** Interface Object Type
- **Property Datatype:** PDT_UNSIGNED_INT
- **Datapoint Type:** DPT_PropDataType (DPT_ID = 7.010)

For the common specification of PID_OBJECT_TYPE, please refer to [03].

The Object Type of the KNX RF Medium Object shall be 19.

2.3.2.3 PID_RF_MULTI_TYPE (PID: 51)

This clause shall be added to [03].

- **Property name:** RF Multi type
- **Property Datatype:** PDT_GENERIC_01
- **Datapoint Type:** none

2.3.2.3.1 Abstract Resource definition

The Property *RF Multi Type* shall indicate and control whether the device is or operates as a KNX RF Ready device or as a KNX RF Multi device.

2.3.2.3.2 Format

Field	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
Name	reserved	reserved	reserved	reserved	reserved	reserved	reserved	RF-Multi Type
Values	0	0	0	0	0	0	0	0,1

Bit	Name	Description	Encoding
0	RF-Multi Type	*RF-Multi Type* shall indicate whether the KNX RF device is configured as “KNX RF Ready” type or as “KNX RF Multi” type.	0: KNX RF Multi Device set in KNX RF Ready type 1: KNX RF Multi Device set in KNX RF Multi type
1 to 7	reserved	These bits are reserved and shall always be 0.	not applicable

2.3.2.3.3 Usage by the Management Client

The MaC shall use the Property *RF Multi Type* to set the value of the KNX RF Type in the device.

The MaC shall only modify this parameter in a MaS that is in ex-factory state. This shall avoid any inconsistency in the device configuration.

EXAMPLE 2 If this Property is set in an already configured MaS, then this may lead to a mixture of links added in RF Multi type mode with Fast Ack management and links added in RF Ready type mode without Fast Ack management.

2.3.2.3.4 Usage by the Management Server

Please refer to clause 2.7 “Profile definition” for the mandatory and optional support of the Property *RF Multi Type*.

If the MaS is an RF Ready device, not supporting the RF Multi features, and the MaC writes this Property RF Multi Type, with any value, then the MaS shall always respond with RF-Multi Type = 0.

2.3.2.4 PID_RF_DOMAIN_ADDRESS (PID: 56)

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

- **Property name:** RF Domain Address
- **Property Datatype:** PDT_GENERIC_06
- **Datapoint Type:** None.

For the common specification of PID_RF_DOMAIN_ADDRESS, please refer to [03].

2.3.2.5 PID_RF_RETRANSMITTER (PID: 57)

- **Property name:** RF Retransmitter Flag
- **Property Datatype:** PDT_BINARY_INFORMATION
- **Datapoint Type:** DPT_Bool (DPT_ID: 1.003)

2.3.2.5.1 Abstract Resource definition

This Property shall control whether the RF Retransmitter function of the device is inactive or active.

2.3.2.5.2 Format

This shall be a 1 bit Property that shall be encoded as follows.

0: Disable: The KNX RF Retransmitter function in the MaS shall be disabled.

1: Enable: The KNX RF Retransmitter function in the MaS shall be enabled.

The default value shall be 0: Disable.

2.3.2.5.3 Usage by the Management Server (device)

The MaS shall enable and disable its KNX RF Retransmitter functionality according the setting of this *RF Retransmitter Flag*.

If the MaC sets *RF Retransmitter Flag* to “Enable” respectively “Disable”, then the MaS shall immediately activate respectively inactivate the retransmission RF Telegrams.

2.3.2.5.4 Usage by the Management Client

The MaC shall set this value according the selection by the installer.

2.3.2.6 PID_RF_FILTERING_MODE_-SUPPORT (PID: 58)

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

- **Property name:** Supported RF Filtering Modes
- **Property Datatype:** PDT_BITSET8 (alt.: PDT_GENERIC_01)
- **Datapoint Type:** None.

For the common specification of PID_RF_FILTERING_MODE_-SUPPORT, please refer to [03].

2.3.2.7 PID_RF_FILTERING_MODE_-SELECT (PID: 59)

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

- **Property name:** RF Filtering Mode Selection
- **Property Datatype:** PDT_ENUM8 (alt: PDT_UNSIGNED_CHAR)
- **Datapoint Type:** None.

For the common specification of PID_RF_FILTERING_MODE_SELECT, please refer to [03].

2.3.2.8 PID_RF_BIDIR_TIMEOUT (PID: 60)

- **Property name:** Bidirectional Mode Time-out
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** None

This Property *Bidirectional Mode Time-out* is designed for semi-directional devices. These devices shall enable bidirectional mode according 2.2.1.4. This Property shall allow the MaC to retrigger the time-out for the bidirectional mode and thus keep the bidirectional mode active in the MaS as long as necessary.

This Property shall not change the default time-out for bidirectional mode as specified in 2.2.1.4; it shall only retrigger or stop the bidirectional mode.

1) Write (A_FunctionPropertyCommand-PDU)

Octet 10
RF_Bidir_Time-Out (second)

RF_Bidir_TimeOut shall be the duration in seconds, from 1 s to 255 s, during which the RF bidirectional mode shall be active after reception of this Telegram.

- If the MaC writes *RF_Bidir_TimeOut* to value different from 0 then the MaS shall activate RF bidirectional mode and start the RF Bidirectional Mode Time-out timer with the value as given in the command.

  NOTE 4 This command can actually only be received if bidirectional mode is already active in the MaS, by any of the triggers as specified in 2.2.1.4.

- If the MaC writes the value 0 for the field *RF_Bidir_TimeOut* then the MaS shall immediately stop the RF Bidirectional Mode Time-out timer and deactivate RF bidirectional communication immediately after sending out the below response.

Response (A_FunctionPropertyState_Response-PDU)

Octet 10	Octet 11	Octet 12
Return Code	Max RF_Bidir_TimeOut	Time Left

Return Code	00h:	SUCCESS
	FFh:	ERROR
Max RF_Bidir_TimeOut	The MaS shall fill this field with the maximum time that it can be in in RF bidirectional mode, from 0 s to 254 s.	
	For permanent bidirectional devices, this value shall be 255 s (FFh) and shall denote an infinite bidirectional mode time.	
Time Left	For semi-directional devices, this shall be the current value of the Bidirectional Mode Time-out timer, this is, it shall be the time, in seconds, until the device will deactivate the RF bidirectional mode. This shall be a value in the range from 0 s to 254 s.	
	For permanent bidirectional devices, this value shall be 255 s (FFh) and shall denote an infinite bidirectional mode time.	

2) Read (A_FunctionPropertyState-Read-PDU)

No Data

The MaC shall use the function read PID_RF_BIDIR_TIMEOUT to read the value and state of the RF bidirectional mode and the time-out timer. The MaS shall indicate in the response the maximum time-out value and the current time left value.

The format, contents and interpretation of the response (A_FunctionPropertyState_-Response-PDU) shall be identical as to the above response to the A_FunctionProperty_-Command.

2.3.2.9 PID_RF_DIAG_SA_FILTER_TABLE[] (PID: 61)

- **Property name:** RF Diagnose Source Address Filter Table
- **Property Datatype:** PDT_GENERIC_03[]
- **Datapoint Type:** none

2.3.2.9.1 Abstract Resource definition

The Property *RF Source Address Filter Table* shall contain the Individual Addresses of the RF devices for which the MaS (device) shall measure the link budget.

This table shall be readable and writeable.

Please refer to the Management Procedure NMP_LinkBudget_Measure in 2.4.3 for how the Properties PID_RF_DIAG_SA_FILTER_TABLE, PID_RF_DIAG_BUDGET_TABLE and PID_RF_DIAG_PROBE work together.

2.3.2.9.2 Format

This shall be an array Property; the minimal size shall be 1.

Array Index	Filter Info	Value
1	1st Filter Info	1st Individual Address (2 octets)
2	2nd Filter Info	2nd Individual Address (2 octets)
n		

Figure 2 - Structure of RF Source Address Filter Table

NOTE 5 This table is not sorted.

- Filter Info

Bit	7	6	5	4	3	2	1	0
Description	Reserved =0	Reserved =0	Reserved =0	Reserved =0	0: don't check repeat count 1: check repeat count	Repeat count value [0-7]		

2.3.2.9.3 Usage by the MaS (device)

If the MaS (device) receives any RF Telegram with a SA and repeat count value equal to one of the elements this table, then the MaS shall use the received signal strength of this Telegram to update the corresponding array element in the *RF Diagnose Budget Table* (see 2.3.2.10).

A Source Address can be present in more than one element.

The value 0000h shall be an invalid value for the field *Individual Address*. The MaC may use this value to denote an unused entry in the array Property. The MaS shall not evaluate entries with this value for the Individual Address, regardless of the repeat count value.

NOTE 6 This allows that the diagnosis can differentiate between communication directly from the device sending with that Source Address and communication from that same sender that is repeated by one or more RF Retransmitters.

If the MaC writes any element of this table, the MaS shall reset the associated element in the *RF Diagnose Budget Table* (see 2.3.2.10), even if the element is overwritten with the same data.

2.3.2.10 PID_RF_DIAG_BUDGET_TABLE[] (PID: 62)

- **Property name:** RF Diagnose Budget Table
- **Property Datatype:** PDT_GENERIC_04[]
- **Datapoint Type:** none

2.3.2.10.1 Abstract Resource definition

The *RF Diagnose Budget Table* shall for each Source Address contained in the or the RF Source Address Filter Table contain the calculated link budget information based on the received and filtered Telegrams from that Source Address.

Please refer to the Management Procedure NMP_LinkBudget_Measure in 2.4.3 for how the Properties PID_RF_DIAG_SA_FILTER_TABLE, PID_RF_DIAG_BUDGET_TABLE and PID_RF_DIAG_PROBE work together.

2.3.2.10.2 Format

The *RF Diagnose Budget Table* shall be an array Property as shown in Figure 3.

Array Index	Nr of Telegrams	Link Budget		
		Min	Max	Average
	(1 octet)	(1 octet)	(1 octet)	(1 octet)
		dBm	dBm	dBm
1		0 to 255	0 to 255	0 to 255
2				
…				
n				

Figure 3 - Structure of RF Source Address Filter Table

This shall be a read-only Property. The default value shall be 0 for all fields. The MaC can only clear the contents of any entry in this Property by writing the corresponding element in the Property RF Diagnose Source Address Filter Table (see 2.3.2.9).

2.3.2.10.3 Usage by the MaS (device)

Every element of the *RF Diagnose Budget Table* with array index m shall relate 1-to-1 to the element of the *RF Diagnose Source Address Filter Table* with the same array index m.

- Nr of Telegrams
  The MaS (device) shall count in this field all received Telegrams with Source Address and repeat count value as in the corresponding entry in the *RF Diagnose Source Address Filter Table*. If more than 255 matching Telegrams are received then the *Nr of Telegrams* shall stay at 255, but the Link Budget (min, max and average) shall still be updated.
- Link Budget (in dBm)
  The Link Budget shall be the difference between the RSSI of the received Telegram and the sensitivity of the MaS. This value will normally be greater than 0. A big value will indicate a good communication between to products.

- Min
  This shall be the minimal value of the Link Budget calculated for all received Telegrams for this entry.
- Max
  This shall be the maximal value of the Link Budget calculated for all received Telegrams for this entry.
- Average
  This shall be the floating average value of the Link Budget calculated for all received Telegrams for this entry. This shall be updated with each received matching Telegram nr n as follows.

$$average_n = \frac{average_{n-1} + new\ value}{2}$$

Reset of an element

The MaS shall manage the *RF Diagnose Budget Table* and shall make it contain exactly the same number of elements as the *RF Diagnose Source Address Filter Table*. The MaS shall clear (value 0) all fields in an entry in the *RF Diagnose Budget Table* if the MaC writes the corresponding element in the *RF Diagnose Source Address Filter Table*.

- If the MaC writes a new Property array element in the *RF Diagnose Source Address Filter Table* beyond the current nr of elements, then the MaS (device) shall also grow the *RF Diagnose Budget Table* to have the same length and fill the newly created entry or entries with 0.
- The clearing shall happen on any write event to *RF Diagnose Source Address Filter Table*, even with identical data.

2.3.2.10.4 Usage by the MaC

The MaC shall read out any entry in this array Property to learn the Link Budget of the RF communication between this MaS and the KNX RF device with the given IA.

In order to clear any data in this Property, the MaC shall access the corresponding entries in the Property *RF Diagnose Source Address Filter Table*.

2.3.2.11 PID_RF_DIAG_PROBE (PID: 63)

- **Property name:** RF Diagnose Probe
- **Property Datatype:** PDT_FUNCTION
- **Datapoint Type:** None

2.3.2.11.1 Abstract Resource definition

The Property *RF Diagnose Probe* (PID_RF_DIAG_PROBE) shall support the measuring of the Link Budget between the MaS and another KNX RF device. It shall allow the MaC to trigger the MaS to transmit this Property State to the network, of which the other KNXRF device can measure the signal strength and calculate the Link Budget.

Please refer to the Management Procedure NMP_LinkBudget_Measure in 2.4.3 for how the Properties PID_RF_DIAG_SA_FILTER_TABLE, PID_RF_DIAG_BUDGET_TABLE and PID_RF_DIAG_PROBE work together.

2.3.2.11.2 Format

PID_RF_DIAG_PROBE shall be a Function Property that shall be formatted as follows.

2.3.2.11.3 Usage by the MaS (device)

1) Write (A_FunctionPropertyCommand-PDU)

Octet 10
Trigger: 00h

The field Trigger shall have the fixed value 00h. Other values are reserved for later extensions and shall not be used. If the MaS (device) receives a value for this field *Trigger* that it does not support, then it shall give a negative Function response.

Response (A_FunctionPropertyState_Response-PDU)

Octet 10	Octet 11
test_info	test_result
00h	00h

The MaS shall respond to the MaC with this A_FunctionPropertyState_Response-PDU in point-to-point connectionless communication mode.

NOTE 7 The trigger is specified as a constant 00h. Future alternative values may control the information to be sent, like the RF Channel to use, to what receivers to send it and the communication mode (e.g. point-to-point connectionless).

Additionally, the MaS (device) shall send the state of this function using the service A_NetworkParameter_InfoReport in broadcast communication mode. The field test_result shall be 00h.

Octet 6								Octet 7								Octet 8								Octet 9								Octet 10								Octet 11								Octet 12							
						APCI										parameter_type																				test_info								test_result							
																object_type																property_id																			
																RF Medium Object																63								00								00h							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
						APCI	APCI	APCI	APCI	APCI	APCI	APCI	APCI																																								
						1	1	1	1	0	1	1	0	1	1																																					

Figure 4 – A_NetworkParameter_InfoReport - RF Diagnose Probe message (example)

2) Read (A_FunctionPropertyState-Read-PDU)

If the MaC reads that status of the Function Property PID_RF_DIAG_PROBE, then the MaS shall respond with the same value as on the writ confirmation.

Octet 10	Octet 11
test_info	test_result
00h	00h

NOTE 8 The octet 10, test_info, shall in this be seen as an identifier of the type of information that is transmitted. Future values may indicate additional octets in the field test_info or different information in the field test_resulT;

The MaS shall not transmit any second message like A_NetworkParameter_InfoReport on the network in this case.

2.3.3 cEMI Server Object

2.3.3.1 Overview

This clause is only an overview and is not intended for integration in the KNX Specifications.

For an overview of the mandatory an optional Interface Objects and Properties in the KNX RF Interface Type 1, please refer to clause 2.7.2.

Property Identifier (PID)	Property Data Type	Value
1 PID_OBJECT_TYPE	PDT_UNSIGNED_INT	cEMI Server Object: Object Type = 8
54 PID_ADD_INFO_TYPES	PDT_ENUM8[]	Additional Information Types

2.3.3.2 PID_ADD_INFO_TYPES (PID: 54)

- **Property name:** Additional Information Types
- **Property Datatype:** PDT_ENUM8[]
- **Datapoint Type:** DPT_AddInfoTypes (DPT_ID = 20.1001)

For the common definition of PID_ADD_INFO_TYPES, please refer to [03].

For the KNX RF Interface Type 1, the Addinfo-Type 02h “RF Control Octet and Serial Number or DoA” shall be supported.

2.3.4 Programming Mode

The following shall be added to the specification of Programming Mode in clause 4.19 in [03], preceding the existing clause 4.19.2 “Programming Mode – Realisation Type 1”.

2.3.4.1 Abstract Resource definition

The “Abstract Resource definition” shall be moved from the clause “Format and encoding” of clause 4.19.3.1 to this clause.
Additionally, the following shall be added.

Autonomous inactivation of Programming Mode

This is an optional feature for the MaS. Please refer to [05] for the conditions for the support of this feature.

If this feature is supported, then, if the Programming Mode becomes enabled, by any means, the MaS (device) shall start a time-out timer of 4 minutes.

This timer shall count down and it shall not be possible to restart this timer.

If the timer expires, the MaS shall autonomously and automatically disable its Programming Mode.

2.4 Management Procedures

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

2.4.1 Introduction

The Management Procedures needed for Configuration are identical as to the existing Management Procedures as used for mask 0705h and 07B0h.

Additionally, new Management Procedures are specified, mainly for Network Discovery, Configuration and Diagnosis. These are not specified in this paper but are contained in [24].

2.4.2 NM_SubnetworkDevices_Scan2

Use

This Network Management Procedure shall be used to determine which devices exist on a Subnetwork.

The MaC shall to this one after the other address each possible Individual Address in the Subnetwork by reading the Device Descriptor Type 0 with the service A_DeviceDescriptor_-Read, using point-to-point connectionless communication mode. If the MaC does not receive a response within one second, it shall repeat the request once and wait again for one second for a possible response.

- If the MaC receives a response then it shall assume that the tested IA is occupied.
- If the MaC receives no response, then it shall assume that the tested IA is not occupied.

For this procedure the Individual Address of the used Routers and the Domain Address have to be configured.

Used Application Layer Services for Management

- A_DeviceDescriptor_Read

Parameters of the Management Procedure

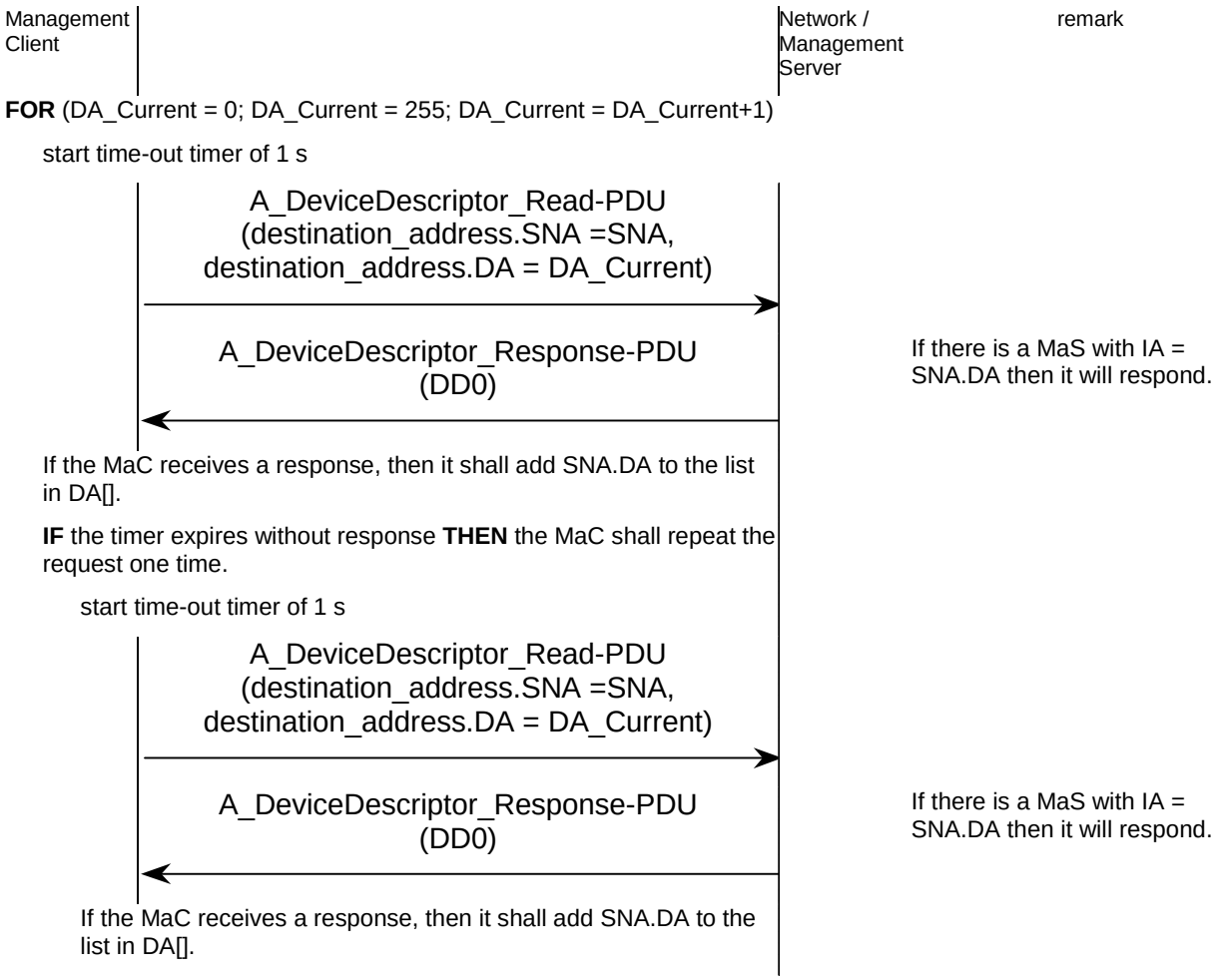
NM_SubnetworkDevices_Scan2(/* [in] */ SNA, /* [out] */ DA[])

SNA:	Subnetwork Address of the Subnetwork in which the occupied Individual Addresses are to be scanned.
DA[]:	The collection of all Device Addresses of the devices discovered in the investigated Subnetwork.

Variables

DA_Current:	The current Device Address of which it will be checked whether a device with this Device Address exists on the Subnetwork that is being checked.

Sequence



Constraints

1. This procedure uses the connectionless communication mode for the service A_Device-Descriptor_Read. This service will therefore give incomplete results if there are KNX devices in the Subnetwork that do not support this service in connectionless communication mode.
2. Additionally, this procedure does not use the T_Connect-service. Devices that do not react to A_DeviceDescriptor_Read but only negatively react to a T_Connect-PDU by sending a T_Disconnect-PDU, will not be discovered with this procedure.

   EXAMPLE 3 Certain KNX Profiles support multiple Individual Addresses in one device, like the KNXnet/IP Tunnelling Server. These Additional IA may not be discovered.

3. The MaC shall only apply this Management Procedure if the target Subnetwork is of the KNX RF Communication Medium.

2.4.3 NMP_LinkBudget_Measure

Use

This Network Management Procedure shall be used to diagnose the Link Budget of the communication between one MaS and one or more other devices (probes).

To this, the MaC will firstly set the IAs and possibly the hop count and test criteria in the MaS. Then, it will trigger each subsequent probe to send a message. Finally, the MaC will read out the results from the MaS.

Extended use: larger number of probe messages

If step 2 in the below procedure is executed only once then this test will only give a simple basic result.

The MaC may also repeat step 2 a number of times, so that there is richer information in the resulting minimal - , maximal - and average values.

Alternative use: long term recording

The MaC may also skip part 2 and not actively trigger the probes.

In this case, it will wait a longer period (hours, days) before reading out the result from the MaS.

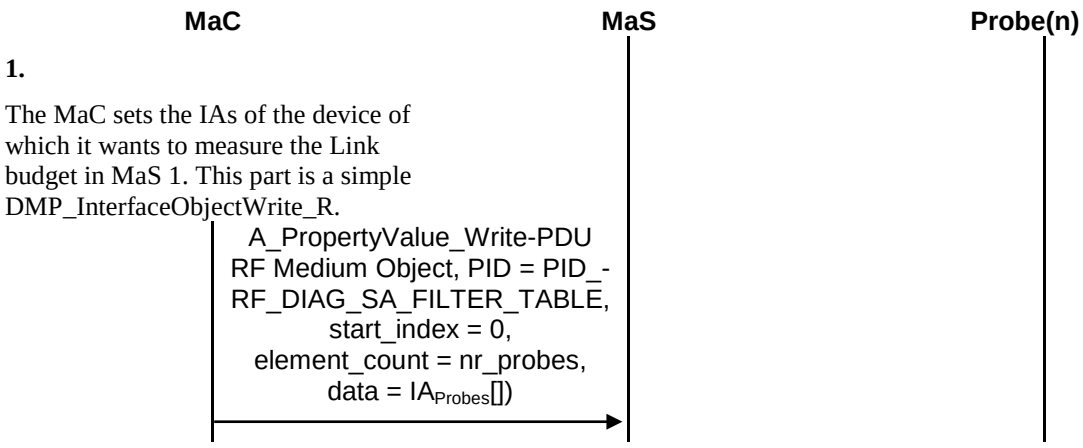
Used Application Layer Services for Management

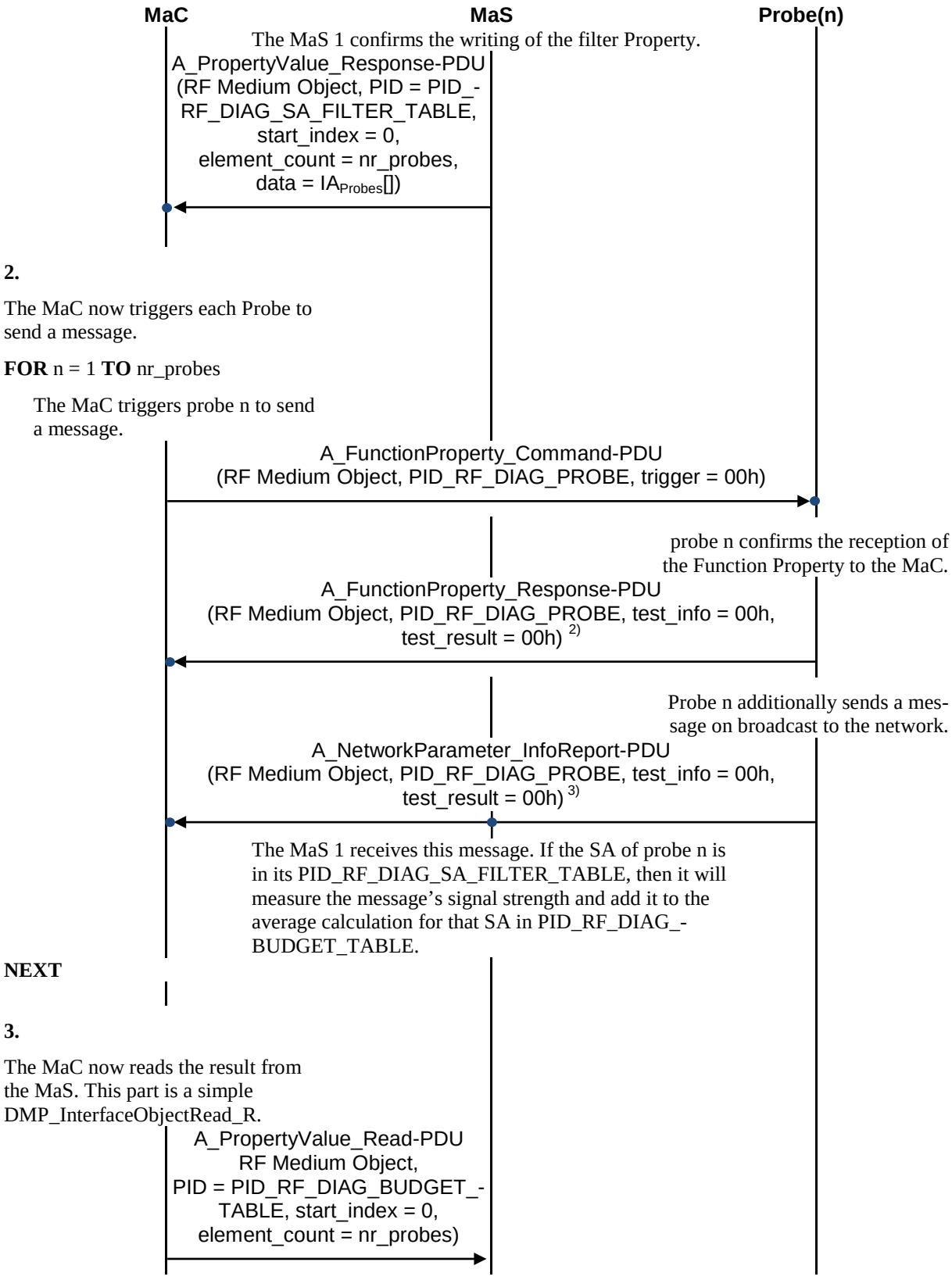
- A_PropertyValue_Write
- A_PropertyValue_Read
- A_FunctionProperty_Command
- A_NetworkParameter_InfoReport

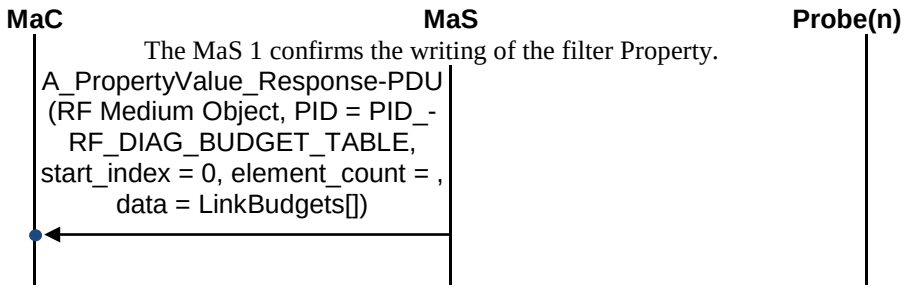
Parameters of the Management Procedure

IA_{Probes}[]:	The list of Individual Addresses of devices of which the MaC wants to read the Link Budget of their communication with MaS 1.
nr_probes:	This is the number of probes in IA_{Probes}[], thus the size of IA_{Probes}[].
LinkBudgets[]	The array of Link Budgets (nr of Telegrams, minimal, average and maximal values) according Figure 3. The size is nr_probes. this is the result of this Management Procedure.

Procedure







NOTES TO THE PROCEDURE

1. This procedure is designed for the MaS and the probe n to be on the same Subnetwork (with the same Subnetwork Address). If the MaS1 and the probe would be on different Subnetworks, with different DoAs, then the result will not be the link budget between the MaS and the probe, but between the MaS and the Media Coupler between the MaS and the rest of the KNX network.
2. This A_FunctionProperty_Response-PDU is in all cases sent to the MaC: this I the default behaviour of Function Properties and it adds reliability of this procedure to the MaC, as its command is confirmed.
3. The A_NetworkParameter_InfoReport PDU is sent on broadcast communication.
   - Other devices than the involved MaS within the RF Domain will receive this RF Telegram as well and may also update their RF Diagnose Budget Table. This way, the MaC may run more than one test, to different MaS, in parallel.
   - Additionally, this broadcast Telegram gives an indication to the MaC that the Probe sends the test Telegram.

Further notes

- It is assumed that the RF communication between two RF devices 1 and 2 is symmetrical, this is, that the link budget with device 1 as MaS and device 2 as probe, will be the same as the link budget with device 1 as probe and device 2 as MaS.
- If the, while this procedure is running, the IA of any probe device in this procedure changes or is assigned (by the MaC) to a different KNX RF device, then this procedure will return an incorrect result for that IA entry. There are not requirements towards the MaS to detect this. The MaC should only run this procedure in an environment where the IAs are stable during the execution of this procedure.

2.5 Configuration Procedures

2.5.1 Network Configuration Procedures

2.5.1.1 Deep Scan

This procedures searches a Subnetwork on KNX with a given Subnetwork Address for all the Individual Addresses that are used in it.

```
        /* Search for all the Individual Addresses in the Subnetwork X. */
NM_SubnetworkDevices_Scan2(SNA =X, DA[]);
```

This returns a list of all Device Addresses DA[] that are occupied in the Subnetwork.

This Network Configuration Procedure shall only be applied on a KNX RF Subnetwork.

2.5.1.2 Other Network Configuration Procedures

These will be specified in a next version of this document.

2.5.2 Device Configuration Procedures

These will be specified in a next version of this document.

2.6 External Message Interface

In clause 4.1.4.3.2 in [13], it was so far only foreseen that the cEMI Server fills in its own KNX Serial Number, this was written explicitly. It was not foreseen that the cEMI Server would fill in alternatively its own DoA. So, this had to be added.

<table>
<tr><th>Field</th><th>Bit</th><th>Description</th><th>Coding</th></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
<tr><td>RF-‘SN’</td><td></td><td>KNX Serial Number or DoA: shall be interpreted as KNX Serial Number or DoA according to ‘SB’ flag in cEMI Control Field 1.
- L_Data.req:
<table>
<tr><th rowspan="2">SB – flag</th><th colspan="2">SN field = 000000000000h</th></tr>
<tr><th>yes (= void)</th><th>no</th></tr>
<tr><td>0</td><td>The cEMI Server shall insert its own KNX Serial Number in the RF message to be sent.</td><td>The cEMI Server shall insert the value of the ‘SN’ field in the RF message to be sent.</td></tr>
<tr><td>1</td><td>The cEMI Server shall insert its own DoA in the RF message to be sent.</td><td>The cEMI Server shall insert the value of the ‘SN’ field as DoA the RF message to be sent.</td></tr>
</table>
- L_Data.con: shall contain the effective value of SN or DoA in the sent RF message.
- L_Data.ind: shall contain the received value of the SN or the DoA from RF message</td><td>000000000000h: void (no valid KNX Serial number or DoA)</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td></tr>
</table>

2.7 Profile definition

2.7.1 Profile: mask 2705h

It is the intention in the future that each Profile gets its own document in Volume 6 “Profiles”, so that the own identity, background and motivation of the Profile does not go lost, as has been the case with the System B specifications as these got integrated in the KNX Specifications.
This clause, and the following clauses for the System B implementation (2.7.2) and the KNX RF USB interface (2.7.2.1) are considered to become such stand-alone papers.

2.7.1.1 Introduction and common requirements

Mask 2705h shall be a realisation of the System 7 model in the KNX RF Communication Medium.

2.7.1.2 Profile: Operation

2.7.1.2.1 Aims

This Profile describes the features of a device necessary for operation. The aim is to guarantee runtime Interworking between all devices in the system. The main components for this objective are the support of group oriented (multicast) communication.

2.7.1.2.2 Common Profile

These Profiles refer to all devices in all Configuration Modes on all media. Medium dependent features are described in clause 2.7.1.3.

Feature	mask 2705h
1 Medium dependent Layers	KNX RF medium
2 Physical Layer General	M
3 Link Layer general	M
4 Link Layer - Router	n/a
5 Link Layer - Bridge	n/a
6 Network Layer general	M
7 Network Layer - Router	n/a
8 Network Layer – Bridge	n/a
9 Network Layer RF Retransmitter	O
10 Network Layer RF Media Coupler	X
11 Transport Layer – multicast	M
12 Transport Layer – connection oriented minimal	X
13 AL – Group Object services	M
14 AL – Property Value Services	M
15 AL – Function Property Services	M
16 AIL – GO	M
17 AIL – IO	M
18 AIL – Function Properties	M
19 AIL – GO indirection	O
20 Application Interface Layer for unidirectional devices	X

2.7.1.2.2.1 Medium dependent Layers

The Profiles defined in clause 2.7.1.3 “Medium dependent layers” of this document apply.

2.7.1.2.2.2 Physical Layer - general

	Specification	Test
	• **General**	
[06]	(contains no requirement)	none.

2.7.1.2.2.3 Data Link Layer - general

Specification	Test
• **General**	
[07] - §1.1 “Functions of the Data Link Layer” - §1.2 “Possible Media and their Impact on Layer-2” - §1.3 “Objective”	tested with medium specific tests
• **Individual address /Group Address**	
[07] - §1.4 “Definitions”	tested with medium specific tests
• **Data Link Layer Protocol**	
[07] - §3 “Data Link Layer Protocols”	tested with medium specific tests
• **Parameters**	
[07] - §4 “Parameters of Layer-2” (except TP1 Fast Polling)	tested with medium specific tests

2.7.1.2.2.4 Network Layer - general

Specification	Test
• **General**	
[08] - §1 “Overview”	[15]
• **NPDU**	
[08] - §2.1 “NPDU”	[15] - §3 (Black Box Tests) - All end devices [14] - RF bidirectional end device
• **Parameters**	
[08] - §2.3 “Parameters of Network Layer” - hop_count: preferred value: 6.	- §3(Black Box Tests) - All end devices [14] - RF bidirectional end device
• **state machine**	
[08] - §2.4.1	[15] - §3 (Black Box Tests) - All end devices - [14] - RF bidirectional end device

2.7.1.2.2.5 Network Layer- RF Retransmitter

Specification	Test
[02] - §5.5.1 “History List” - §5.5.2 “RF Repeat Counter” - §5.5.3 “Filtering”	[14] - §3

2.7.1.2.2.6 Transport Layer–multicast

Specification	Test
• **TPDU**	
[09] - §1.2 “Point-to-Multipoint, Connectionless (Multicast) Communication Mode” - §3.1 “T_Data_Group Service”	[16] - All end devices **[14]** - RF bidirectional end device

2.7.1.2.2.7 Application Layer – Group Oriented

Specification	Test
• **APDU**	
[10] - §2 “APDU” o A_GroupValue_Read-PDU o A_GroupValue_Response-PDU o A_GroupValue_Write-PDU	[18] - All end devices [14] - RF bidirectional end device
• **Data length**	
[10] data must be coded as indicated in §3.1 “Application Layer Services on Multicast Communication Mode”	[18] - All end devices [14] - RF bidirectional end device

2.7.1.2.2.8 Application Layer – Property Value Services

Specification	Test
• **APDU**	
[10] - §2 “APDU” - §3.4.3.1 “A_PropertyValue_Read-service” - §3.4.3.2 “A_PropertyValue_Write-service”	

2.7.1.2.2.9 Application Layer – AL services on system broadcast communication mode

Unless indicated explicitly, the services shall be supported in full, this means, all service primitives shall be supported (.req, .Lcon, .ind, .res, .Rcon and .Acon).

Specification	Test
• **APDU**	
[10] - A_DomainAddress_Write - A_DomainAddress_Read - A_IndividualAddressSerialNumber_-Write - A_DomainAddressSerialNumber_Read - A_DomainAddressSerialNumber_Write - A_DomainAddressSelective_Read [24] - A_SystemNetworkParameter_Read	

2.7.1.2.2.10 Application Layer – AL services **on broadcast** communication mode

Unless indicated explicitly, the services shall be supported in full, this means, all service primitives shall be supported (.req, .Lcon, .ind, .res, .Rcon and .Acon).

Specification	Test
• **APDU**	
[10] - A_IndividualAddress_Write - A_IndividualAddress_Read - A_IndividualAddressSerialNumber_-Read - A_IndividualAddressSerialNumber_-Write - A_NetworkParameter_Read - A_NetworkParameter_Write	

2.7.1.2.2.11 Application Layer – Function Services

Specification	Test
• **APDU**	
[10] - §2 “APDU” - §3.4.5 “Function Property Services”	

2.7.1.2.2.12 Application Interface Layer – Group Objects

Specification	Test
• **Group Objects**	
[11] - §3 “Group Object Server”	[17] - §1.3 “Group Object Tests” - all end devices **[14]** - - RF bidirectional end device

2.7.1.2.2.13 Application Interface Layer – Interface Objects

Specification	Test
• **Interface Objects**	
[11] - §4 “Interface Object Server”	

2.7.1.2.2.14 Application Interface Layer – Function Properties

Specification	Test
• **Function Properties**	
[11] - §4.4.2 “Function Properties”	

2.7.1.2.2.15 Application Layer Interface Layer – Group Objects indirection

Specification	Test
• **Group Object Indirection**	
[11] - §3.4 “Group Object Indirection – Group Object Handles and PID-OBJECT_-VALUE (PID = 62) [03] - §4.3.13 “PID_OBJECT_VALUE (PID = 62)”	

2.7.1.2.3 Specific parts

These Profiles describe runtime mechanisms that are not relevant for all devices.

Feature	mask 2705h
1 Link Layer – polling	n/a
2 Detection of Usage of own Individual Address	X
3 Extended Group Object Flags	M

2.7.1.2.3.1 Extended Group Object Flags

Specification	Test
• **configuration flags: update on read response (1 bit)**	
[11] - §3.3.2 “Reading the Group Object Value ” (Update Enable)	[17] - §1.3 (Group Object Tests)

2.7.1.3 Medium dependent layers

2.7.1.3.1 Goal

This Profile describes the requirements on a device in order to guarantee compliance with one of the standard communication media of the system. Compliance with one of these Profiles is a prerequisite for both runtime and configuration Interworking.

2.7.1.3.2 RF medium dependent layers

	mask 2705h
1. Physical Layer Sender	X
2. Physical Layer Transceiver	M
3. Link Layer	M
4. KNX Serial Number for multicast communication	O
5. RF Domain Address for multicast communication	M
6. LTE frame format	O
7. Extended LTE Group Addresses and frame acceptance	O
8. Link Layer-Retransmitter	O
9. Link Layer-Media Coupler	O
10. Local Services Metering	O
11. KNX Serial Number	M
12. APDU-length	≥ 15

2.7.1.3.2.1 Physical Layer Sender

Specification	Test
[02] - §5.1 “Physical Layer for KNX RF Ready and BiBat” - §5.2 “Datagram structure for RF Ready and BiBat”	

2.7.1.3.2.2 Physical Layer Transceiver

Specification	Test
[02] - §5.1 “Physical Layer for KNX RF Ready and BiBat” - §5.2 “Datagram structure for RF Ready and BiBat”	

The S-Mode Configuration Procedures may result in a lot of Telegrams on the KNX RF Medium. For this, it is strongly recommended, both for the MaS as for the S-Mode MaC, to apply Listen Before Talk (LBT, see [02]), to avoid RF collision. LBT is part of the KNX RF Multi type, but with S-Mode Configuration Procedures, the Telegram rate is higher than the duty cycle requirements for the KNX RF Ready type. Therefore, in order not to limit the Telegram rate, LBT is a recommended solution.

2.7.1.3.2.3 Link Layer

	Specification	Test
	• **Frame format**	
[02]	- §5.1.2 “Datagram structure for RF Ready and BiBat	
	• **Medium Access Control**	
[02]	- §5.1.3 “Medium access”	
	• **Addressing**	
[02]	- §6.1.1.1 “Extended Group Address” – point-to-multipoint – DoA of sender - §6.1.4.2 “Duplication prevention”	

The reference to clause 6.1.1.1 is actually a forward reference to where this will be specified after integration of this document. For the time being, what is referred to here as clause 6.1.1.1 is actually specified in clause 2.2.1.1 of this document.

2.7.1.3.2.4 KNX Serial Number for multicast communication

	Specification	Test
[02]	- §6.1.1.1 “Extended Group Address”	

Please note that the contents of the referred clause will be modified by this document.

2.7.1.3.2.5 RF Domain Address for multicast communication

	Specification	Test
[02]	- §6.1.1.4 “RF Domain Address”	

Please note that the contents of the referred clause will be modified by this document.

2.7.1.3.2.6 LTE frame format

	Specification	Test
[02]	- §6.1.2.4 “First block” - §6.1.2.6 “Second block for LTE Extended Datagrams”	

2.7.1.3.2.7 Extended LTE Group Addresses and frame acceptance

	Specification	Test
[02]	- §6.1.5.3.1 “Additional Frame acceptance criteria for LTE Extended Datagrams”	

2.7.1.3.2.8 Link Layer-Retransmitter

	Specification	Test
[02]	- §6.2.2 “The Layer-2 of an RF Retransmitter”	

2.7.1.3.2.9 Link Layer-Media Coupler

	Specification	Test
[02]	- §6.1.6 “The Layer-2 of an RF-TP Media Coupler”	

2.7.1.3.2.10 Local Services Metering

	Specification	Test
[13]	- §2 “Message format” – code for L_Meter.ind - §3.3.3.5 “L_Meter.ind”	

2.7.1.3.2.11 KNX Serial Number

	Specification	Test
[03]	- §4.15.1 “Abstract Resource definition” - §4.15.2 “KNX Serial Number – Realisation Type 1”	

2.7.1.4 Configuration & Management (S-Mode, Server)

This clause describes the requirements on an S-Mode device, which are relevant for configuration as a Management Server accessed only via the bus. The objective is to guarantee Interworking with the configuration tool (ETS).

2.7.1.4.1 Communication

	Feature	Mask 2705h
1	TL - broadcast	M
2	TL - connection oriented	M
3	TL - connection oriented minimal	X
4	TL - connectionless	M

2.7.1.4.1.1 TL - broadcast

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives. - §1.3 “Point-to-all-Points Connectionless (Broadcast) Communication Mode” - §2 “TPDU” - §3.4 “T_Data_Broadcast” - §4 “Parameters of Transport Layer”	

2.7.1.4.1.2 TL - connection oriented

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives. - §1.6 “Point-to-Point, Connection-Oriented Communication Mode” - §2 “TPDU” - §3.7 “T_Connect service” - §3.8 “T_Disconnect service” - §3.9 “T_Data_Connected service” - §4 “Parameters of Transport Layer” - §5.1 “States” - §5.2 “Actions” - §5.3.3 “Style 3”	[16]

2.7.1.4.1.3 TL - connectionless

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives: - §1.5 “Point-to-point connectionless Communication Mode” - §2 “TPDU” - §3.6 “T_Data_Individual”	[16]

2.7.1.4.2 Device Management

In this clause all general requirements on a device concerning the mechanisms used for access by the Management Client are described.

Feature	mask 2705h
1 Direct memory Access	M
2 DMA on User Memory	M
3 Verify mode [1)]	M
4 Interface Object Handling [2)]	M
5 Reduced Interface Objects	X
6 Function Properties	M
7 Load and Run State Machines	
a. Realisation Type 1	M
b. Realisation Type 2	O
8 Hardware Specific Parameters	n/a
9 RAM cleared	M
10 User EEPROM	M
11 Restart	
a. connectionless	O
b. connection-oriented	M
c. Master Reset	O
12 Authorization [3)]	M
nr of access levels	16
13 Memory Control Table	O

1) If Verify Mode is not implemented, it shall always be off.

2) Please refer to the specification of mandatory and optional Interface Objects, Properties and Property fields further in this Profile definition.

3) The support of the A_Authorize- and the A_Keywrite-service does not imply that the device itself has access protected areas. If this is not the case, a device shall always allow – regardless of the attributed keys – access to the highest level (0), including when receiving an illegal key (‘illegal’ in this sense meaning another key than any of the keys entered in the key table).

2.7.1.4.2.1 Direct Memory Access

Specification	Test
[12] - §3.2.1 “DMP_Connect_RCo” - §3.3.2 “DMP_Disconnect_RCo” - §3.18.2 “DMP_MemRead_RCo” - §3.16.2 “DMP_MemWrite_RCo”	[17] - §2 “Network Management Server Tests corresponding tests

2.7.1.4.2.2 DMA on user memory

Specification	Test
[12] - §3.19 “DM_UserMemWrite”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.1.4.2.3 Verify Mode (for A_Memory_Write)

Specification	Test
[03] - §4.2.14.7 "Verify Mode Control" management server part.	[17] - §2 ”Network Management Server Tests” Tests 7.1 to 7.7

2.7.1.4.2.4 Interface Object Handling

Please also refer to the specification of mandatory and optional Interface Objects, Properties and Property fields further in this Profile definition.

Specification	Test
[12] - §3.22.2 “DMP_InterfaceObjectWrite_R” - §3.24.2 “DM_InterfaceObjectRead_R” - §3.23.2 “DMP_InterfaceObjectVerify_R” - §3.25.2 “DM_InterfaceObjectScan_R”	[17] - §2 “Network Management Server Tests” corresponding tests
[11] - §4 “Interface Object Server”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.1.4.2.5 Reduced Interface Objects

Specification	Test
[12] - §3.22.2 “DMP_ReducedInterface-ObjectWrite_R” - §3.24.3 “DMP_ReducedInterfaceObject-Read_R” - §3.25.3 “DMP_ReducedInterfaceObject-Scan_R”	[17] - §2 “Network Management Server Tests” corresponding tests
[11] - §4.1 “Common structure” - §4.3.2 “Reduced Interface Object”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.1.4.2.6 Function Properties

Specification	Test
[10] - §2 “APDU” - §3.4.1 “Introduction” - §3.4.5 “Function Property Services”	
[11] - §4.1 “Common structure” - §4.2 “Minimal requirements of Interface Objects” - §4.4.2 “Function Properties”	
[12] - §3.26 “DM_FunctionProperty_Write_R”	

2.7.1.4.2.7 Load - and Run State Machines

a) Realisation Type 1 - Property based	
Specification	**Test**
[03] - records	
[12] - §3.27.3 “DMP_LoadStateMachine-Write_Rco_IO” - §3.28.3 “DM_LoadStateMachine-Verify_R_IO” - §3.29.3 “DMP_LoadStateMachineRead_R_IO” - §3.30.3 “DMP_RunStateMachineWrite_R_IO” - §3.31.3 “DMP_RunStateMachineVerify_R_IO” - §3.32.3 “DMP_RunStateMachineRead_R_IO”	[17] - §2 “Network Management Tests” corresponding tests
b) Realisation Type 2 – Memory mapped	
Specification	**Test**
[03] records	
[12] - §3.27.2 “DMP_LoadStateMachineWrite_Rco_Mem” - §3.28.2 “DMP_LoadStateMachineVerify_Rco_Mem” - §3.29.2 “DMP_LoadStateMachineRead_Rco_Mem” - §3.30.2 “DMP_RunStateMachineWrite_Rco_Mem” - §3.31.2 “DMP_RunStateMachineVerify_Rco_Mem” - §3.32.2 “DMP_RunStateMachineRead_Rco_Mem”	[17] - §2 “Network Management Tests” corresponding tests

2.7.1.4.2.8 Hardware specific Parameters

	Specification	Test
	• **0100h (RW)**	
[03]	- §4.18 “OptionReg”	

2.7.1.4.2.9 RAM cleared

This is the RAM to be cleared by the Management Client during download of an application program.

	Specification	Test
	• **RAM**	
[03]	- Resources from 0700h	

2.7.1.4.2.10 User EEPROM

This part of the EEPROM can be used by an application program.

e) System 7		
	Specification	**Test**
[03]	- from 4000h to CFFFh	

2.7.1.4.2.11 Restart

2.7.1.4.2.11.1 Restart connectionless

	Specification	Test
[12]	- §3.7.2 DM_Restart_RCl	

2.7.1.4.2.11.2 Restart connection-oriented

	Specification	Test
[12]	- §3.2.1 DMP_Connect_RCo - §3.7.3 DM_Restart_RCo	- §2.9.1 “Send Restart”

2.7.1.4.2.11.3 Master Reset

Master Reset is optional for the mask 2705h. However, as KNX RF is an open medium, it would be possible for any hacker to set a KNX RF S-Mode device out of operation by an unprotected Master Reset. Therefore, it is recommended that if Master Reset is implemented, that it only be accepted from an authenticated communication partner, using the A_Authorize-service or KNX Data Security (see [21]).

	Specification	Test
[03]	- §4.9.1 “Group Address Table” – “default state” - §4.10.1 “Group Object Association Table” – “default state”	To be completed.
[12]	- §3.7.2 “DM_Restart_RCl” – master reset - §3.7.3 “DM_Restart_RCo” – master reset	To be completed.

2.7.1.4.2.12 Authorization

	Specification	Test
	• **Management Procedures**	
[12]	- §3.5.1 “DM_Authorize_RCo” - §3.5.2 “DM_Authorize2_RCo” - §3.6 “DM_SetKey”	

2.7.1.4.2.13 Memory Control Table

	Specification	Test
	• **PID_MCB_TABLE**	
[03]	- §4.2.27 “PID_MCB_TABLE” (PID = 27) as read-only.	

2.7.1.4.3 Device Discovery

	Feature	Mask 2705h
1	Management Procedures with A_SystemNetwork-Parameter_Read-PDU	
	1.1 NM_Read_SerialNumber_By_ProgrammingMode	M
	1.2 NM_Read_SerialNumber_By_ExFactoryState	M
	1.3 NM_Read_SerialNumber_By_PowerReset	M

2.7.1.4.4 Device Identification

Feature	Mask 2705h
1 Device Descriptor Service - connection oriented	M
2 Device Descriptor Service - connectionless	M
3 Device Descriptor Type 0	M
4 Device Descriptor Type 2	O
5 Device Descriptor InfoReport	O
6 Management Descriptor 1	O
7 Identification of hardware	M
8 Identification of Application	O

2.7.1.4.4.1 Device Descriptor Service - connection oriented

Specification	Test
[10] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.7.1.4.4.2 Device Descriptor Service - connectionless

Specification	Test
[10] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)
[12] - §3.2.2 “DMP_Connect_RCl”	

2.7.1.4.4.3 Device Descriptor Type 0

Specification	Test
[03] - §4.1.2 “Device Descriptor Type 0”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.7.1.4.4.4 Management Descriptor 1

Specification	Test
• **Management Descriptor 1**	
[03] - §4.3.23 “PID_MGT_DESCRIPTOR (PID = 72)”	
[12] - §3.4.2 “DM_Identify_R”	

2.7.1.4.4.5 Identification of the hardware

	Specification	Test
[12]	- §3.4.3 “DM_Identify_RCo2”	

2.7.1.4.4.6 Identification of Application

	Specification	Test
[10]	- §3.5.6.5 "A_UserManufacturerInfo_-Read_service"	

2.7.1.4.5 Device Individualisation

In this clause all requirements on a device for device individualisation and assignment of the Individual Address are described.

	Feature	Mask 2705h
1	Programming Mode	
	1.a Connection oriented	M
	1.b Connectionless	O
	Autonomous Inactivation	M
2	KNX Serial Number	
	a client initiated	M
	b server initiated	O
3	Domain Address Assignment	M
4	Default Individual Address	O

2.7.1.4.5.1 Programming Mode

2.7.1.4.5.1.1 connection oriented

a) Realisation Type 1 - Property based	
Specification	**Test**
[12] - §2.2 “NM_IndividualAddress_Read” - §2.3 “NM_IndividualAddress_Write” 4)	
Programming Mode Control • via HMI: device selection and indication of Programming Mode • via bus: [03] - §4.3.5 “PID_PROGMODE” [12] - §3.22.2 “DMP_InterfaceObject-Write_R” - §3.23.2 “DMP_InterfaceObject-Verify_R” - §3.24.2 “DMP_InterfaceObject-Read_R”	

2.7.1.4.5.1.2 Programming Mode – connectionless

Specification	Test
[12] - §2.10 “NM_DomainAnd-IndividualAddress_Write2” - §2.2 “NM_IndividualAddress_Read”	[17] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”
Programming Mode Control • via HMI: device selection and indication of Programming Mode [12] - §3.13.2 “DMP_ProgModeSwitch_RCo” [03] - §4.19.3 “Programming Mode – Realisation Type 2”	[17] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”

2.7.1.4.5.1.3 Programming Mode – Autonomous Inactivation

Specification	Test
[03] - §4.19.1 “Autonomous inactivation of Programming Mode”	-

4) Implies connection-oriented TL and Application Layer services for accessing the Device Descriptor.

2.7.1.4.5.2 KNX Serial Number

2.7.1.4.5.2.1 Client initiated

Specification	Test
[12] - §2.4 “NM_IndividualAddress_Serial-Number_Read” - §2.5 “NM_IndividualAddress_Serial-Number_Write”	[17] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test” - §2.17 “Testing of A_IndividualAddress-SerialNumber_Read-Service : Server Test”

2.7.1.4.5.2.2 Server initiated

Specification	Test
[12] - §2.6 “NM_IndividualAddress_Serial-Number_Write2	[17] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test”

2.7.1.4.5.3 Domain Address Assignment

Specification	Test
[12] - §2.7 “NM_DomainAddress_Read” - §2.12 “NM_DomainAddress_Scan”	[17]

2.7.1.4.5.4 Default Individual Address

Specification	Test
[03] - §3.3 “Individual Address” - Subnetwork Address, value according to the medium. - Device Address, fixed value FFh	

2.7.1.4.6 Device Linking

In this clause all requirements on a device for linking of Group Objects are described. This includes the configuration of Address - and Association Tables.

Feature	Mask 2705h
1 Group Address Table	M
2 Group Object Association Table	M
3 Linking via Properties	O
4 Direct Link	O

2.7.1.4.6.1 Group Address Table

a) **Group Address Table – Realisation Type 8** • Mask 2705h	
Specification	**Test**
[03] - §4.9.8 “Group Address Table – Realisation Type 8”	

2.7.1.4.6.2 Group Object Association Table

a) **Group Address Table – Realisation Type 8**	
Specification	**Test**
[03] - §4.10.5 “Group Object Association Table – Realisation Type 8”	

2.7.1.4.7 Application Handling

In this clause all requirements necessary for an application program to be able to run on a target device are listed.

	Mask 2705h
1 Group Object Table	M
2 Application Program & Parameters	M
3 Application Specific Parameters	O
4 Application Programming Interface (API)	O
5 Functional Parameters	O
• unidirectional	n/a
• bidirectional	O

2.7.1.4.7.1 Group Object Table

The specification of the Group Object Table format for BIM M112 is currently under work in KSG. Please refer to the Working Document [25].
The Realisation Type number is not yet assigned.

a) **Group Object Table – Realisation Type ?** • System 7	
Specification	**Test**
[25] - §2.3.2 “Group Object Association Table - Realisation Type x”	

2.7.1.4.7.2 Application Program and Parameters

Feature	**Specification**	**Test**
• **pointer to user program**	[03]	
• **user program**		

2.7.1.4.7.3 Application specific system parameters

	Specification	Test
- user software version	[03]	

2.7.1.4.7.4 Application Programming Interface (API)

It is not required that mask 2705h complies with a standard API. An API is optional for mask 2705h.

2.7.1.5 Local device access

2.7.1.5.1 Introduction and general requirements

Local device access is optional for the mask 2705h.

Local devices access is only specified using cEMI, not for EMI1 or EMI2.

The below clauses concern the local devices access via the cEMI Transport Layer to the device. For the usability of the device as bus interface, the “Profile: Domain Address based RF Retransmitter” as specified in 2.7.3 has to be implemented additionally.

2.7.1.5.2 cEMI Profiles

2.7.1.5.2.1 Overview

	Mask 2705h
1. Transport Layer Interface	M
1. T_Data_Individual	M
2. T_Data_Connected	M
2. Local Device Management Services:	M
1. M_PropRead	M
2. M_PropWrite	M
3. M_PropInd	M
4. M_Reset.req	M
5. M_Reset.ind	M
6. M_FuncPropCommand.req	M
7. M_FuncPropCommand.con	M
8. M_FuncPropStateRead.req	M
9. M_FuncPropStateRead.res	M
10. M_Reset.req	M
11. M_Reset.ind	M

2.7.1.5.2.2 Transport Layer interface

Specification	Test
[13] - §4.1.6 “Transport Layer messages”	

2.7.1.5.3 Local device management services

Specification	Test
[13] - §4.1.7 “Services for local device management”	

2.7.1.5.4 Interface Objects and Properties

2.7.1.5.4.1 Overview

Interface Object	mask 2705h
0 Device Object	M
8 cEMI Server Object	O

2.7.1.5.4.2 Device Object

Property		mask 2705h
1 PID_OBJECT_TYPE	Data	3/1
71 PID_IO_LIST	Data	3/1

2.7.1.5.4.3 cEMI Server Object

Property		mask 2705h
1 PID_OBJECT_TYPE	Data	3/1
52 PID_COMM_MODE	Data	(3/2) a

[a] PID_COMM_MODE is mandatory if the cEMI Server can also be used for allowing the cEMI Client to communicate to the bus, e.g. in a KNX RF USB interface. In this case, the cEMI Server does not automatically switch to cEMI Transport Layer if the cEMI Client connects.

2.7.1.6 Interface Objects and Properties

2.7.1.6.1 Overview

S-Mode Profiles → End devices → Interface Objects

Mask 2705h requires the following Interface Objects. Additionally, form most Interface Objects, mandatory fix Object Indexes are required as specified below.

The Interface Object "Application Program 2" is the new name what was named "Interfaceprogram Object" before.

Interface Object	mask 2705h M/O	mask 2705h Object Index
0 Device Object	M	0
1 Addresstable Object	M	1
2 Association Table Object	M	2
3 Applicationprogram Object	M	3
4 Application Program 2	M	4
8 cEMI Server Object	O	- [a]
9 Group Object Table Object	O	- [a]
19 RF Medium Object	M	5

[a] No fix Object Index is required.

2.7.1.6.2 Device Object

S-Mode Profiles → End-devices → Device Object

Property		System 7	mask 2705h
1 PID_OBJECT_TYPE	Data	3/x	3/1
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
8 PID_SERVICE_CONTROL 5)	Data	(3/3)	(3/3)
9 PID_FIRMWARE_REVISION	Data	(3/x)	(3/1)
11 PID_SERIAL_NUMBER	Data	(3/x)	3/1
12 PID_MANUFACTURER_ID	Data	3/x	3/1

5) PID_SERVICE_CONTROL is a bitfield of which not all fields have to be supported. For the specification of which Profile shall support which field, please refer to 2.7.1.6.2.1.

Property		System 7	mask 2705h
14 PID_DEVICE_CONTROL	Data	3/3	3/3
15 PID_ORDER_INFO	Data	(3/3)	3/1
16 PID_PEI_TYPE 6)	Data	(3/3)	3/1
17 PID_PORT_CONFIGURATION	Data	(3/3)	(3/3)
19 PID_MANUFACTURER_DATA	Data	(3/3)	(3/1)
21 PID_DESCRIPTION	Data	(3/3)	(3/3)
25 PID_VERSION	Data	(3/3)	3/1
51 PID_ROUTING_COUNT	Data	(3/3)	n/a

2.7.1.6.2.1 PID_SERVICE_CONTROL (PID = 8)

S-Mode Profiles → End-devices → Device Object → PID_SERVICE_CONTROL

Bit#	Bit function	mask 2705hh
00	User Stopped_ServiceInfo Enable	O
01	OwnIndividual AddressReceived_ServiceInfo Enable	O
02	IndividualAddress_Write Enable	O
03	Reserved	O
04	Reserved	O
05	Reserved	O
06	Reserved	O
07	Reserved	O
08	Application Interface Layer Services on EMI Disable	O
09	Link Layer Services on EMI Disable	O
10	Network Layer Services on EMI Disable	O
11	Transport Layer Group Services on EMI Disable	O
12	Switch Service-Services on EMI Disable	O
13	Transport Layer Connection Oriented Services on EMI Disable	O
14	Application Layer Services on EMI Disable	O
15	Management Services on EMI Disable	O

6) PID_PEI_TYPE is mandatory for devices with PEI and supporting Properties.
7) PID_PEI_TYPE is only mandatory if a PEI is implemented.

2.7.1.6.3 Group Address Table Object (Object Type = 1)

S-Mode Profiles → End-devices → Group Address Table Object

	Property		System 7	mask 2705h
1	PID_OBJECT_TYPE	Data	3/x	3/x
2	PID_OBJECT_NAME	Data	(3/3)	(3/3)
5	PID_LOAD_STATE_CONTROL	Data	3/3	3/3
7	PID_TABLE_REFERENCE	Data	3/3	3/3
23	PID_TABLE	Data	(3/3)	(3/3)
27	PID_MCB_TABLE	Data	(3/3)	(3/x)
28	PID_ERROR_CODE	Data	(3/x)	(3/x)
53	PID_GROUP_RESPONDER_TABLE	Data	(3/3)	n/a

2.7.1.6.3.1 PID_LOAD_STATE_CONTROL (PID = 5)

Table 4 – Required Load Controls

Load Control	Sub-type	Description	System 7 mask 2705h
00h		No operation	M
01h		Start Loading	M
02h		Load Completed	M
03h		Additional Load Controls	M
	00h	Absolute Code/Data Allocation	M
	01h	Absolute Stack Allocation	
	02h	Segment Control Record	M
	03h	Task Pointer Record	M
	04h	Task Control Record-1	M
	05h	Task Control Record-2	M
	0Ah	Relative Allocation	n/a
	0Bh	Data Relative Allocation	n/a
04h		Unload	M

NOTE Table 4 specifies globally for the device which Load Controls shall be supported. Which Load Controls shall be supported for the management of a specific Resource is specified in the (Realisation Type of) the Resource in [03].

2.7.1.6.4 Association Table Object (Object Type = 2)

S-Mode Profiles → End-devices → Association Table Object

Property		System 7	mask 2705h
1 PID_OBJECT_TYPE	Data	3/x	3/x
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
5 PID_LOAD_STATE_CONTROL	Data	3/3	3/3
7 PID_TABLE_REFERENCE	Data	3/3	3/3
23 PID_TABLE	Data	(3/3)	(3/3)
27 PID_MCB_TABLE	Data	(3/3)	(3/3)
28 PID_ERROR_CODE	Data	(3/3)	(3/3)

2.7.1.6.4.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.1.6.3.1.

2.7.1.6.5 Applicationprogram Object (Object Type = 3)

S-Mode Profiles → End-devices → Application Object

Property		System 7	mask 2705h
1 PID_OBJECT_TYPE	Data	3/x	3/x
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
5 PID_LOAD_STATE_CONTROL	Data	3/3	3/3
6 PID_RUN_STATE_CONTROL	Data	3/3	3/3
7 PID_TABLE_REFERENCE	Data	3/x	3/x
13 PID_PROGRAM_VERSION	Data	3/(3)	3/(3)
16 PID_PEI_TYPE	Data	(3/(3) [7])	(3/(3)) [7]
27 PID_MCB_TABLE	Data	(3/3)	(3/3)
28 PID_ERROR_CODE	Data	(3/x)	(3/x)
51 PID_PARAM_REFERENCE	Data	(3/x)	(3/x)

2.7.1.6.5.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.1.6.3.1.

7) PID_PEI_TYPE is only mandatory if a PEI is implemented.

2.7.1.6.6 Application Program 2 Object (Object Type = 4)

S-Mode Profiles → End-devices → Application Program 2 Object

Property		System 7	mask 2705h
1 PID_OBJECT_TYPE	Data	3/x	3/x
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
5 PID_LOAD_STATE_CONTROL	Data	3/3	3/3
6 PID_RUN_STATE_CONTROL	Data	3/3	3/3
7 PID_TABLE_REFERENCE	Data	3/x	3/x
13 PID_PROGRAM_VERSION	Data	3/(3)	3/(3)
16 PID_PEI_TYPE	Data	(3/(3) 8)	(3/(3)) 7)
27 PID_MCB_TABLE	Data	(3/3)	(3/3)
28 PID_ERROR_CODE	Data	(3/x)	(3/x)
51 PID_PARAM_REFERENCE	Data	(3/x)	(3/x)

2.7.1.6.6.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.1.6.3.1.

2.7.1.6.7 RF Medium Object

S-Mode Profiles → Interfaces → RF Medium Object

Property		RF USB Interface 1
1 PID_OBJECT_TYPE	Data	3/1
51 PID_RF_MULTI_TYPE	Data	3/2
56 PID_RF_DOMAIN_ADDRESS	Data	3/2
58 PID_RF_FILTERING_MODE_SUPPORT	Data	(3/2)
59 PID_RF_FILTERING_MODE_SELECT	Data	(3/2)
60 PID_RF_BIDIR_TIMEOUT	Function	M
61 PID_RF_DIAG_SA_FILTER_TABLE	Data	3/3
62 PID_RF_DIAG_BUDGET_TABLE	Data	3/X
63 PID_RF_DIAG_PROBE	Function	M

8) PID_PEI_TYPE is only mandatory if a PEI is implemented.

2.7.2 Profile: mask 27B0h

2.7.2.1 Introduction and common requirements

The mask 27B0h shall be an implementation of the System B Profile class on the KNX RF Communication Medium.

2.7.2.2 Profile: Operation

2.7.2.2.1 Aims

This Profile describes the features of a device necessary for operation. The aim is to guarantee runtime Interworking between all devices in the system. The main components for this objective are the support of group oriented (multicast) communication.

2.7.2.2.2 Common Profile

These Profiles refer to all devices in all Configuration Modes on all media. Medium dependent features are described in clause 2.7.2.3.

Feature	mask 27B0h
01 Medium dependent Layers	KNX RF medium
02 Physical Layer General	M
03 Link Layer general	M
04 Link Layer - Router	n/a
05 Link Layer - Bridge	n/a
06 Network Layer general	M
07 Network Layer - Router	n/a
08 Network Layer – Bridge	n/a
09 Network Layer RF Retransmitter	O
10 Network Layer RF Media Coupler	X
11 Transport Layer – multicast	M
12 Transport Layer – connection oriented minimal	X
13 AL – Group Object services	M
14 AL – Property Value Services	M
15 AL – Function Property Services	M
16 AIL – GO	M
17 AIL – IO	M
18 AIL – Function Properties	M
19 AIL – GO indirection	O
20 Application Interface Layer for unidirectional devices	X

2.7.2.2.2.1 Medium dependent Layers

The Profiles defined in clause 2.7.2.3 “Medium dependent layers” of this document apply.

2.7.2.2.2.2 Physical Layer - general

Specification	Test
• **General**	
[06] (contains no requirement)	none.

2.7.2.2.2.3 Data Link Layer - general

Specification	Test
• **General**	
[07] - §1.1 “Functions of the Data Link Layer” - §1.2 “Possible Media and their Impact on Layer-2” - §1.3 “Objective”	tested with medium specific tests
• **Individual address /Group Address**	
[07] - §1.4 “Definitions”	tested with medium specific tests
• **Data Link Layer Protocol**	
[07] - §3 “Data Link Layer Protocols”	tested with medium specific tests
• **Parameters**	
[07] - §4 “Parameters of Layer-2” (except TP1 Fast Polling)	tested with medium specific tests

2.7.2.2.2.4 Network Layer - general

Specification	Test
• **General**	
[08] - §1 “Overview”	[15]
• **NPDU**	
[08] - §2.1 “NPDU”	[15] - §3 (Black Box Tests) - All end devices [14] - RF bidirectional end device
• **Parameters**	
[08] - §2.3 “Parameters of Network Layer” - hop_count: preferred value: 6.	- §3(Black Box Tests) - All end devices [14] - RF bidirectional end device
• **state machine**	
[08] - §2.4.1	[15] - §3 (Black Box Tests) - All end devices - [14] - RF bidirectional end device

2.7.2.2.2.5 Network Layer- RF Retransmitter

Specification	Test
[02] - §5.5.1 “History List” - §5.5.2 “RF Repeat Counter” - §5.5.3 “Filtering”	[14] - §3

2.7.2.2.2.6 Transport Layer–multicast

Specification	Test
• **TPDU**	
[09] - §1.2 “Point-to-Multipoint, Connectionless (Multicast) Communication Mode” - §3.1 “T_Data_Group Service”	[16] - All end devices **[14]** - RF bidirectional end device

2.7.2.2.2.7 Application Layer – Group Oriented

Specification	Test
• **APDU**	
[10] - §2 “APDU” o A_GroupValue_Read-PDU o A_GroupValue_Response-PDU o A_GroupValue_Write-PDU	[18] - All end devices [14] - RF bidirectional end device
• **Data length**	
[10] data must be coded as indicated in §3.1 “Application Layer Services on Multicast Communication Mode”	[18] - All end devices [14] - RF bidirectional end device

2.7.2.2.2.8 Application Layer – Property Value Services

Specification	Test
• **APDU**	
[10] - §2 “APDU” - §3.4.3.1 “A_PropertyValue_Read-service” - §3.4.3.2 “A_PropertyValue_Write-service”	

2.7.2.2.2.9 Application Layer – AL services on system broadcast communication mode

Unless indicated explicitly, the services shall be supported in full, this means, all service primitives shall be supported (.req, .Lcon, .ind, .res, .Rcon and .Acon).

Specification	Test
• **APDU**	
[10] - A_DomainAddress_Write - A_DomainAddress_Read - A_IndividualAddressSerialNumber_-Write - A_DomainAddressSerialNumber_Read - A_DomainAddressSerialNumber_Write - A_DomainAddressSelective_Read [24] - A_SystemNetworkParameter_Read	

2.7.2.2.2.10 Application Layer – AL services **on broadcast** communication mode

Unless indicated explicitly, the services shall be supported in full, this means, all service primitives shall be supported (.req, .Lcon, .ind, .res, .Rcon and .Acon).

Specification	Test
• **APDU**	
[10] - A_IndividualAddress_Write - A_IndividualAddress_Read - A_IndividualAddressSerialNumber_-Read - A_IndividualAddressSerialNumber_-Write - A_NetworkParameter_Read - A_NetworkParameter_Write	

2.7.2.2.2.11 Application Layer – Function Services

• **APDU**	
[10] - §2 “APDU” - §3.4.5 “Function Property Services”	

2.7.2.2.2.12 Application Interface Layer – Group Objects

Specification	Test
• **Group Objects**	
[11] - §3 “Group Object Server”	[17] - §1.3 “Group Object Tests” - all end devices [14] - - RF bidirectional end device

2.7.2.2.2.13 Application Interface Layer – Interface Objects

Specification	Test
• **Interface Objects**	
[11] - §4 “Interface Object Server”	

2.7.2.2.2.14 Application Interface Layer – Function Properties

Specification	Test
• **Function Properties**	
[11] - §4.4.2 “Function Properties”	

2.7.2.2.2.15 Application Layer Interface Layer – Group Objects indirection

Specification	Test
• **Group Object Indirection**	
[11] - §3.4 “Group Object Indirection – Group Object Handles and PID-OBJECT_-VALUE (PID = 62) [03] - §4.3.13 “PID_OBJECT_VALUE (PID = 62)”	

2.7.2.2.3 Specific parts

These Profiles describe runtime mechanisms that are not relevant for all devices.

Feature	mask 27B0h
01 Link Layer – polling	n/a
02 Detection of Usage of own Individual Address	X
03 Extended Group Object Flags	M

2.7.2.2.3.1 Extended Group Object Flags

Specification	Test
• **configuration flags: update on read response (1 bit)**	
[11] - §3.3.2 “Reading the Group Object Value ” (Update Enable)	[17] - §1.3 (Group Object Tests)

2.7.2.3 Medium dependent layers

2.7.2.3.1 Goal

This Profile describes the requirements on a device in order to guarantee compliance with one of the standard communication media of the system. Compliance with one of these Profiles is a prerequisite for both runtime and configuration Interworking.

2.7.2.3.2 RF medium dependent layers

	mask 27B0h
1. Physical Layer Sender	X
2. Physical Layer Transceiver	M
3. Link Layer	M
4. KNX Serial Number for multicast communication	O
5. RF Domain Address for multicast communication	M
6. LTE frame format	O
7. Extended LTE Group Addresses and frame acceptance	O
8. Link Layer-Retransmitter	O
9. Link Layer-Media Coupler	O
10. Local Services Metering	O
11. KNX Serial Number	M
12. APDU-length	≥ 15

2.7.2.3.2.1 Physical Layer Sender

Specification	Test
[02] - §5.1 “Physical Layer for KNX RF Ready and BiBat” - §5.2 “Datagram structure for RF Ready and BiBat”	

2.7.2.3.2.2 Physical Layer Transceiver

Specification	Test
[02] - §5.1 “Physical Layer for KNX RF Ready and BiBat” - §5.2 “Datagram structure for RF Ready and BiBat”	

The S-Mode Configuration Procedures may result in a lot of Telegrams on the KNX RF Medium. For this, it is strongly recommended, both for the MaS as for the S-Mode MaC, to apply Listen Before Talk (LBT, see [02]), to avoid RF collision. LBT is part of the KNX RF Multi type, but with S-Mode Configuration Procedures, the Telegram rate is higher than the duty cycle requirements for the KNX RF Ready type. Therefore, in order not to limit the Telegram rate, LBT is a recommended solution.

2.7.2.3.2.3 Link Layer

	Specification	Test
	• **Frame format**	
[02]	- §5.1.2 “Datagram structure for RF Ready and BiBat	
	• **Medium Access Control**	
[02]	- §5.1.3 “Medium access”	
	• **Addressing**	
[02]	- §6.1.1.1 “Extended Group Address” – point-to-multipoint – DoA of sender - §6.1.4.2 “Duplication prevention”	

The reference to clause 6.1.1.1 is actually a forward reference to where this will be specified after integration of this document. For the time being, what is referred to here as clause 6.1.1.1 is actually specified in clause 2.2.1.1 of this document.

2.7.2.3.2.4 KNX Serial Number for multicast communication

	Specification	Test
[02]	- §6.1.1.1 “Extended Group Address”	

Please note that the contents of the referred clause will be modified by this document.

2.7.2.3.2.5 RF Domain Address for multicast communication

	Specification	Test
[02]	- §6.1.1.4 “RF Domain Address”	

Please note that the contents of the referred clause will be modified by this document.

2.7.2.3.2.6 LTE frame format

	Specification	Test
[02]	- §6.1.2.4 “First block” - §6.1.2.6 “Second block for LTE Extended Datagrams”	

2.7.2.3.2.7 Extended LTE Group Addresses and frame acceptance

	Specification	Test
[02]	- §6.1.5.3.1 “Additional Frame acceptance criteria for LTE Extended Datagrams”	

2.7.2.3.2.8 Link Layer-Retransmitter

	Specification	Test
[02]	- §6.2.2 “The Layer-2 of an RF Retransmitter”	

2.7.2.3.2.9 Link Layer-Media Coupler

	Specification	Test
[02]	- §6.1.6 “The Layer-2 of an RF-TP Media Coupler”	

2.7.2.3.2.10 Local Services Metering

	Specification	Test
[13]	- §2 “Message format” – code for L_Meter.ind - §3.3.3.5 “L_Meter.ind”	

2.7.2.3.2.11 KNX Serial Number

	Specification	Test
[03]	- §4.15.1 “Abstract Resource definition” - §4.15.2 “KNX Serial Number – Realisation Type 1”	

2.7.2.4 Configuration & Management (S-Mode, Server)

This clause describes the requirements on an S-Mode device, which are relevant for configuration as a Management Server accessed only via the bus. The objective is to guarantee Interworking with the configuration tool (ETS).

2.7.2.4.1 Communication

	Feature	Mask 27B0h
01	TL - broadcast	M
02	TL - connection oriented	M
03	TL - connection oriented minimal	X
04	TL - connectionless	M

2.7.2.4.1.1 TL - broadcast

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives. - §1.3 “Point-to-all-Points Connectionless (Broadcast) Communication Mode” - §2 “TPDU” - §3.4 “T_Data_Broadcast” - §4 “Parameters of Transport Layer”	

2.7.2.4.1.2 TL - connection oriented

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives. - §1.6 “Point-to-Point, Connection-Oriented Communication Mode” - §2 “TPDU” - §3.7 “T_Connect service” - §3.8 “T_Disconnect service” - §3.9 “T_Data_Connected service” - §4 “Parameters of Transport Layer” - §5.1 “States” - §5.2 “Actions” - §5.3.3 “Style 3”	[16]

2.7.2.4.1.3 TL - connectionless

	Specification	Test
[09]	All features of the following clauses are mandatory except for the coding of the internal service primitives: - §1.5 “Point-to-point connectionless Communication Mode” - §2 “TPDU” - §3.6 “T_Data_Individual”	[16]

2.7.2.4.2 Device Management

In this clause all general requirements on a device concerning the mechanisms used for access by the Management Client are described.

	Feature	mask 27B0h
01	Direct memory Access	M
02	DMA on User Memory	M
03	Verify mode [9)]	M
04	Interface Object Handling [10)]	M
05	Reduced Interface Objects	X
06	Function Properties	M
07	Load and Run State Machines	
08	a. Realisation Type 1	M
09	b. Realisation Type 2	O
10	Hardware Specific Parameters	n/a
11	RAM cleared	n/a
12	User EEPROM	n/a
13	Restart	
14	a. connectionless	O
15	b. connection-oriented	M
16	c. Master Reset	O
17	Authorization [11)]	M
18	nr of access levels	4
19	Memory Control Table	O

9) If Verify Mode is not implemented, it shall always be off.

10) Please refer to the specification of mandatory and optional Interface Objects, Properties and Property fields further in this Profile definition.

11) The support of the A_Authorize- and the A_Keywrite-service does not imply that the device itself has access protected areas. If this is not the case, a device shall always allow – regardless of the attributed keys – access to the highest level (0), including when receiving an illegal key ('illegal' in this sense meaning another key than any of the keys entered in the key table).

2.7.2.4.2.1 Direct Memory Access

Specification	Test
[12] - §3.2.1 “DMP_Connect_RCo” - §3.3.2 “DMP_Disconnect_RCo” - §3.18.2 “DMP_MemRead_RCo” - §3.16.2 “DMP_MemWrite_RCo”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.2.4.2.2 DMA on user memory

Specification	Test
[12] - §3.19 “DM_UserMemWrite”	[17] - §2 “Network Management Server Tests corresponding tests

2.7.2.4.2.3 Verify Mode (for A_Memory_Write)

Specification	Test
[03] - §4.2.14.7 "Verify Mode Control" management server part.	[17] - §2 ”Network Management Server Tests” Tests 7.1 to 7.7

2.7.2.4.2.4 Interface Object Handling

Please also refer to the specification of mandatory and optional Interface Objects, Properties and Property fields further in this Profile definition.

Specification	Test
[12] - §3.22.2 “DMP_InterfaceObjectWrite_R” - §3.24.2 “DM_InterfaceObjectRead_R” - §3.23.2 “DMP_InterfaceObjectVerify_R” - §3.25.2 “DM_InterfaceObjectScan_R”	[17] - §2 “Network Management Server Tests” corresponding tests
[11] - §4 “Interface Object Server”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.2.4.2.5 Reduced Interface Objects

Specification	Test
[12] - §3.22.2 “DMP_ReducedInterface-ObjectWrite_R” - §3.24.3 “DMP_ReducedInterfaceObject-Read_R” - §3.25.3 “DMP_ReducedInterfaceObject-Scan_R”	[17] - §2 “Network Management Server Tests” corresponding tests
[11] - §4.1 “Common structure” - §4.3.2 “Reduced Interface Object”	[17] - §2 “Network Management Server Tests” corresponding tests

2.7.2.4.2.6 Function Properties

Specification	Test
[10] - §2 “APDU” - §3.4.1 “Introduction” - §3.4.5 “Function Property Services”	
[11] - §4.1 “Common structure” - §4.2 “Minimal requirements of Interface Objects” - §4.4.2 “Function Properties”	
[12] - §3.26 “DM_FunctionProperty_Write_R”	

2.7.2.4.2.7 Load - and Run State Machines

a) Realisation Type 1 - Property based	
Specification	**Test**
[03] - records	
[12] - §3.27.3 “DMP_LoadStateMachine-Write_Rco_IO” - §3.28.3 “DM_LoadStateMachine-Verify_R_IO” - §3.29.3 “DMP_LoadStateMachineRead_R_IO” - §3.30.3 “DMP_RunStateMachineWrite_R_IO” - §3.31.3 “DMP_RunStateMachineVerify_R_IO” - §3.32.3 “DMP_RunStateMachineRead_R_IO”	[17] - §2 “Network Management Tests” corresponding tests
b) Realisation Type 2 – Memory mapped	
Specification	**Test**
[03] records	
[12] - §3.27.2 “DMP_LoadStateMachineWrite_Rco_Mem” - §3.28.2 “DMP_LoadStateMachineVerify_Rco_Mem” - §3.29.2 “DMP_LoadStateMachineRead_Rco_Mem” - §3.30.2 “DMP_RunStateMachineWrite_Rco_Mem” - §3.31.2 “DMP_RunStateMachineVerify_Rco_Mem” - §3.32.2 “DMP_RunStateMachineRead_Rco_Mem”	[17] - §2 “Network Management Tests” corresponding tests

2.7.2.4.2.8 Hardware specific Parameters

	Specification	Test
	• **0100h (RW)**	
[03]	- §4.18 “OptionReg”	

2.7.2.4.2.9 RAM cleared

System B does not have any standard memory organisation or standard memory locations, so this feature is optional.

2.7.2.4.2.10 User EEPROM

System B does not have any standard memory organisation or standard memory locations, so this feature is optional.

2.7.2.4.2.11 Restart

2.7.2.4.2.11.1 Restart connectionless

	Specification	Test
[12]	- §3.7.2 DM_Restart_RCl	

2.7.2.4.2.11.2 Restart connection-oriented

	Specification	Test
[12]	- §3.2.1 DMP_Connect_RCo - §3.7.3 DM_Restart_RCo	- §2.9.1 “Send Restart”

2.7.2.4.2.11.3 Master Reset

Master Reset is optional for the mask 2705h. However, as KNX RF is an open medium, it would be possible for any hacker to set a KNX RF S-Mode device out of operation by an unprotected Master Reset. Therefore, it is recommended that if Master Reset is implemented, that it only be accepted from an authenticated communication partner, using the A_Authorize-service or KNX Data Security (see [21]).

	Specification	Test
[03]	- §4.9.1 “Group Address Table” – “default state” - §4.10.1 “Group Object Association Table” – “default state”	To be completed.
[12]	- §3.7.2 “DM_Restart_RCl” – master reset - §3.7.3 “DM_Restart_RCo” – master reset	To be completed.

2.7.2.4.2.12 Authorization

	Specification	Test
	• **Management Procedures**	
[12]	- §3.5.1 “DM_Authorize_RCo” - §3.5.2 “DM_Authorize2_RCo” - §3.6 “DM_SetKey”	

2.7.2.4.2.13 Memory Control Table

Specification	Test
• **PID_MCB_TABLE**	
[03] - §4.2.27 “PID_MCB_TABLE” (PID = 27) as read-only.	

2.7.2.4.3 Device Discovery

Feature	Mask 2705h
1 Management Procedures with A_SystemNetwork-Parameter_Read-PDU	
1.1 NM_Read_SerialNumber_By_ProgrammingMode	M
1.2 NM_Read_SerialNumber_By_ExFactoryState	M
1.3 NM_Read_SerialNumber_By_PowerReset	M

2.7.2.4.4 Device Identification

Feature	Mask 27B0h
01 Device Descriptor Service - connection oriented	M
02 Device Descriptor Service - connectionless	M
03 Device Descriptor Type 0	M
04 Device Descriptor Type 2	O
05 Device Descriptor InfoReport	O
06 Management Descriptor 1	O
07 Identification of hardware	M
08 Identification of Application	M

2.7.2.4.4.1 Device Descriptor Service - connection oriented

Specification	Test
[10] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.7.2.4.4.2 Device Descriptor Service - connectionless

Specification	Test
[10] - §3.4.2.1 “A_DeviceDescriptor_Read-service”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)
[12] - §3.2.2 “DMP_Connect_RCl”	

2.7.2.4.4.3 Device Descriptor Type 0

Specification	Test
[03] - §4.1.2 “Device Descriptor Type 0”	[17] - §2.5.1 “Read Mask-version” (Network Management Tests)

2.7.2.4.4.4 Management Descriptor 1

Specification	Test
• **Management Descriptor 1**	
[03] - §4.3.23 “PID_MGT_DESCRIPTOR (PID = 72)”	
[12] - §3.4.2 “DM_Identify_R”	

2.7.2.4.4.5 Identification of the hardware

Specification	Test
[12] - §3.4.3 “DM_Identify_RCo2”	

2.7.2.4.4.6 Identification of Application

Specification	Test
[10] - §3.5.6.5 "A_UserManufacturerInfo_-Read_service"	

2.7.2.4.5 Device Individualisation

In this clause all requirements on a device for device individualisation and assignment of the Individual Address are described.

	Feature	Mask 27B0h
01	Programming Mode	
	1.a Connection oriented	M
	1.b Connectionless	O
	Autonomous Inactivation	M
02	KNX Serial Number	
	a client initiated	M
	b server initiated	O
03	Domain Address Assignment	M
04	Default Individual Address	O

2.7.2.4.5.1 Programming Mode

2.7.2.4.5.1.1 connection oriented

a) Realisation Type 1 - Property based		
	Specification	**Test**
[12]	- §2.2 “NM_IndividualAddress_Read” - §2.3 “NM_IndividualAddress_Write” 12)	
	Programming Mode Control • via HMI: device selection and indication of Programming Mode • via bus: [03] - §4.3.5 “PID_PROGMODE” [12] - §3.22.2 “DMP_InterfaceObject-Write_R” - §3.23.2 “DMP_InterfaceObject-Verify_R” - §3.24.2 “DMP_InterfaceObject-Read_R”	

12) Implies connection-oriented TL and Application Layer services for accessing the Device Descriptor.

2.7.2.4.5.1.2 Programming Mode – connectionless

Specification	Test
[12] - §2.10 “NM_DomainAnd-IndividualAddress_Write2” - §2.2 “NM_IndividualAddress_Read”	[17] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”
Programming Mode Control • via HMI: device selection and indication of Programming Mode [12] - §3.13.2 “DMP_ProgModeSwitch_RCo” [03] - §4.19.3 “Programming Mode – Realisation Type 2”	[17] - §2.3 “Testing of A_IndividualAddress-_Read-service – Server Test”

2.7.2.4.5.1.3 Programming Mode – Autonomous Inactivation

Specification	Test
[03] - §4.19.1 “Autonomous inactivation of Programming Mode”	-

2.7.2.4.5.2 KNX Serial Number

2.7.2.4.5.2.1 Client initiated

Specification	Test
[12] - §2.4 “NM_IndividualAddress_Serial-Number_Read” - §2.5 “NM_IndividualAddress_Serial-Number_Write”	[17] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test” - §2.17 “Testing of A_IndividualAddress-SerialNumber_Read-Service : Server Test”

2.7.2.4.5.2.2 Server initiated

Specification	Test
[12] - §2.6 “NM_IndividualAddress_Serial-Number_Write2	[17] - §2.16 “Testing of A_IndividualAddress-SerialNumber_Write-Service : Server Test”

2.7.2.4.5.3 Domain Address Assignment

Specification	Test
[12] - §2.7 “NM_DomainAddress_Read” - §2.12 “NM_DomainAddress_Scan”	[17]

2.7.2.4.5.4 Default Individual Address

Specification	Test
[03] - §3.3 “Individual Address” - Subnetwork Address, value according to the medium. - Device Address, fixed value FFh	

2.7.2.4.6 Device Linking

In this clause all requirements on a device for linking of Group Objects are described. This includes the configuration of Address - and Association Tables.

Feature	Mask 27B0h
01 Group Address Table	M
02 Group Object Association Table	M
03 Linking via Properties	O
04 Direct Link	O

2.7.2.4.6.1 Group Address Table

b) Group Address Table – Realisation Type 8 • Mask 5705h	
Specification	**Test**
[03] - §4.9.7 “Group Address Table – Realisation Type 7”	

2.7.2.4.6.2 Group Object Association Table

b) Group Object Association Table – Realisation Type 6	
Specification	**Test**
[03] - §4.10.4 “Group Object Association Table – Realisation Type 6”	

2.7.2.4.7 Application Handling

In this clause all requirements necessary for an application program to be able to run on a target device are listed.

	Mask 27B0h
01 Group Object Table	M
02 Application Program & Parameters	M
03 Application Specific Parameters	M
04 Application Programming Interface (API)	O
05 Functional Parameters	O
• unidirectional	n/a
• bidirectional	O

2.7.2.4.7.1 Group Object Table

b) Group Object Table – Realisation Type 7 • System B	
Specification	**Test**
[03] - §4.15.5 “Group Object Table – Realisation Type 7”	

2.7.2.4.7.2 Application Program and Parameters

Feature	Specification	Test
- pointer to user program - user program	[03]	

2.7.2.4.7.3 Application specific system parameters

Feature	Specification	Test
- user software version	[03]	Application Layer Tests, System B Test specification

2.7.2.4.7.4 Application Programming Interface (API)

- System B: No standard API is specified.

2.7.2.5 Local device access

2.7.2.5.1 Introduction and general requirements

Local device access is optional for the mask 27B0h.

Local devices access is only specified using cEMI, not for EMI1 or EMI2.

The below clauses concern the local devices access via the cEMI Transport Layer to the device. For the usability of the device as bus interface, the “Profile: Domain Address based RF Retransmitter” as specified in 2.7.3 has to be implemented additionally.

2.7.2.5.2 cEMI Profiles

2.7.2.5.2.1 Overview

	Mask 27B0h
1. Transport Layer Interface	M
1. T_Data_Individual	M
2. T_Data_Connected	M
2. Local Device Management Services:	M
1. M_PropRead	M
2. M_PropWrite	M
3. M_PropInd	M
4. M_Reset.req	M
5. M_Reset.ind	M
6. M_FuncPropCommand.req	M
7. M_FuncPropCommand.con	M
8. M_FuncPropStateRead.req	M
9. M_FuncPropStateRead.res	M
10. M_Reset.req	M
11. M_Reset.ind	M

2.7.2.5.2.2 Transport Layer interface

Specification	Test
[13] - §4.1.6 “Transport Layer messages”	

2.7.2.5.3 Local device management services

Specification	Test
[13] - §4.1.7 “Services for local device management”	

2.7.2.5.4 Interface Objects and Properties

2.7.2.5.4.1 Overview

Interface Object	mask 27B0h
0 Device Object	M
8 cEMI Server Object	O

2.7.2.5.4.2 Device Object

Property		mask 27B0h
1 PID_OBJECT_TYPE	Data	3/1
71 PID_IO_LIST	Data	3/1

2.7.2.5.4.3 cEMI Server Object

Property		mask 27B0h
1 PID_OBJECT_TYPE	Data	3/1
52 PID_COMM_MODE	Data	(3/2) a

[a] PID_COMM_MODE is mandatory if the cEMI Server can also be used for allowing the cEMI Client to communicate to the bus, e.g. in a KNX RF USB interface. In this case, the cEMI Server does not automatically switch to cEMI Transport Layer if the cEMI Client connects.

2.7.2.6 Interface Objects and Properties

2.7.2.6.1 Overview

S-Mode Profiles → End devices → Interface Objects

Mask 27B0h requires the following Interface Objects. Additionally, form most Interface Objects, mandatory fix Object Indexes are required as specified below.

Interface Object	mask 27B0h M/O	mask 27B0h Object Index
0 Device Object	M	0
1 Addresstable Object	M	1
2 Association Table Object	M	2
3 Applicationprogram Object	M	4
4 Application Program 2 Object	M	5
8 cEMI Server Object	O	- [a]
9 Group Object Table Object	M	3
19 RF Medium Object	M	6

[a] No fix Object Index is required.

2.7.2.6.2 Device Object

S-Mode Profiles → End-devices → Device Object

Property		System B	mask 27B0h
1 PID_OBJECT_TYPE	Data	3/x	3/1
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
8 PID_SERVICE_CONTROL 13)	Data	(3/3)	(3/3)
9 PID_FIRMWARE_REVISION	Data	(3/x)	(3/1)
11 PID_SERIAL_NUMBER	Data	3/x	3/1
12 PID_MANUFACTURER_ID	Data	3/x	3/1
14 PID_DEVICE_CONTROL	Data	3/3	3/3
15 PID_ORDER_INFO	Data	3/x	3/1
16 PID_PEI_TYPE 14)	Data	3/x	3/1
17 PID_PORT_CONFIGURATION	Data	(3/3)	(3/3)
19 PID_MANUFACTURER_DATA	Data	(3/x)	(3/1)
21 PID_DESCRIPTION	Data	(3/3)	(3/3)
25 PID_VERSION	Data	3/x	3/1

13) PID_SERVICE_CONTROL is a bitfield of which not all fields have to be supported. For the specification of which Profile shall support which field, please refer to 2.7.1.6.2.1.

14) PID_PEI_TYPE is mandatory for devices with PEI and supporting Properties.

15) This Property is mandatory for PL110 devices. For all other media, this Property shall not be implemented.

Property		System B	mask 27B0h
51 PID_ROUTING_COUNT	Data	3/3	n/a

2.7.2.6.2.1 PID_SERVICE_CONTROL (PID = 8)

S-Mode Profiles → End-devices → Device Object → PID_SERVICE_CONTROL

		System B
Bit#	Bit function	mask 27B0h
00	User Stopped_ServiceInfo Enable	O
01	OwnIndividual AddressReceived_ServiceInfo Enable	O
02	IndividualAddress_Write Enable	M
03	Reserved	O
04	Reserved	O
05	Reserved	O
06	Reserved	O
07	Reserved	O
08	Application Interface Layer Services on EMI Disable	O
09	Link Layer Services on EMI Disable	O
10	Network Layer Services on EMI Disable	O
11	Transport Layer Group Services on EMI Disable	O
12	Switch Service-Services on EMI Disable	O
13	Transport Layer Connection Oriented Services on EMI Disable	O
14	Application Layer Services on EMI Disable	O
15	Management Services on EMI Disable	O

2.7.2.6.3 Group Address Table Object (Object Type = 1)

S-Mode Profiles → End-devices → Group Address Table Object

Property			System B	mask 27B0h
1	PID_OBJECT_TYPE	Data	3/x	3/x
2	PID_OBJECT_NAME	Data	(3/3)	(3/3)
5	PID_LOAD_STATE_CONTROL	Data	3/(3)	3/(3)
7	PID_TABLE_REFERENCE	Data	3/x	3/x
23	PID_TABLE	Data	3/(3)	3/(3)
27	PID_MCB_TABLE	Data	(3/3)	(3/3)
28	PID_ERROR_CODE	Data	(3/x)	(3/x)
53	PID_GROUP_RESPONDER_TABLE	Data	3/3[15]	X

2.7.2.6.3.1 PID_LOAD_STATE_CONTROL (PID = 5)

Table 5 – Required Load Controls

Load Control	Sub-type	Description	System B mask 27B0h
00h		No operation	O
01h		Start Loading	M
02h		Load Completed	M
03h		Additional Load Controls	M
	00h	Absolute Code/Data Allocation	O
	01h	Absolute Stack Allocation	O
	02h	Segment Control Record	O
	03h	Task Pointer Record	O
	04h	Task Control Record-1	O
	05h	Task Control Record-2	O
	0Ah	Relative Allocation	0
	0Bh	Data Relative Allocation	M
04h		Unload	M

NOTE Table 5 specifies globally for the device which Load Controls shall be supported. Which Load Controls shall be supported for the management of a specific Resource is specified in the (Realisation Type of) the Resource in [03].

15) This Property is mandatory for PL110 devices. For all other media, this Property shall not be implemented.

2.7.2.6.4 Association Table Object (Object Type = 2)

S-Mode Profiles → End-devices → Association Table Object

Property		System B	mask 27B0h
1 PID_OBJECT_TYPE	Data	3/x	3/x
2 PID_OBJECT°NAME	Data	(3/3)	(3/3)
5 PID_LOAD_STATE_CONTROL	Data	3/(3)	3/(3)
7 PID_TABLE_REFERENCE	Data	3/x	3/x
23 PID_TABLE	Data	3/(3)	3/(3)
27 PID_MCB_TABLE	Data	(3/3)	(3/3)
28 PID_ERROR_CODE	Data	(3/x)	(3/x)

2.7.2.6.4.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.2.6.3.1.

2.7.2.6.5 Applicationprogram Object (Object Type = 3)

S-Mode Profiles → End-devices → Application Object

Property		System B	mask 27B0h
1 PID_OBJECT_TYPE	Data	3/x	3/x
2 PID_OBJECT_NAME	Data	(3/3)	(3/3)
5 PID_LOAD_STATE_CONTROL	Data	3/3	3/3
6 PID_RUN_STATE_CONTROL	Data	(3/3)	(3/3)
7 PID_TABLE_REFERENCE	Data	3/x	3/x
13 PID_PROGRAM_VERSION	Data	3/3	3/3
16 PID_PEI_TYPE	Data	3/3	3/3
27 PID_MCB_TABLE	Data	(3/3)	(3/3)
28 PID_ERROR_CODE	Data	(3/x)	(3/x)
51 PID_PARAM_REFERENCE	Data	(3/x)	(3/x)

2.7.2.6.5.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.2.6.3.1.

2.7.2.6.6 Applicationprogram Object (Object Type = 4)

S-Mode Profiles → End-devices → Application Program 2 Object

Property			System B	mask 27B0h
1	PID_OBJECT_TYPE	Data	3/x	3/x
2	PID_OBJECT_NAME	Data	(3/3)	(3/3)
5	PID_LOAD_STATE_CONTROL	Data	3/3	3/3
6	PID_RUN_STATE_CONTROL	Data	(3/3)	(3/3)
7	PID_TABLE_REFERENCE	Data	3/x	3/x
13	PID_PROGRAM_VERSION	Data	3/3	3/3
16	PID_PEI_TYPE	Data	3/3	3/3
27	PID_MCB_TABLE	Data	(3/3)	(3/3)
28	PID_ERROR_CODE	Data	(3/x)	(3/x)
51	PID_PARAM_REFERENCE	Data	(3/x)	(3/x)

2.7.2.6.6.1 PID_LOAD_STATE_CONTROL (PID = 5)

Please refer to 2.7.2.6.3.1.

2.7.2.6.7 Group Object Table Object (Object Type = 9)

S-Mode Profiles → End-devices → Group Object Table Object

Property			System B	mask 27B0h
1	PID_OBJECT_TYPE	Data	3/x	3/x
2	PID_OBJECT_NAME	Data	(3/3)	(3/3)
5	PID_LOAD_STATE_CONTROL	Data	3/(3)	3/(3)
7	PID_TABLE_REFERENCE	Data	3/x	3/x
23	PID_TABLE	Data	3/(3)	3/(3)
27	PID_MCB_TABLE	Data	(3/3)	(3/3)
28	PID_ERROR_CODE	Data	(3/x)	(3/x)
51	PID_GRP_OBJTABLE	Data	(3/3)	(3/3)
52	PID_EXT_GRPOBJREFERENCE	Data	(3/3)	(3/3)

2.7.2.6.8 RF Medium Object

S-Mode Profiles → Interfaces → RF Medium Object

	Property		RF USB Interface 1
1	PID_OBJECT_TYPE	Data	3/1
51	PID_RF_MULTI_TYPE	Data	3/2
56	PID_RF_DOMAIN_ADDRESS	Data	3/2
58	PID_RF_FILTERING_MODE_SUPPORT	Data	(3/2)
59	PID_RF_FILTERING_MODE_SELECT	Data	(3/2)
60	PID_RF_BIDIR_TIMEOUT	Function	M
61	PID_RF_DIAG_SA_FILTER_TABLE	Data	3/3
62	PID_RF_DIAG_BUDGET_TABLE	Data	3/X
63	PID_RF_DIAG_PROBE	Function	M

2.7.3 Profile: Domain Address based RF Retransmitter

2.7.3.1 Introduction and common requirements

The Domain Address based RF Retransmitter is not a device Profile, but is a Profile for functionality that can be added to any KNX RF S-Mode Profile. The below clauses do therefore not specify a full Profile, but only indicate the features that are required for this Module Profile.

2.7.3.2 Common Profile

2.7.3.2.1.1 Overview

	Feature	DoA Based RF Retransmitter
1	Network Layer – Domain Address based RF Retransmitter	M

2.7.3.2.1.2 Physical Layer - general

	Specification		Test
	- 2.2.1.3 “RF Domain Address based RF Retransmitter””		- to be defined

2.7.3.3 Medium dependent layers

2.7.3.3.1 Goal

This Profile describes the requirements on a device in order to guarantee compliance with one of the standard communication media of the system. Compliance with one of these Profiles is a prerequisite for both runtime and configuration Interworking.

2.7.3.3.2 RF medium dependent layers

	DoA Based RF Retransmitter
1. Link Layer-Retransmitter 2. APDU-length	M ≥ 55

NOTE 9 The Profile “Domain Address based RF Retransmitter” is not a full device Profile but a Profile Module, this is, it is only an extension to a full device Profile. The APDU-length that is indicated here shall be the minimal APDU-length that the implementation shall be able to retransmit. For the specification of the APDU-length that the device shall support for its own runtime communication and management, please refer to the respective RF device Profiles.

2.7.3.3.2.1 Link Layer-Retransmitter

Specification	Test
[02] - §6.2.2 “The Layer-2 of an RF Retransmitter”	

2.7.3.4 Interface Objects

2.7.3.4.1 Overview

Interface Object	DoA Based RF Retransmitter
0 Device Object	M
19 RF Medium Object	M

2.7.3.4.2 Device Object

	Property		DoA Based RF Retransmitter
1	PID_OBJECT_TYPE	Data	3/1

2.7.3.4.3 RF Medium Object

	Property		DoA Based RF Retransmitter
1	PID_OBJECT_TYPE	Data	3/1
57	PID_RF_RETRANSMITTER	Data	3/2

2.7.4 KNX RF USB Interface 1

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

The specification of the KNX RF USB Interface 1 is no longer contained in this paper and will be specified in a separate KSG paper.

One open topic is the identification of the type of interface, as discussed in the KSG meeting of 2013.04.19.

2.7.5 Not more supported

- Address 60h is not used anymore to switch programming mode OFF. => Update test for adding that no reaction of device is done when writing this address.
  PID_PROGRAMMING_MODE is now used for this Profile.
  There will be no negative tests on this.

2.8 RF Multi device handling by ETS

In the first version ETS will only handle RF ready Devices. ETS shall switch all RF devices TP/RF Multi Media Coupler to RF ready by writing PID_MULTI_TYPE to 0.

2.9 Identifiers and discovery

3 Impact and dependencies

3.1 System specification (“Handbook”) dependencies

This clause is itself not considered for integration in the KNX Specifications.

Indications about the modification of the existing KNX Specifications and the integration of new features are marked through appropriate editor notes in the various clauses in this document.

Additionally, the following shall be done.

Doc.	Modification
[05]	**Feature “Autonomous Inactivation of Programming Mode”** This feature shall be added as “optional” to all other Profiles.
[03], [04], [05]	**Interface Object Type “Interfaceprogram**” This will for all Profiles be renamed to “Application Program 2”. In the specification in [03], a note shall be added that this Interface Object type was named “Interfaceprogram” before.

3.2 On Push-Button and link services

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

3.2.1 Introduction

This clause gives considerations for a future definition of a Ctrl-Mode flavour that may support Domain Addresses on KNX RF. This clause may serve as possible input for a next definition of FEC Profiles.

For Ctrl-Mode or PB-Mode, it is also possible to configure extended Group Addresses based on an RF Domain Address.

To assign a link based on a Domain Address and a Group Address, the Function Property PID_OBJECTLINK (PID = 63) can be used. The usage of the DoA can be indicated via a coding of the flags that already exist in the telegram format.

The below is an extract from AN134 “Flexible E-Mode Channels” ([19]) and is a proposal of how PID_OBJECTLINK may be modified to support Domain Addresses for group communication on KNX RF.

3.2.2 E-Mode Channel Object – PID_OBJECTLINK (PID = 63)

- Property name: Object Link
- Property Datatype: PDT_FUNCTION
- Datapoint Type: None.
-

Used by: Ctrl FEC PB FEC

3.2.2.1 Function Write Object Link

3.2.2.1.1 Format

octet 10	octet 11	octet 12	octet 13	octet 14	octet 15	octet 16	octet 17	octet 18	octet 19	octet 20	octet 21
Flags	00h	SN (high)	SN	SN	SN	SN	SN (low)	GA (high)	GA (low)	handle (hi)	handle (lo)
		Extended Group Address									

Figure 5 – Function *Write Object Link* A_FunctionPropertyCommand-PDU (example)

3.2.2.1.2 Usage by the Management Client

The Management Client shall use the Function *Write Object Link* to establish and to break a single link between a Group Object and a Group Address or an Extended Group Address.

1. Flags

Description: This field shall indicate whether a link to the referred Group Object shall be established or shall be broken. In case the link shall be added, it shall also indicate whether the link shall be a sending Group Address or not.

Encoding:

bit nr	7	6	5	4	3	2	1	0
name:	r	r	r	r	r	aet	d	s

- **Sending (s)**
  - position: bit 0
  - description: In case the contained (Extended) GA is added, then this field *Sending* (s) shall indicate whether it shall be the sending GA for the referred GO or not. The flag s shall only be interpreted in case the flag d equals 0; in case the flag d equals 1, the value of the flag s shall be don't care.
  - format, encoding:
    - 0: not sending: The contained GA shall not be the sending GA for the referred GO.
    - 1: sending: The contained GA shall be the sending GA for the referred GO.

  Error and exception handling

  - If a link is Added using PID_OBJECTLINK to a GO to which the contained GA is already linked, bit different value of the field S, then this new value of the flag s shall be used.
  - A GA set as "sending" shall take precedence on other previous sending GA for that GO.
- **Add/Delete (d)**
  - position: bit 1
  - description: The field *Add/Delete* (d) shall indicate whether the contained (Extended) GA shall be added to or removed from the list of GA assigned to the referred GO.
  - format, encoding:
    - 0: add: Add the contained GA to the list of GAs assigned to the referred GO.
    - 1: delete: Remove the contained GA from the list of GAs assigned to the referred GO.
- **Address extension type (aet)**
  - position: bits 2: Address Extension Type
  - description: DOA or SN.

format, encoding: 0 : SN
1: DoA

- **reserved (r)**
  position: bits 7 to 3
  description: These bits are reserved.
  format, encoding: The E-Mode Management Client shall clear (0) these bits. The E-Mode Management Server (device) shall ignore the value of these bits.

2. KNX Serial Number (SN) or RF Domain Address (DoA)
Description: This shall be the KNX Serial Number or 000000000000 for DoA part of the Extended GA. If this Function request does not contain an Extended GA but only a basic GA, then this field shall be 000000000000h.
Encoding: DPT_SerNum (DPT_ID = 221.001) or 000000000000

3. Group Address (GA)
Description: This field shall be the basic GA or the GA part of the Extended GA, to which the Link shall be established or from which the Link shall be broken.
Encoding: U_{16} value

4. Handle
Description: This shall be a local handle that shall identify the GO.
This shall be the GO number of the GO within the E-Mode Channel.
Encoding: U_{16} value

3.2.2.2 Function Read Object LinkObject

5. Flags
Description: This field shall indicate whether a the contained GA is the sending GA for the referred GO or not.
Encoding:

bit nr	7	6	5	4	3	2	1	0
name:	r	r	r	r	r	aet	r	s

- **Sending (s)**
  position: bit 0
  description: This field Sending(s) shall indicate whether the contained (extended) GA is the sending GA for the referred GO pr not. It shall also indication the extended address type.

  format, encoding: 0: not sending: The contained GA shall not be the sending GA for the referred GO.
  1: sending: The contained GA shall be the sending GA for the referred GO.

Address extension type (aet)
position: bits 2: Address Extension Type
description: DOA or SN.
format, encoding: 0 : SN
1: DoA

- **reserved (r)**
  position: bits 7 to 3
  description: These bits are reserved.
  format, encoding: The E-Mode Management Client shall clear (0) these bits. The E-Mode Management Server (device) shall ignore the value of these bits.

3.3 Configuration interworking

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

The Configuration of KNX RF S-Mode devices does not conflict with the Management – or Configuration Procedures of other S-Mode Profiles.

In order to separate communication from neighbouring installation or – Subnetworks, the Domain Address is introduced.

For the assignment of the DoAs, forwarding of system broadcast messages is controlled in the KNX TP1/RF Media Couplers through the Property “RF System Broadcast routing enable” (PID_RF_ENABLE_SBC; see [23]).

Additionally, the Programming Mode shall time-out automatically and in case of multiple devices with enabled Programming Mode, the ETS user shall select the device with which to continue (see 3.6.1).

3.4 Run-time Interworking

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

Runtime Interworking is not affected by this specification.

The KNX RF S-Mode devices use the same group communication and DPTs as existing TP1 S-Mode devices.

The Telegrams are routed transparently for the Interworking by the standard KNX TP1/RF Media Coupler, without the need for mapping or translation.

3.5 Registration and certification

Marking is done by combination of existing logos: RF1.R (for RF ready) and S-Mode

3.6 Integration and common tool impact

3.6.1 Programming Mode

Any KNX RF S-Mode device will automatically disable its Programming Mode after 4 minutes. This period cannot be retriggered.

If ETS scans for devices in which Programming Mode is active, for the assignment of DoA and/or IA, then ETS shall if more than one response is received show the KNX Serial Numbers contained in the responses and allow the ETS user to select the device with which to continue.

This shall allow that, if during these 4 minutes, Programming Mode is active in more than one device, typically in a neighbouring installation, the ETS user can still continue the DoA and/or IA assignment, without having to wait until the conflicting device(s) exits Programming Mode automatically.

3.6.2 Domain Address management

3.6.2.1 General

This clause contains two parts.

1. New DoA: DoA creation and assignment: see clause 3.6.2.2
2. Accessing an unkown installation with an unkown DoA: see clause 3.6.2.3

3.6.2.2 New DoA: DoA creation and assignment

3.6.2.2.1 DoA creation

Under no condition, ETS shall assign the numerical value 0000:00000000h to any KNX RF S-Mode device.

To “create” a new DoA, ETS shall let user choose between the following options (really a radio button list), in the ETS UI.

1. Use an existing DoA (that can be entered by the ETS user. E.g. when an Interface or Media Coupler is replaced by a new device)
2. Create a new DoA (with KNXA ManufID as upper 2 octets in the KNX Serial Number) + ETS random created part + additional check if the DoA is free

For the creation of a new DoA, ETS shall use a DoA in which the Manufacturer Code part is "KNX Association" (00FAh). That leaves 2^32 possible different DoAs (more than 4 000 000 000), which would be sufficient for some random chosen value. This effectively gives the ETS user the "look and feel" of a TP1 installation and prevent from RF specific user actions before being able to assign addresses.

3.6.2.2.2 Check if the created DoA is free

In addition to the DoA creation by ETS, ETS can check if the DoA address is free by using the service A_DomainAddressSelective_Read (see [24]).

3.6.2.2.3 DoA assigment

3.6.2.2.3.1 Media Coupler

This procedure sets the RF DoA and the Media Coupler’s Individual Address; the media Coupler is identified by the user by activating the Programming Mode in the device.

```
/* The Domain Address “Device.DoA” is assigned to the device. In the same go, the Individual */
/* Address Device.IAnew is assigned to the device. */
NM_DomainAndIndividualAddress_Write(NmpDoANew = Device.DoA, NmpIANew = Device.IAnew)
```

3.6.2.2.4 Extension of an existing RF installation with TP1

If an installation is initially configured without Media Coupler and later on extended with TP1, then it would be far more easy to make the Media Coupler use the DoA already used in the installation, than to use the Media Coupler's DoA and reprogram all installed RF devices.

3.6.2.2.4.1 Interface

The DoA shall be set in the local interface by writing the Property PID_RF_DOMAIN_-ADDRESS in the RF Medium Object of the interface.

3.6.2.2.4.2 RF end device

DoA assignment is only possible if the Media Coupler or the RF interface has already a DoA.

For setting the DoA of RF end devices, ETS has to know the medium to which it is connected and use normal broadcast when on TP1 for those services where system broadcast could be used on RF. If using a Media Coupler, ETS has to enable the PID_RF_ENABLE_SBC before starting the procedure with the end device.

DoA setting can be associated with the IA assignment procedure. This has not to be repeated for application parameter download.

This procedure sets the RF DoA and the device's Individual Address; the device is identified by the user by activating the Programming Mode in the device.

```
/* The Domain Address "Device.DoA" is assigned to the device. In the same go, the Individual */
/* Address Device.IAnew is assigned to the device. */
NM_DomainAndIndividualAddress_Write(NmpDoANew = Device.DoA, NmpIANew = Device.IAnew)
```

If using a Media Coupler, ETS has to disable the PID_RF_ENABLE_SBC after the procedure with the end device.

3.6.2.3 Accessing an unkown installation with an unkown DoA

3.6.2.3.1 Description

If ETS needs to access an already configured and running installation and does not know the DoA(s) that is (are) used in the installation, then it shall retrieve the DoA through one of the below possibilities.

3.6.2.3.2 Using the DoA of the interface

This shall be applicable if the interface is a fixed part of the installation. (RS232 -, USB – or KNXnet/IP Tunnelling Interface).

ETS shall read the DoA from the interface and display the DoA in the dialog of the Connection Manager.

3.6.2.3.3 Read the DoA from an installed device

ETS shall read the RF DoA via the diagnostic function "Read Individual Address".

The device shall be identified by the user by activating the Programming Mode in the device.

```
/* The Domain Address "Device.DoA" is read out from the device.*/
NM_DomainAndIndividualAddress_Read()
```

3.6.3 Interfaces

3.6.3.1 Domain Address

There are two types of KNX RF interfaces.

1. KNX RF USB Interfaces that remain permanently in the installation.
2. KNX RF USB "dongles" that are only used to access more than one installation and do not remain permanently in the installation.

The procedure will be the same on the two types of RF interfaces.

The difference will be the retention of the DoA by the interface. As the USB RF dongle will be used for different installations, it may not keep any RF DoA in permanent memory and need to be initialized each time from ETS.

Recommendation from TFRF is not to use too long RF telegrams. Probability to be collided is bigger than for small telegram. In case of disturbed RF, long telegram will frequently be corrupted. (telegram size : 100 ms max => preamble 4,8 ms overhead = 19 octets (1octet = 0,5 ms) max APDU length = 85 ms = 190 octets). Recommended frame length : 64

For download procedure, ETS should use a adaptive algorithm to start with the longest frame (after checking the longest frame it can use) and the according to numbers of errors, can reduce the frame length).

3.6.3.2 Local and – Remote Configuration Procedures

Download of a KNX RF S-Mode device locally via USB or IP can be meaningful

- for semi-directional devices, or
- if a lot of information need to be downloaded to the device.

Using USB interface is also a solution for battery powered devices. For this additional support the cEMI Local Transport Layer has to be used. ETS firstly has to connected to the interface, discover it and select it as a local interface. ETS then downloads locally the parameters. The IA of the device must the same as the local IA of ETS.

3.6.4 Domain Address based KNX RF Retransmitter

Each KNX RF S-Mode end device can be used as RF Retransmitter through the Property *RF Retransmitter Flag* (PID_RF_RETRANSMITTER; see 2.3.2.4).

Each KNX RF S-Mode device can be used as RF Retransmitter.

The existing KNX Serial Number based RF Retransmitter can be used in installations without restriction, but it will not be visible as device in ETS project. There is no requirement for ETS to support this RF Retransmitter type.

3.6.5 Configuration Procedures for mask 27B0h

3.6.5.1 Introduction and general requirements

3.6.5.1.1 Management Procedures

The Configuration Procedures of the KNX RF S-Mode device Profiles base on the same Management Procedures as KNX TP1 devices.

3.6.5.1.2 Semi-directional devices and Partial Download

The bidirectional mode of the semi-directional devices is specified so that the Configuration Procedures do not have to take explicit care. There is no need for the MaC to explicitly keep alive the bidirectional mode. The MaC may also restart the MaS without losing the communication to the MaS.

One exception is Partial Download. That Configuration Procedure does not use the Programming Mode, so the MaC user has to activate the bidirectional mode locally on the device.

3.6.6 Common requirements for all KNX RF end device Profiles and – interfaces

3.6.6.1 Device Descriptor Type 0 – RF medium indication

The medium type 2h indication in the Device Descriptor Type 0 shall indicate the KNX RF Communication Medium. This shall be indicated like that in ETS.

3.6.6.2 Group Object Config flags

ETS can use the standard Group Object flags known from KNX TP1 devices.

3.6.7 ETS Database

The KNX RF S-Mode devices will need an ETS database that will be certified and supported by ETS4.5 Manufacturer Tool. There is no need to inherit E-Mode functionality or newly introduced FEC features.

This is limited to ETS4.5 new versions and will not be supported for ETS3.

3.6.8 Integration of RF Media Coupler

The Main Line can be RF. In this, also the Line shall be KNX RF and shall inherit the DoA from the Main Line and no “Line Coupler’s” (RF/RF) are possible.

The Backbone of “RF-only” projects will be TP1 or IP (=default).

3.6.9 Integration of RF interface

ETS shall check PID_MEDIUM_TYPE (PID = 51) of cEMI Server Object.

ETS may also use the cEMI additional information of RF medium type to send information.

If no additional info field then the cEMI server will use is local values for DoA and KNX Serial number. ETS will by default use DoA = 00000000000h for all RF telegram to use local cEMI DoA.

In the future, if the DOA is part of project database, ETS may use the RF-‘SN’ and other RF fields of the additional information to send frame to RF. These additional infos (currently used by EITT) could also be displayed with RF infos on BusMonitor.

Diagnostic data that ETS could offer to the installer.

- Signal strength of received frame,

Future extensions:

- (Support RF multi as optional)
- RF multi channel usage
- (RF interface filtering on DoA via a property AN : cEMI extension for RF)

3.6.10 Integration of RF devices

RF device : integration of new mask version but fully compliant with TP devices.

For non permanent bidirectional devices, the device may not be reachable and ETS needs to warn the user. For this a information “not always present on the bus” must included in the product data. Indication could be the ETS indicates that AP must be downloaded immediately after addressing the device.

If ETS knows the procedure will take some time, it can use the PID_BIDIR_TIMEOUT to retrigger the bidirectional mode.

3.6.11 End device or TP1/RF Media Coupler with Retransmitter function

RF S-Mode End Device may also integrate a Retransmitter function.

To this, the Profile Module “Domain Address based RF Retransmitter” is defined: see 2.7.3.

If the function is embedded (declared in the database with the ETS MT) then it can be activated and deactivated on the ETS UI. This parameter is managed by *PID_RF_RETRANSMITTER* (see 2.3.2.5) in the *RF Medium Object*.

Retransmission shall be restricted to Telegrams in the RF Domain of the device, as specified in 2.2.1.3.

3.6.12 Diagnostics

- Scan Line

  Define a new DomainAddressSelectiveRead to get DoA, IA and SN in the response.

- Check IA

  Use existing procedure or Scan Line with dedicated IA.

- Scan device in prog mode

  using IARead, DoARead, SNRead (as in program IA procedure).

3.7 Risks and compatibility issues

Unidirectional devices can't fit to this new specification, because no RF DoA can not be assigned.

For existing RF devices, if the history list (filtering with LinkFrame Number) is based only on the serial number and not the source address, they may reject the Multicast telegram coming from extended addresses based on DoA. As they can't be configured today for such links it is a problem only if the tool wants to access them with this DoA.