Application Domain Specific Standards

10

System Conformity Testing

2

Extended Frame Format (EFF) & Logical Tag Extended (LTE)

1

Summary:

These specifications are a part of the KNX System Conformity Test Specifications for Application Domain specific standards.

This Test Specification contains the Test Specifications for Extended Frame Format and Logical Tag Extended (LTE) Mode.

Version 01.01.01 is a KNX Approved Standard.

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

Document Updates

Version	Date	Modifications
0.1	06-Aug-2002	S. Wichert: initial version
0.2	22-Aug-2002	S. Wichert: update
0.3	04-Apr-2003	S. Wichert: update for review in KonCert
0.4	30-Apr-2003	S. Wichert: update after 1st presentation in KonCert
0.5	12-Jun-2003	S. Wiupdate after 1st use during tests for a system/stack certification
0.6	18-Sep-2003	S. Wichert: update after use in a 2nd test session + comments from KonCert
0.7	24-Oct-2003	Preparation update of document for release for voting
0.8	May 2004	Preparation update of document for final voting – no comments received during RfV KTB ending February 2004
AS 1.0	September 2004	Preparation update of document to AS – integrated comments received during FV KTB ending August 2004
AS 1.1	September 2004	Readying document for publication in V2.0 of the KNX specifications
01.01.01	2013.10.23	Editorial updates for the publication of KNX Specifications 2.1.

Filename:	10_02_01 System Conformity Tests - EFF and LTE Tests v01.01.01 AS.docx
Version:	01.01.01
Status:	Approved Standard
Savedate:	2013.10.28
Number of pages:	40

Contents

1 Introduction

1.1 Scope

This document contains the test specifications for Extended Frame Format (EFF) and especially LT Extended Mode & Frame Formats.

In an initial phase, the (media dependent layer) tests are described only for TP1 media, because first implemented products using the extended frame format (LTE) will be based on the TP1 media.
If extended frame format/LTE would be implemented also on other media, the applicant of such implementation shall ask the KNX certification body (contact KNX certification department) for definition/adaptation of supplementary test specifications (media dependent parts).

1.2 Objectives

This document includes information necessary for system conformity testing of 'LTE Mode devices'. What features an 'LTE device' actually shall implement to bear the KNX logo and which tests shall be done for verification are laid down in Vol. 6 "Profiles" of KNX System Standards.

This document includes descriptions of the tests for 'extended frame format' (communication layer testing).

The focus of this document is test specifications for:

- Standard messages using long frame format (EFF = 0h; standard messages with APDU > 15 octets)
- LTE frame formats (EFF = 4h … 7h, used for messages with APDU-Length from 1 to more than 15 octets)
- LTE Configuration Mode (→ LTE Addressing/Logical Tag Extended Linking)

In addition, this document contains guidelines for functional block testing of applications using the LTE application model.

As far as applicable, the test specifications in this document are based on the device model with implementation independent resources according supplement 5 of KNX system specification.

1.3 Referenced documents

References, relevant for this document:

[1] Supplement 13	KNX System Specification: Extended Frame Format
[2] Vol. 10-1	KNX System Specification: LTE Specification
[3] Supplement 5	KNX System Specification: Implementation Independent Resources
[4] Vol. 8-2-2	KNX System Specification: TP1 Physical and Link Layer Tests
[5] Vol. 8-3-3	KNX System Specification: Network Layer Tests
[6] Vol. 8-3-4	KNX System Specification: Transport Layer Tests
[7] Vol. 8-3-7	KNX System Specification: Application (Interface) Layer Testing – Network Management Server Testing
[8] Vol. 8-7	KNX System Specification: Interworking Tests
[9] Supplement 13	KNX System Specification: Supplement 13 Extended Frame Format – Chapter “Data Link Layer General – Extended”
[10] Supplement 15	KNX System Specification: Easy Common Parts
[11] Vol. 6	KNX System Specification: Profiles
[12] AN 30	KNX System Specification: Application Note “SNA Read from Router”
[13] TS GO & NM	KNX System Specification: Test Supplement “Group Object & management Testing for Devices with Implementation Independent Resources”
[14]	Test Suite Supplement: Testing for Routers
[15] AN 42	L2-ACK for LTE messages

1.4 Abbreviations

APDU	Application Layer PDU
APCI	Application Layer Protocol Control Information
BAU	Bus Access Unit
BCU	Bus Coupling Unit
BDUT	Bus Device under Test
DAA	Distributed Address Assignment
DO	Device Object
EFF	Extended Frame Format
EITT	Interworking Test Tool; (formerly “EIB Interworking Test Tool”)
GA	Group Address
GO	Group Object
HMI	Human Machine Interface
IA	Individual Address
IO	Interface Object
IOT	Interface Object Type
LTE	Logical Tag Extended
PDU	Protocol Data Unit
PEI	Physical External Interface
SNA	Subnet Address
TPCI	Transport Layer Protocol Control Information
USB	Universal Serial Bus

2 Test Set-Ups

Please refer to the corresponding test specifications in Vol. 8 of KNX System Specification for test set-ups to be used with the tests described in this document.

3 Communication Layer Tests

3.1 Physical Layer

No (additional) Tests

3.2 Link Layer – Media Dependent Part: TP1

3.2.1 Physical Part of Link Layer

No (additional) tests

3.2.2 Data Part of Link Layer [1]

To be integrated into KNX Handbook Series, Volume 8, Part 2, Chapter 2, clause 9.2

3.2.2.1 Control Field – Receive

The test in chapter 8-2-2, clause 9.2.1.1 applies.

Note: Ctrl-Field combinations in L-Date-Frames with Frame Type Flag cleared are allowed for extended frames.

3.2.2.2 Control Field – Priorities: Send

Purpose: Check if the BDUT can send telegrams with all priorities.
In addition to the test in chapter 8-2-2, clause 9.2.1.2, a Value Read Test shall be done with LTE-HEE Datapoints ("LTE group objects").

Procedure: Stimulate BDUT to send telegrams with **all** priorities: send Value Read to implemented server Datapoint, using A_GroupPropValue_Read or A_PropertyValue_Read Service (whatever is applicable).

Stimuli (Example with A_PropertyValue_Read):

IN *B**C** AFFE 1101 65 03 D5 09 33 10 01*

IN *B**4** AFFE 1101 …..*

IN *B**8** AFFE 1101 …..*

IN *B**0** AFFE 1101 …..*

Acceptance: BDUT sends answers with priorities corresponding to those set in the Value Read frames:

*OUT3**C** 60 1101 AFFE 11 03 D6 09 33 10 01 …(property value)*

OUT *3**4** 60 1101 AFFE 11 03 D6 09 33 10 01 …*

OUT *3**8** 60 1101 AFFE 11 03 D6 09 33 10 01 …*

OUT *3**0** 60 1101 AFFE 11 03 D6 09 33 10 01 …*

3.2.2.3 Control Field – Repetition Flag: Send

No (additional) Tests

3.2.2.4 Control Field – Repetition Flag: Receive

No (additional) Tests

[1] Please refer also to chapter 8-2-2, clause 9.2

Note: The verification whether the BDUT only writes the value when receiving the first frame and not in case of the second (repeated) frame can only be checked in case of devices in view of availability of a Physical (External) Message Interface, but not in case of “closed devices”.

3.2.2.5 Control Field – Frame Type Flag: Send

- Frame Type Flag Test (Send) is done implicitly with the destination address tests using both formats (L_Data **Extended** and L-Data **Standard** Frames), i.e. test according to clause 3.2.2.13to 3.2.2.19in this document for L_Data Extended, and the corresponding test described in chapter 8-2-2 for L_Data Standard frames.
  Consider in these tests: The extended frame format shall not be used instead of standard frame format if encoding capabilities of L_Data-Standard frame are sufficient (e.g. for short frames). [2]
- No other additional Tests

3.2.2.6 Control Field – Frame Type Flag: Receive

- Frame Type Flag Test (Receive) is done implicitly with the destination address tests using both formats (L_Data **Extended** and L-Data **Standard** Frames), i.e. tests according to clauses 3.2.2.13, 3.2.2.15, 3.2.2.17 for L_Data Extended, in this document and the corresponding tests described in chapter 8-2-2 for L_Data Standard frames.
- Additional Test: Frame Type Flexibility, see clause 3.2.2.7

3.2.2.7 Frame Type Flexibility (Receive)

Purpose: Check BDUT’s Frame Type Flexibility in received frames.
The decision whether to use standard (short) or extended (short/long) frame format is made in the application layer and selected by the frame format parameter in T_Data_.... services. A device supporting L_Data extended frame format shall be tolerant towards usage of long frames if short frames would be sufficient. [3]

Procedure: Use a telegram generator (e.g. EITT) to send extended frame types to the BDUT, even if standard (short) frame types are sufficient; send frames with all relevant AL-Services (as far as possible/implemented):

a) A_GroupValue_Write

b) A_GroupValue_Read

c) A_PropertyValue_Write

d) A_PropertyValue_Read

e) A_IndividualAddress_Read (broadcast addressing)

Stimuli (examples):
a) IN 3C E0 AFFE 2000 04 00 80 0B 21 16: GroupValueWrite
b) IN 3C E0 AFFE 2000 01 00 00: GroupValueRead
c) IN 3C 60 AFFE 1101 06 03 D7 00 36 10 01 01: PropertyValueWrite
Note: sets Programming Mode of BDUT to ON, for step e)
d) IN 3C 60 AFFE 1101 05 03 D5 00 01 10 01: PropertyValueRead
e) IN 3C E0 AFFE 0000 01 01 00: IndividualAddressRead

[2] Please refer to supplement 13, remark in clause 2.1.3

[3] Please refer to supplement 13, remark in clause 2.1.3

Acceptance: BDUT accepts extended frames instead of standard short frame:
- case a): BDUT sends IACK
- case b): BDUT sends the response with A_GroupValue_Response using standard short or extended long frame according the length of APDU.
- cases c) & d): BDUT sends the response with A_PropertyValue_Response using standard short or extended long frame, according the length of the expected APDU.
- case e) BDUT sends the response with A_IndividualAddress_Response using a standard short frame.

Expected reactions to the stimuli examples above:

a)	*CC: IACK*
b) OUT	*BC 1101 2000 E4 00 40 00 00 00: GroupValueResponse*
c) OUT	*BC 1101 AFFE 66 03 D6 00 36 10 01 01: PropertyValueResponse*
d) OUT	*BC 1101 AFFE 67 03 D6 00 01 10 01 00 00: PropertyValueResponse*
e) OUT	*BC 1101 0000 E1 01 40: IndividualAddressResponse*

Note: In cases b) to d), the BDUT sends the answer in the appropriate frame format (quoted from supplement 13, remark in clause 2.1.3):
The extended frame format shall not be used instead of standard frame format if encoding capabilities of L_Data-Standard frame are sufficient (e.g. for short frames).
Read requests (A_GroupValue_Read, A_PropertyValue_Read) always fit into standard short frame format. Nevertheless, if they are received in the extended (long) frame format, the remote Application Layer (BDUT) shall accept them. But the BDUT’s Response-PDU shall use the appropriate short or long frame format.

3.2.2.8 Extended Control Field – Address Type Flag

- Address Type Flag Test is done implicitly with the tests for destination address (clauses 3.2.2.13, 3.2.2.15, with L-Data **Extended** Frames).
- No other additional Tests

3.2.2.9 Extended Control Field – Hop Count

The Hop Count field (Routing Counter) is part of network layer information, for tests see 3.5.

3.2.2.10 Extended Control Field – Extended Frame Format (EFF): LTE address Type

- The Tests described in clauses 3.2.2.17, 0 and 0 shall be done with each (implemented) EFF-LTE Frame Type
- No other additional Tests

3.2.2.11 Extended Control Field – EFF: Individual and Group Addressed L-Data Extended Frames

- The “Destination Address Tests” described in chapter 8-2-2, clause 9.2.3 shall all be done in addition with L-Data Extended Frames (if applicable, testability is depending on implemented application features)
- These additional Tests are listed below: clauses 3.2.2.13, 3.2.2.14, 3.2.2.15, 3.2.2.16, 0, 0.

3.2.2.12 Source Address

Purpose: Check the correct setting of the source address in a frame sent by the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.2, the test shall also be done with an Extended Frame; e.g. by sending an LTE-HEE Datapoint.
This test is to check the SA´s right position, compared to the position in a Standard Frame.

Procedure: Stimulate BDUT to send a frame, e.g. by reading a property with data > 10 octets

Stimulus (example):
IN *BC AFFE 1101 65 03 D5 09 33 10 01*

Acceptance: The source address in the response matches with the BDUT's individual address.

Expected reaction to the stimulus example above:
OUT *3C 60* ***1101*** *AFFE 11 03 D6 09 33 10 01 …(data with > 10 octets)*

3.2.2.13 Destination Address – Individual Address

Purpose: Check the acceptance of individually addressed telegrams by the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.3.1, the test shall also be done with an individual addressed L-Data **Extended** Frame (EFF = 0h, APDU-length > 15 octets).

Procedure: Use a telegram generator (e.g. EITT) to send an individual addressed extended frame to the BDUT.

Stimulus (example):
IN *3C 60 AFFE* ***1101*** *11 03 D7 08 33 20 04 … (2 elements of GO table)*

Acceptance: BDUT accepts the frame (sends IACK).

3.2.2.14 Destination Address – Unused Individual Address

Purpose: Check the rejection of telegrams not addressed to the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.3.2, the test shall also be done with an individual addressed L-Data **Extended** Frame (EFF = 0h, APDU-length > 15 octets).

Procedure: Use a telegram generator (e.g. EITT) to send an individual addressed extended frame to the BDUT.

Stimulus (example):
IN *3C 60 AFFE* ***1234*** *11 03 D7 08 33 20 04 … (2 elements of GO table)*

Acceptance: BDUT does not accept the frame (sends no IACK).

3.2.2.15 Destination Address – Group Address

Purpose: Check the acceptance of group addressed telegrams by the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.3.3, the test shall also be done with a group addressed L-Data **Extended** Frame (EFF = 0h, APDU-length > 15 octets).

Procedure: Use a telegram generator (e.g. EITT) to send a group-addressed frame to the BDUT. Use a group address that is used by the BDUT's application.

Stimulus (example):
IN *3C E0 AFFE* ***1000*** *10 00 80 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E*

Acceptance: BDUT accepts the frame (sends IACK).

3.2.2.16 Destination Address – Unused Group Address

Purpose: Check the rejection of telegrams not addressed to the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.3.2, the test shall also be done with a group addressed L-Data **Extended** Frame (EFF = 0h, APDU-length > 15 octets).

Procedure: Use a telegram generator (e.g. EITT) to send group addressed extended frame onto the bus. Use a group address that is NOT used by the BDUT's application.

Stimulus (example):
IN 3C E0 AFFE ***2222*** *10 00 80 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E*

Acceptance: BDUT does not accept the frame (sends no IACK).

3.2.2.17 Destination Address – LTE Address

Purpose: Check the acceptance of LTE-addressed (multicast or broadcast) telegrams and check if IACK is sent by the BDUT. In addition to the test in chapter 8-2-2, clause 9.2.3.3, the test shall also be done with an LTE-addressed L-Data **Extended** Frame.

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame to the BDUT:

Stimulus (example: A_GroupPropValue_Read with LTE Broadcast Address):
*IN 3C E****7*** *AFFE* ***0000*** *08 07 E8 C3 B4 00 FF 00 FD 33*

Acceptance: BDUT accepts the frames (sends IACK and, in case of a Read Request, sends the response)

Expected reaction to the stimulus example above (IACK not listed):
*OUT 34 E****7*** *1101* ***0000*** *14 07 E9 C3 B4 01 FF 00 FD 33 … (property value)*

3.2.2.18 Destination Address – Unused LTE Address

Purpose: Check the rejection of telegrams not addressed to the BDUT.
In addition to the test in chapter 8-2-2, clause 9.2.3.4, the test shall also be done with an LTE-addressed L-Data **Extended** Frame.

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame to the BDUT.

Ensure that the used LTE Address (in stimuli frame) is not contained in the BDUT as the address of any receiving LTE-HEE Datapoint.

Stimulus (example: A_GroupPropValue_Read with an unassigned tag):
*IN 3C E****7*** *AFFE* ***0001*** *08 07 E8 C3 B4 00 FF 00 FD 33*

Acceptance: BDUT does not accept the frames.
Note: The BDUT may answer with an IACK. LTE-devices, particularly LTE-devices with limited memory resources, may take the decision, if a received LTE frame is addressed to it, later up in the stack (on Datapoint level).
For such devices, acceptance check must be done by other means then checking the IACK e.g. by reading to an LTE-HEE server Datapoint (A_GroupPropValue_Read); then the device must not answer (no A_GroupPropValue_Response) to the LTE read request.

Expected reaction to the stimulus example above:
- if no IACK is sent, then accepted.
- if an IACK is sent, then no response must be sent.

3.2.2.19 Destination Address – Send Telegram

Purpose: Check if BDUT can send a group-, an LTE-, a broadcast and an individual addressed telegram. In addition to the test in chapter 8-2-2, clause 9.2.3.5, the tests for group-, broadcast- and individual addressed telegrams shall be done with L-Data **Extended** Frames (APDU > 15 octets). [4]

Note: Tests for LTE-addressed frames are not shown here, as they are covered implicitly in several other test cases described in this document, e.g. in the tests under clause 4.1.3.2.1.

Procedure: Stimulate BDUT to send a group telegram, a broadcast telegram and an individually addressed telegram. Check telegrams with a busmonitor tool, e.g. EITT.

Stimulus (example for individual addressing): stimulus in 3.2.2.13
*IN 3C 6**0** AFFE **1101** 11 03 D7 08 33 20 04 … (data: 2 elements of GO table)*

Acceptance: BDUT sends all telegrams correctly.

Expected reaction to the stimulus example above (response to stimulus in 3.2.2.13):
*OUT 3C 6**0** 1101 **AFFE** 11 03 D6 08 33 20 04…(data: 2 elements of GO table)*

3.2.2.20 BDUT is a router

Test substituted by Link Layer Tests in Test Suite Supplement "Testing for Routers"

Remark: clause 9.2.3.6 in chapter 8-2-2 shall be deleted.

3.2.2.21 Information Length – Send

Purpose: Check that BDUT can send frames with different (minimum and maximum supported) info lengths. In addition to the test in chapter 8-2-2, clause 9.2.4.1, the test shall be done also with L-Data **Extended** Frames.
Note: test case b) in chapter 8-2-2, clause 9.2.4.1, is not applicable for devices based on implementation independent resources.

Procedure: Stimulate BDUT to send a

a) frame with minimum information length (e.g. a T-Disconnect)
b) frame with maximum information length (if applicable)
c) frame with APDU-length 15 octets
d) frame with APDU-length 16 octets
e) frame with APDU-length > 16octets: Check response to stimulus *in 3.2.2.13*

Stimuli (examples):

a) IN	*BC AFFE 1101 61 43 00: DeviceDescriptorRead; connection-oriented*
b) IN	*no applicable example yet*
c) IN	*BC AFFE 1101 65 03 D5 00 0F 10 01: PropertyValueRead(OrderID)*
d) IN	*no applicable example yet*

Acceptance: BDUT creates frames with information length set correctly.

[4] Tests for group- and broadcast-addressed frames are not (yet) applicable, as there is not one application yet over the whole KNX System. (Sep-2003)

Expected reactions to the stimuli examples above:

*a) OUT B0 1101 AFFE 6**0** 81: T-Disconnect*

b) OUT

*c) OUT BC 1101 AFFE 6**F** 03 D6 00 0F 10 01 … (data = Order ID of BDUT)*

d) OUT

*e) OUT 3C 60 1101 AFFE **11** 03 D6 08 33 20 04…(2 elements of GO table)*

3.2.2.22 Information Length – Receive

Purpose: Check that BDUT receives frames with different (minimum and maximum supported) info lengths. In addition to the test in chapter 8-2-2, clause 9.2.4.2, the test shall also be done with L-Data **Extended** Frames.

Procedure: Check Write and Response in test according clause 3.2.2.13.

Stimulus (example for individual addressing): stimulus in 3.2.2.13

*IN 3C 60 AFFE 1101 **11** 03 D7 08 33 20 04 … (data: 2 elements of GO table)*

Acceptance: BDUT accepts the frame(s).

Expected reaction to the stimulus example above (response to stimulus in 3.2.2.13):

OUT 3C 60 1101 AFFE 11 03 D6 08 33 20 04…(data: 2 elements of GO table)

Note: IACK must not be used as check criterion, because frame length information is not taken into account for IACK generation. Check shall be done by other means, e.g. writing and reading back of properties.

3.2.2.23 Incorrect Information Length – Receive

Purpose: Check that BDUT does not accept frames with incorrectly set info length. In addition to the test in chapter 8-2-2, clause 9.2.4.3, the test shall also be done with L-Data **Extended** Frames.

Procedure: Send frames from EITT to BDUT with information length not corresponding to number of sent data:

Stimuli (examples: write to a property, try to set programming mode):

*IN 3C 60 AFFE 1101 **07** 03 D7 00 36 10 01 01; length information is too long*

*IN 3C 60 AFFE 1101 **05** 03 D7 00 36 10 01 01; length information is too short*

Note: the frame type flexibility feature is used for this test.

Acceptance: BDUT does not accept the frame

Expected reaction to the stimulus examples above: no responses generated

Note: IACK must not be used as check criterion, because frame length information is not taken into account for IACK generation. BCU and TP-Uart based devices always generate an IACK independently of information length.
Check shall be done by other means, e.g. writing and reading back of properties.

3.2.2.24 Check Octet – Checksum: Send

Purpose: Check if BDUT sets the checksum byte correctly when sending a frame.
In addition to the test in chapter 8-2-2, clause 9.2.5.1, the test shall also be done with L-Data **Extended** Frames to check if the Checksum is set at the **right frame position** in an L-Data extended frame.

Note: This test is not necessary for devices using the certified TP-Uart.

Procedure: Stimulate BDUT to send a frame. Use EITT with specific data interface device[5] or a logic analyser to check the correct calculation of the checksum.

Acceptance: BDUT sets checksum correctly.

3.2.2.25 Check Octet – (Incorrect) Checksum: Receive

Purpose: Check if BDUT rejects a frame with incorrect checksum
In addition to the test in chapter 8-2-2, clause 9.2.5.2, the test shall also be done with L-Data **Extended** Frames to check an incorrect checksum received at **right frame position**.

Note: This test is not necessary for devices using the certified TP-Uart.

Procedure: Use waveform or telegram generator to send a frame addressed to the BDUT with incorrect checksum.

Acceptance: BDUT sends INACK.

3.2.2.26 Confirmation Field – Sending of Ack, Nack, Busy

Purpose: Check if BDUT is always able to respond to frames addressed to it by means of the appropriate acknowledge frame (ACK, NACK or BUSY). This shall be ensured for physically and group-/LTE-addressed frames.
In addition to the test in chapter 8-2-2, clause 9.2.6.1, the test shall also be done with L-Data **Extended** Frames (as stimuli).

Procedure: Send frames to BDUT and check correct generation of acknowledge frame. Check e.g. in PICS under which conditions a Busy signal is transmitted.

Acceptance: BDUT sends correct acknowledgement frames

3.2.2.27 Confirmation Field – Repetition

Purpose: Check if BDUT is able to repeat a frame. In addition to the test in chapter 8-2-2, clause 9.2.6.2, the test shall also be done with L-Data **Extended** Frames (as stimuli). This additional test shall be done with LTE-frames (EFF = 4 …7) and, if applicable, with standard long frames (EFF = 0 and APDU-length > 15 octets).

Note: For L-Data Extended frames with EFF = 4…7 (LTE-addressed frames with A_GroupPropValue AL-Services):

- For the A_GroupPropValue_InfoReport AL-Service, repetitions are not appropriate and therefore not allowed, see AN 42 ([15]).

- For A_GroupPropValue_Write/Read/Response AL-Services, repetitions are allowed, if foreseen in the BAU-device profile, see AN 42 ([15]). In case repetitions are enabled, correct repetition behaviour shall be tested according the following test procedure.

Procedure: Stimulate BDUT to send telegram (L_Data.req with repetition flag set), which triggers via the sent line of the BDUT the Waveform/Telegram Generator to send a

a) NACK

[5] Ask KNX system or certification department, where to get such a device.

b) BUSY

c) no frame

d) overlapping NACK and BUSY

Acceptance: In case of standard long frames (EFF = 0):

- BDUT repeats frame with the repetition flag set up to three times.

In case of LTE frames (EFF = 4 … 7):

- BDUT **never** repeats frames sent with A_GroupPropValue_InfoReport
- BDUT may repeat (if repeated: up to three times, with the repetition flag set) frames sent with A_GroupPropValue_Read/Response/Write services. The information, if frames are repeated or not for the mentioned services, shall be given by the manufacturer in the BAU profile.

3.2.2.28 Confirmation Field – Receive IACK

Purpose: Check if BDUT does not repeat a frame if it receives an IACK.
In addition to the test in chapter 8-2-2, clause 9.2.6.3, the test shall also be done with L-Data **Extended** Frames (as stimuli).

Procedure: Stimulate BDUT to send a frame to which another device responds with IACK.

Acceptance: BDUT does not repeat frame.

3.2.2.29 Polling Frames

- Polling Mechanism is not foreseen with Extended Frame Format, therefore
- no (additional) tests

3.2.2.30 Busmonitor Mode of the Link Layer

To be completed

Tests are applicable only on devices supporting busmonitor mode and providing the monitored frames on an external message interface, e.g. a data interface from bus to RS232 or USB.

3.3 Link Layer – Media Dependent Part: PL110

To be completed

3.4 Link Layer General - Media Independent Part

- Tested implicitly with media specific tests.
- No (additional) tests.

3.5 Network Layer

To be integrated into KNX Handbook Series, Volume 8, Part 3, chapter 3.

3.5.1 Test case 1: Black Box Test

3.5.1.1 Multicast communication – group- & LTE-addressed frames

Purpose: Check if BDUT sends with correct hop count.
In addition to the test in chapter 8-3-3, clause 3.1, the test shall be done with stimuli to Datapoints where the answer uses an L-Data **Extended** Frame. This is intended to check also the correct position of hop count within the extended (response) frame.

Procedure: Send telegrams with Routing Count 6, 5, 4, 3, 2, 1 and 0 to the BDUT.

a) Send A_GroupPropValue_Read telegrams to a BDUT's LTE server output Datapoint or to a read-enabled LTE server input Datapoint.
Note: LTE server output Datapoints are always read-enabled

b) If applicable: send A_GroupValue_Read telegrams to a BDUT's group object with Response-APDU-Length > 15 octets.
Note: the group object's read enable flag must be set.

	Stimuli (examples for case a), geographical Tag A=1, R=1, S=1):
IN	*34* ***E****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***D****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***C****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***B****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***A****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***9****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*
IN	*34* ***8****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*

Acceptance:BDUT sends the responses with Routing Count according Property 51 in its Device Object (RC = 6 or lower).

	Expected reaction to all stimulus examples above:
OUT	*34* ***E****4 1101 0411 0A 07 E9 01 80 01 FF 00 FD FE 00 00*

Procedure: Send telegrams with Routing Count 7 to the BDUT.

c) Send A_GroupPropValue_Read telegrams to a BDUT's LTE server output Datapoint or to a read-enabled LTE server input Datapoint.
Note: LTE server output Datapoints are always read-enabled

d) If applicable: send A_GroupValue_Read telegrams to a BDUT's group object with Response-APDU-Length > 15 octets.
Note: the group object's read enable flag must be set.

	Stimuli (examples for case c), geographical Tag A=1, R=1, S=1):
IN	*34* ***F****4 AFFE 0411 08 07 E8 01 80 00 FF 00 FD FE*

Acceptance: BDUT sends the responses with Routing Count 7 (the BDUT may optionally answer with routing counter 6).

	Expected reaction to the stimulus example above:
OUT	*34* ***E****4 1101 0411 0A 07 E9 01 80 01 FF 00 FD FE 00 00*

3.5.1.2 Unicast / Device oriented communication – connected

- No (additional) tests.

- Note: The test stimuli described in chapter 8-3-3, clause 3.2 (T-Connect), will be answered with a T-Disconnect by devices which do not support connection oriented communication

3.5.1.3 Unicast / Device oriented communication – connectionless

Purpose: Check if BDUT sends with correct hop count.
In addition to the test in chapter 8-3-3, clause 3.3, the test shall be done with stimuli to Datapoints where the answer uses an L-Data **Extended** Frame. This is intended to check also the correct position of hop count within the extended (response) frame.

Procedure: Send telegrams with Routing Count 6, 5, 4, 3, 2, 1 and 0 to the BDUT.

If applicable: send A_PropertyValue_Read telegrams to a BDUT's Datapoint with Response-APDU-Length > 15 octets.

Stimuli (examples, property data length > 10 octets):

IN *BC AFFE 1101* ***6****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***5****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***4****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***3****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***2****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***1****5 03 D5 09 33 10 01*
IN *BC AFFE 1101* ***0****5 03 D5 09 33 10 01*

Acceptance: BDUT sends the responses with Routing Count according Property 51 in its Device Object (RC = 6 or lower).

Expected reaction to all stimulus examples above:

OUT *3C* ***6****0 1101 AFFE 11 03 D6 09 33 10 01 … (property data, 12 octets)*

Procedure: Send telegrams with Routing Count 7 to the BDUT.

If applicable: send A_PropertyValue_Read telegrams to a BDUT's Datapoint with Response-APDU-Length > 15 octets.

Stimuli (examples, property data length > 10 octets):

IN *BC AFFE 1101* ***7****5 03 D5 09 33 10 01*

Acceptance: BDUT sends the responses with Routing Count 7 (the BDUT may optionally answer with routing counter 6). [6]

Expected reaction to the stimulus example above:

OUT *3C* ***6****0 1101 AFFE 11 03 D6 09 33 10 01 … (property data, 12 octets)*

[6] RC = 7 is preferred, RC = 6 is also allowed.

3.5.1.4 Broadcast communication

Purpose: Check if BDUT sends with correct hop count.
In addition to the test in chapter 8-3-3, clause 3.4, the test shall be done with stimuli that lead to an answer using an L-Data **Extended** Frame (if applicable). This is intended to check also the correct position of hop count within the extended (response) frame.

Note: This test covers the Standard Mode System Broadcast (EFF=0000b & destination GA=0000h) with APDU-length > 15 octets. The LTE-Broadcast is covered by the test according 3.5.1.1, since the LTE-Broadcast is an address within the LTE address range with EFF = 7.

Procedure: Send telegrams with Routing Count 6, 5, 4, 3, 2, 1 and 0 to the BDUT.

Note:No application example is available yet over the whole range of the current KNX System specification. The test description here can be seen as a placeholder for future system extensions.

Acceptance:BDUT sends the responses with Routing Count according Property 51 in its Device Object (RC = 6 or lower).

Procedure: Send telegrams with Routing Count 7 to the BDUT.

Note:No application example is available yet over the whole range of the current KNX System specification. The test description here can be seen as a placeholder for future system extensions.

Acceptance: BDUT sends the responses with Routing Count 7 (the BDUT may optionally answer with routing counter 6).

3.5.2 Test case 2: Bus/PEI Test

Clause 4 in chapter 8-3-3 shall be deleted

3.5.3 Test case 3: Testing of routing algorithm in routers

Tests are covered/substituted by Test Suite Supplement "Testing for Routers"

Remark: clause 5 in chapter 8-3-3 shall be deleted.

3.6 Transport Layer

To be integrated into KNX Handbook Series, Volume 8, Part 3, chapter 4.

The following tests shall be done using a sample application on the BDUT, or comparable means. The used sample application shall be documented in the test report.

3.6.1 Multicast communication

3.6.1.1 LTE-addressed telegrams: Send

Purpose: Check if BDUT sends with correct Transport Control field (TPCI).
Note: LTE-addressed telegrams use an own TPCI-coding (T_Data_Tag_Group).

Procedure: Stimulate BDUT to send LTE-addressed frames (incl. LTE Broadcast):

a) LTE-HEE server output Datapoint: with A_GroupPropValue_InfoReport Service

b) LTE-HEE client output Datapoint: with A_GroupPropValue_Write Service

c) Read request **from** a BDUT´s LTE-HEE client Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Read Service

d) Response **from** a BDUT´s LTE-HEE server Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Response Service.
Stimuli: send with A_GroupPropValue_Read Service to a BDUT´s HEE-LTE (server) Datapoint.

Acceptance: BDUT sends LTE-HEE Datapoints with correct TPCI-coding

Note: This test can be done implicitly with the APCI-coding tests according clause 4.1.1.

3.6.1.2 LTE-addressed telegrams: Receive

Purpose: Check BDUT´s acceptance of LTE-addressed frames with correct TPCI-coding.
Note: LTE-addressed telegrams use an own TPCI-coding (T_Data_Tag_Group).

Procedure: Use a telegram generator (e.g. EITT) to send LTE-addressed frames (incl. LTE Broadcast):

a) to a BDUT´s LTE-HEE client input Datapoint: with A_GroupPropValue_InfoReport

b) to a BDUT´s LTE-HEE server input Datapoint: with A_GroupPropValue_Write

c) Read request **to** a BDUT´s LTE-HEE server Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Read

d) Response **to** a BDUT´s LTE-HEE client Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Response.
Stimuli: Read Request from a BDUT´s HEE-LTE (client) Datapoint.
Note: telegram generator shall respond before BDUT´s response timeout!

Acceptance: BDUT accepts the frames.

Note: This test can be done implicitly with the APCI-coding tests according clause 4.1.1.

3.6.1.3 LTE-addressed telegrams: Receive with incorrect TPCI coding

Purpose: Check BDUT´s acceptance of LTE-addressed frames with incorrect TPCI-coding.
Note: LTE-addressed telegrams use an own TPCI-coding (T_Data_Tag_Group).

Procedure: Use a telegram generator (e.g. EITT) to send LTE-addressed frames with incorrect TPCI coding:

a) to a BDUT´s LTE-HEE client input Datapoint: with A_GroupPropValue_InfoReport

b) to a BDUT´s LTE-HEE server input Datapoint: with A_GroupPropValue_Write

c) Read request **to** a BDUT´s LTE-HEE server Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Read

d) Response **to** a BDUT´s LTE-HEE client Datapoint (input and/or output, whatever is applicable in the BDUT): with A_GroupPropValue_Response.

Stimuli (examples using invalid combinations of TPCI/APCI):

a) IN *34 E4 AFFE 04 1107* ***03 EB*** *… (LTE-HEE Datapoint’s property value)*
Note: APCI: A_GroupPropValue_InfoReport; TPCI: T_Data_Group

b) IN *34 E4 AFFE 0411 08* ***03 EA*** *…(to LTE-HEE server input Datapoint)*
Note: APCI: A_GroupPropValue_Write; TPCI: T_Data_Group

c) IN *34 E4 AFFE 0411 08* ***03 E8*** *…(to LTE-HEE server output Datapoint)*
Note: APCI: A_GroupPropValue_Read; TPCI: T_Data_Group

d) IN *34 E4 AFFE 0411 08* ***03 E9*** *…(to LTE-HEE client input Datapoint)*
Note: APCI: A_GroupPropValue_Response; TPCI: T_Data_Group

Acceptance:BDUT does not accept the frames. Check BDUT’s behaviour, e.g. by reading back properties (in case of InfoReport).

Expected reactions to all stimuli examples above: BDUT does not take over the values.

3.6.1.4 Group addressed telegrams – Send and Receive

Telegrams using T_Data_Group Service/TPCI:

- Group addressed telegrams using extended frame format (APDU-length > 15): checked implicitly with AIL group object tests, using group objects with APDU > 15

- No (additional) tests.

3.6.2 Unicast / Device oriented communication – connected

- No (additional) tests.

3.6.3 Unicast / Device oriented communication – connectionless

Telegrams using T_Data_Individual Service/TPCI:

- Individual addressed telegrams using extended format (APDU-length > 15):
  Tests are done implicitly with (system) interface object tests using A_PropertyValue_Write/Read/ Response Services, and also with management service testing (e.g. property server tests).

- No other (additional) tests.

3.6.4 Broadcast communication

Telegrams using T_Data_Broadcast Service/TPCI (= telegrams with T_Data_Group TPCI and with destination address = 0000h):

- The current [7] KNX system specification does not foresee any transmission using T_Data_Broadcast Service/TPCI **and** extended frame format (APDU-length > 15 octets). Therefore

- No (additional) tests.

Note: LTE-Broadcast-addressed telegrams are considered (and tested) as multicast telegrams; see clause 3.6.1 for testing. LTE-Broadcast-addressed telegrams use the T_Data_Tag_Group Service-TPCI for transmission.

[7] Current state: Spring 2003

4 Application (Interface) Layer Tests

To be integrated into KNX Handbook Series, Test Suite Supplement to Volume 10, Part 1.

4.1 LTE-HEE (Application) Interface Objects

The following tests shall be done using a sample application on the BDUT, or comparable means.

4.1.1 APCI Tests: Group Property Value AL-Services

In case the stack supports private Datapoints for LTE runtime interworking, the following tests shall be carried out with private Datapoints.

4.1.1.1 BDUT sends A_GroupPropValue_Read

Purpose: Check if BDUT sends with correct Application Control Field (APCI) and, for the test according 3.6.1.1, with correct Transport Control Field (TPCI).

Note: Test is applicable only for LTE-HEE client input Datapoints

Procedure: Stimulate BDUT to send an LTE-addressed frame with A_GroupPropValue_Read Service.

Acceptance: BDUT sends the frame with correct APCI-coding

Expected reaction to the stimulus (example with Object Type = 100, Property 51):
OUT 34 E4 1101 0411 05 ***07 E8*** *00 64 00 33*

4.1.1.2 BDUT receives A_GroupPropValue_Read

Purpose: Check if BDUT accepts LTE-addressed frames with correct APCI-coding and, for the test according 3.6.1.2, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE server output Datapoints

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame with A_GroupPropValue_Read Service to BDUT.

Stimulus (example with Object Type = 384, Property 254 = private Datapoint):
IN 34 E4 AFFE 0411 08 ***07 E8*** *01 80 00 FF 00 FD FE*

Acceptance:BDUT accepts the frames, i.e. sends corresponding A_GroupPropValue_Response

Expected reaction to the stimulus example above:
OUT 34 E4 1101 0411 0A ***07 E9*** *01 80 01 FF 00 FD FE 00 00*

Note: This test can be carried out in one sequence together with the test according clause 4.1.1.5.

4.1.1.3 BDUT sends A_GroupPropValue_Write

Purpose: Check if BDUT sends with correct APCI-coding and, for the test according 3.6.1.1, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE client output Datapoints

Procedure: Stimulate BDUT to send an LTE-addressed frame with A_GroupPropValue_Write Service

Acceptance: BDUT sends the frame with correct APCI- & TPCI-coding

Expected reaction to the stimulus (example with Object Type = 50102, Property 61):
OUT 34 E5 1101 1234 0C ***07 EA*** *C3 B6 00 FF 00 FD 3D 66 BB 01 00*

4.1.1.4 BDUT receives A_GroupPropValue_Write

Purpose: Check if BDUT accepts LTE-addressed frames with correct APCI-coding and, for the test according 3.6.1.2, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE server input Datapoints

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame with A_GroupPropValue_Write Service to BDUT.

Stimulus (example with Object Type = 50102, Property 60 = private Datapoint):
IN 34 E5 AFFE 1234 0C ***07 EA*** *C3 B6 01 FF 00 FD 3C 77 CC 01 00*

Acceptance: BDUT accepts the frames. Check by reading back the property value with A_PropertyValue_Read/Response services

4.1.1.5 BDUT sends A_GroupPropValue_Response

Purpose: Check if BDUT sends with correct APCI-coding and, for the test according 3.6.1.1, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE server output Datapoints

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame with A_GroupPropValue_Read Service to BDUT. BDUT sends the response using A_GroupPropValue_Response Service.

Acceptance: BDUT sends the response with correct APCI- & TPCI-coding. Check the response to the stimulus in the test according clause 4.1.1.2.

4.1.1.6 BDUT receives A_GroupPropValue_Response

Purpose: Check if BDUT accepts LTE-addressed frames with correct APCI-coding and, for the test according 3.6.1.2, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE client input Datapoints. Use an LTE-HEE Datapoint with known reaction for this test.

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame with A_GroupPropValue_Response to BDUT.

Stimulus (example with Object Type = 100, Property 51):
IN *34 E4 AFFE 0411 07* ***07 E9*** *00 64 01 33 02 00*

Acceptance: BDUT accepts the frames. Check by BDUT's reaction.

4.1.1.7 BDUT sends A_GroupPropValue_InfoReport

Purpose: Check if BDUT sends with correct APCI-coding and, for the test according 3.6.1.1, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE server output Datapoints

Procedure: Stimulate BDUT to send an LTE-addressed frame with A_GroupPropValue_InfoReport Service

Acceptance: BDUT sends the frame with correct APCI- & TPCI-coding

Expected reaction to the stimulus (example with Object Type = 384, Property 54):
OUT *34 E4 1101 0411 06* ***07 EB*** *01 80 01 36 01*

4.1.1.8 BDUT receives A_GroupPropValue_InfoReport

Purpose: Check if BDUT accepts LTE-addressed frames with correct APCI-coding and, for the test according3.6.1.2, with correct TPCI-coding.

Note: Test is applicable only for LTE-HEE client input Datapoints. Use an LTE-HEE Datapoint with known reaction for this test.

Procedure: Use a telegram generator (e.g. EITT) to send an LTE-addressed frame with A_GroupPropValue_InfoReport Service to BDUT.

Stimulus (example with Object Type = 100, Property 51):
IN *34 E4 AFFE 0411 07* ***07 EB*** *00 64 01 33 03 00*

Acceptance: BDUT accepts the frames. Check by BDUT's reaction

4.1.1.9 BDUT receives invalid APCI

Purpose: In addition to the test in chapter 8-3-7, clause 1.3.1.6, the purpose of this test is to check whether unsupported APCI's and APCI's, which are invalid for LTE group communication, are rejected by the BDUT. The tests ensure that the BDUT does not generate a reaction on the bus and that invalid or unsupported APCI do not update the corresponding LTE-HEE Datapoint. Any other internal reactions to invalid or unsupported APCI's are however not tested and reside under manufacturer's responsibility.

Note: LTE-addressed telegrams use an own TPCI-coding (T_Data_Tag_Group).

Procedure: Use a telegram generator (e.g. EITT) to send LTE-addressed frames with incorrect APCI, but correct TPCI coding, correct IO Type & Property ID and with:

a) invalid combinations in the 4 most significant APCI-bits.

b) invalid APCI-codings out of the APCI-range of user messages

c) invalid APCI-codings that are reserved for future use (escape codes)

Stimuli, examples for case a):
*IN 34 E7 AFFE 0000 08 **04 00** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **04 40** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **04 80** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **05 80** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **05 C0** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 00** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 40** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 80** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **04 C0** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **05 00** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **05 40** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **06 00** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **06 40** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **06 80** C3 B4 00 FF 00 FD 33*

Stimuli, examples for case b):
*IN 34 E7 AFFE 0000 08 **06 C0** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **06 E8** C3 B4 00 FF 00 FD 33*

Stimuli, examples for case c):
*IN 34 E7 AFFE 0000 08 **07 D5** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 D7** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 D8** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 DC** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 DF** C3 B4 00 FF 00 FD 33*
*IN 34 E7 AFFE 0000 08 **07 E5** C3 B4 00 FF 00 FD 33*

Acceptance: BDUT does not accept the frames. Check e.g. by the following means:

- BDUT must not show any visual reaction: visual control during the telegram generator sends the frames

- BDUT must not send any telegrams as responses onto the bus (check with busmonitor)

- Reading back property values: BDUT must not have changed any property value due to the received invalid frames.

4.1.1.10 BDUT receives invalid data length

Purpose: Check whether the LTE-HEE Datapoints implemented in BDUT reject a value write/response addressed to them, of which the indicated info length does not match their own supported field types (Datapoint/property data type). E.g. an LTE-HEE Datapoint with a 3 octets long Datapoint type receives on its attributed LTE address a value-write with info length ≠ 3 octets.

Note: The read request service does not contain any data
InfoReport, Write and Response Services do contain data with a length of at least 1 octet.

Stimuli: Try to read from an LTE-HEE server Datapoint and try to set LTE-HEE Datapoint to new value, with data length larger than size of the regular Datapoint type.

Stimuli (examples):
IN 34 E7 AFFE 0000 09 07 E8 C3 B4 00 FF 00 FD 33 ***00****: GrpPropValueRead*
IN 34 E7 AFFE 0411 08 07 E9 00 64 01 33 01 00 ***00****: GrpPropValueResponse*
IN 34 E7 AFFE 0411 08 07 EB 00 64 01 33 02 00 ***00****: GrpPropValueInfoReport*

Stimuli: Try to read from an LTE-HEE server Datapoint and try to set LTE-HEE Datapoint to new value, with data length smaller than size of the regular Datapoint type.

Stimuli (examples):
IN 34 E7 AFFE 0000 07 07 E8 C3 B4 00 FF 00 FD ~~33~~: GrpPropValueRead
IN 34 E7 AFFE 0411 06 07 E9 00 64 01 33 01 ~~00~~: GrpPropValueResponse
IN 34 E7 AFFE 0411 06 07 EB 00 64 01 33 02 ~~00~~: GrpPropValueInfoReport

Acceptance: BDUT does not accept the frames. Check e.g. by the following means:

- BDUT does not send a response in case of GrpPropValueRead-stimulus.
- BDUT must not show any visual reaction: visual control during the telegram generator sends the frames
- Reading back property values: BDUT must not have changed any property value due to the received invalid frames.

The tests shall be performed for different (>1) value field types, with *GrpPropValueRead* and with at least one of the other LTE-AL-Services (*GrpPropValueResponse,* GrpPropValueWrite, *GrpPropValueInfoReport*).

4.1.2 Point-to-point connectionless communication Mode

The following two tests clauses 4.1.2.1& 4.1.2.2are intended to check whether LTE-HEE server Datapoints are accessible with A_PropertyValue_Read/Write Services.

4.1.2.1 BDUT receives A_PropertyValue_Read

Purpose Check BDUT´s acceptance of point-to-point addressed property read frames

Procedure: Use a telegram generator (e.g. EITT) to send Property Value Read frames to BDUT’s LTE-HEE server Datapoints. Tests shall be done with a few (>1) sample Datapoints.

Note: Manufacturers must declare the implemented properties of application interface object within the PIXIT submitted for testing.

Stimulus (IN): *A_PropertyValue_Read to an LTE-HEE output server Datapoint*

Acceptance: BDUT sends expected responses (with A_PropertyValue_Response-Service)

Reaction (OUT): *A_PropertyValue_Response from the LTE-HEE server Datapoint*

Note: The test can be done implicitly within the LTE-Addressing tests according clause 4.1.3, if reading back of property values is used for acceptance checks.

4.1.2.2 BDUT receives A_PropertyValue_Write

Purpose Check BDUT´s acceptance of point-to-point addressed property write frames

Procedure: Use a telegram generator (e.g. EITT) to send Property Value Write frames to BDUT's LTE-HEE server Datapoints that have property write access enabled. The Test shall be done for at least one sample Datapoint.

Note: Manufacturers must declare the implemented properties of application interface objects within the PIXIT submitted for testing.

Stimulus (example: A_PropValue_Write to an LTE-HEE server input Datapoint):
IN BC AFFE 1101 67 03 D7 0D 33 10 01 02 00

Acceptance: BDUT sends expected responses (with A_PropertyValue_Response-Service)

Expected reaction to the stimulus above (A_PropertyValue_Response):
OUT BC 1101 AFFE 67 03 D6 0D 33 10 01 02 00

4.1.3 LTE Adressing

4.1.3.1 Introduction

For the following LTE address tests, the used LTE-AL-Service is free of choice.

4.1.3.2 Testing of LTE-HEE Group Address Extension

4.1.3.2.1 Mapping of Tags -Send

Purpose: Check if BDUT maps different tag types to the correct EFF and destination (group) address.

Note: EFF and tag/destination (group) address are set by the application. In this sense, it isn't relevant if the tag is of type *geographical*, *application specific* or *unassigned*. This test ensures that tags, as set by the application (LTE-address parameters) are mapped correctly into the outgoing frames.

Procedure: Stimulate BDUT to send LTE-addressed frames with the different tag types:
- geographical tags type 1 & 2; including also wildcard addressing, if applicable
- application specific tags, including the HVAC-broadcast and wildcards
- unassigned tags, including the LTE-Broadcast

At the end of the test procedure, reset BDUT´s LTE-address parameters to the default values.

Acceptance: BDUT sends with expected EFF and tags/destination (group) addresses.

Note: Geographical Tag Type 1: Apartments 1 to 63 are mapped to EFF=0100b and group addresses C 1-63d, 0=Wildcard
Geographical Tag Type 2: Apartments 64 to 126 to EFF=0101b and group addresses (bit15…bit10) 1-63d, 0=Wildcard [8]

4.1.3.2.2 Mapping of Geographical Tags – Receive

Purpose: Check if BDUT maps received EFF and destination group address to the correct geographical tag.

[8] Please refer to Volume 10, Part 1, clause 7.2

Procedure: Use a telegram generator to send an LTE-addressed frame (with geographical tag) to BDUT. Choose an LTE-HEE Datapoint with a geographical tag, which is used by BDUT, and for which the reaction can be observed easily, e.g. send an LTE-Read (A_GroupPropValue_Read service) to a BDUT's LTE-HEE server output Datapoint.

Change LTE-address parameter (according BDUT´s product documentation) and send again a frame to the same Datapoint as before, but also with changed address.
Repeat this test step at least once for each geographical tag type (see note below).

At the end of the test procedure, reset BDUT´s LTE-address parameters to the default values.

Acceptance: BDUT accepts the frames. Check the expected BDUT´s reaction. If A_GroupPropValue_Read was used as stimulus, check if the expected response is sent by BDUT.

Note: Apartments 1 to 63 are mapped to EFF=0100b and group addresses (bit15…bit10) 1-63d (0=Wildcard), Apartments 64 to 126 to EFF=0101b and group addresses 1-63d (0=Wildcard)

4.1.3.2.3 Mapping of Application Specific Tags – Receive

Purpose: Check if BDUT maps received EFF and destination group address to the correct application specific tag.

Note: This test shall be done for different (>1, if applicable) sub-types of application specific tags.

Procedure: Use a telegram generator to send an LTE-addressed frame (with application specific tag) to BDUT. Choose an LTE-HEE Datapoint with an application specific tag, which is used by BDUT, and for which the reaction can be observed easily, e.g. send an LTE-Read (A_GroupPropValue_Read service) to a BDUT's LTE-HEE server output Datapoint.

Change LTE address, according BDUT´s product documentation, and send again a frame to the BDUT to same Datapoint as before, but also with changed LTE address.

At the end of the test procedure, reset BDUT´s LTE-address parameters to the default values.

Acceptance: BDUT accepts the frame.If A_GroupPropValue_Read was used as stimulus, check if the expected response is sent by BDUT.

4.1.3.2.4 Mapping of Unassigned Peripheral Tags - Receive

Purpose: Check if BDUT maps received EFF and destination group address to the correct unassigned peripheral tag.

Procedure: Use a telegram generator to send an LTE-addressed frame (with unassigned peripheral tag) to BDUT. Choose an LTE-HEE Datapoint with an unassigned peripheral tag, which is used by BDUT, and for which the reaction can be observed easily, e.g. send an LTE-Read (A_GroupPropValue_Read service) to a BDUT's LTE-HEE server output Datapoint.

Change LTE address, according BDUT´s product documentation, and send again a frame to the BDUT to same Datapoint as before, but also with changed LTE address.

At the end of the test procedure, reset BDUT´s LTE-address parameters to the default values.

Acceptance: BDUT accepts the frames. If A_GroupPropValue_Read was used as stimulus, check if the expected response is sent by BDUT.

4.1.3.2.5 Mapping of Tags – Receive with incorrect tags

Purpose: Check if BDUT rejects frames with tags that are not used by the BDUT.

Procedure: Use a telegram generator to send LTE-addressed frames with tags (geographical, application specific, unassigned) currently not used by BDUT. Choose an LTE-HEE Datapoint for which the no-reaction can be observed easily, e.g. send an LTE-Read (A_GroupPropValue_Read service) to a BDUT's LTE-HEE server output Datapoint.

Acceptance: BDUT does not accept the frame. Check by the expected BDUT´s (no-) reaction. If A_GroupPropValue_Read was used as stimulus, check if no response is sent by BDUT.

4.1.3.3 Testing of LTE HVAC-Broadcast

Purpose: Check BDUT´s proper use of received frames with HVAC broadcast address as LTE destination address. One LTE-address (0000h) within the range of application specific tags is reserved to distribute messages, which are common within the HVAC application domain.

Note: The usage of the HVAC broadcast together with LTE InfoReport or Write service is specified at Datapoint level in the Application Specifications and restricted to a defined set of Datapoints.
In HVAC devices, a received HVAC broadcast message will be accepted at data link layer level, but the Datapoint is NOT a priori accepted at the level of application. LTE Read request using the HVAC broadcast address is also restricted to a defined set of Datapoints according to the HVAC Application Specifications. For only those Datapoints the corresponding Response shall be generated in the receivers.
In both cases the proper usage of the HVAC broadcast in combination with a Datapoint is checked in the receiving Application Interface Layer. See vol. 10, part 1.

Procedure: Send an HVAC-broadcast-addressed frame to a BDUT's LTE-HEE Datapoint, for which the HVAC-broadcast is allowed according the application specifications.

Stimulus (example with A_GroupPropValue_Read):
*IN 34 E**6** AFFE **0000** 05 07 E8 … (no applicable example yet)*

Acceptance: BDUT does accept the frames at the level of application, if LTE-HVAC-broadcast is foreseen for the Datapoint under test; it does not accept the frames, if LTE-HVAC-broadcast is not foreseen for the Datapoint under test
Check by the expected BDUT´s reaction, or no reaction.

Expected reaction to the stimulus above:
*OUT 34 E**6** 1101 **0000** 14 07 E9 … (expected property value)*

Procedure: Additional negative test: Try to send also an HVAC-broadcast addressed frame to an LTE-HEE Datapoint that has not enabled the HVAC-broadcast.

Stimulus (example with A_GroupPropValue_Read):
*IN 34 E**6** AFFE **0000** 05 07 E8 00 64 01 33*

Acceptance: BDUT does not accept the frame.

Expected reaction to the stimulus above: no response sent by BDUT.

4.1.3.4 Testing of LTE Broadcast

Purpose: Check BDUT´s proper use of received frames with “broadcast to all LTE”-address as LTE destination address. One LTE-address (0000h) within the range of unassigned (peripheral) tags is reserved to distribute messages, which are common within all application domains.

Note: The usage of the LTE broadcast together with LTE InfoReport or Write service is specified at Datapoint level in the Application Specifications and restricted to a defined set of Datapoints.
In LTE devices, a received LTE broadcast message will be accepted at data link layer level, but the Datapoint is NOT a priori accepted at the level of application. LTE Read request using the LTE broadcast address is also restricted to a defined set of Datapoints according to the HVAC Application Specifications. For only those Datapoints the corresponding Response shall be generated in the receivers.
In both cases the proper usage of the HVAC broadcast in combination with a Datapoint is checked in the receiving Application Interface Layer. See vol. 10, part 1.

Procedure: Send an LTE-broadcast-addressed frame to a BDUT’s LTE-HEE Datapoint, for which the LTE-broadcast is allowed according the application specifications.

Stimulus (example with A_GroupPropValue_Read):
*IN 34 E***7** *AFFE* **0000** *08 07 E8 C3 B4 00 FF 00 FD 33*

Acceptance: BDUT does accept the frame.

Expected reaction to the stimulus above:
*OUT 34 E***7** *1101* **0000** *14 07 E9 C3 B4 01 FF 00 FD 33… (property value)*

Procedure: Additional negative test: Try to send also an LTE-broadcast addressed frame to an LTE-HEE Datapoint that has not enabled the LTE broadcast.

Stimulus (example with A_GroupPropValue_Read):
*IN 34 E***7** *AFFE* **0000** *05 07 E8 00 64 01 33*

Acceptance: BDUT does not accept the frame.

Expected reaction to the stimulus above: no response sent by BDUT.

4.1.3.5 Testing of the Wildcard Functionality - Send

The test procedure according clause 4.1.3.2.1covers testing of the wildcard functionality.

4.1.3.6 Testing of the Wildcard Functionality - Receive

Purpose: Check if BDUT accepts wildcard addressed frames

Note: All LTE-HEE (input) Datapoints must accept wildcard-addressed frames, i.e. even if their LTE address is not set to the value in the received frame. Wildcard addresses are allowed for all levels within the structure of geographical tags, and within each type of application specific tags. No wildcard is possible for unassigned (peripheral) tags.

Procedure: Use a telegram generator to send wildcard-addressed frames to a BDUT´s LTE-HEE Datapoint. The test sequence shall consider different tag/LTE-address types (geographical tags 1 & 2, application specific tags), as far as applicable.

Note: The Wildcard on Apartment (highest level within the structure of geographical tags) must be accepted independent on the geographical tag type, 1 (EFF = 4) or 2 (EFF = 5), in the received frame. E.g. an input Datapoint, whose Apartment address is 100 (Type 2 geographical tag, EFF = 5) must accept the apartment wilcard (*) sent as a type 1 geographical tag with EFF = 4)

Stimuli (examples: GroupPropValueRead; BDUT's geogr. Tag set to A.R.S=1.1.1):

IN *34 E**4** AFFE **00**11 08 07 E8 01 80 00 FF 00 FD FE*

IN *34 E**5** AFFE **00**11 08 07 E8 01 80 00 FF 00 FD FE*

IN *34 E**4** AFFE 04**01** 08 07 E8 01 80 00 FF 00 FD FE*

IN *34 E**4** AFFE 041**0** 08 07 E8 01 80 00 FF 00 FD FE*

Acceptance: Check if BDUT accepts the wildcard-addressed frames.
Note: In case of A_GroupPropValue_Read, BDUT shall respond with A_GroupPropValue_Response containing precise LTE address (not the wildcard address).

Expected reaction to all stimuli examples above:

OUT *34 E**4** 1101 **0411** 0A 07 E9 01 80 00 FF 00 FD FE … (property value)*

Procedure: Negative Test: Use a generator to send wildcard-addressed frames to the BDUT, where only one part of the LTE address is wildcard addressed and the other part does not correspond with the BDUT's LTE address (e.g. *.2.1 sent to LTE-HEE input Datapoint with LTE address 1.1.1).

Note: "Negative" tests for the wildcard receiving functionality are applicable only for the structured LTE tag types; these are geographical tags and application specific tags, the unassigned tag type is not structured.

Stimuli (examples: GroupPropValueRead; BDUT's geogr. Tag set to A.R.S=1.1.1):

IN *34 E**4** AFFE **0021** 08 07 E8 01 80 00 FF 00 FD FE: A.R.S=*.2.1*

IN *34 E**4** AFFE **001F** 08 07 E8 01 80 00 FF 00 FD FE: A.R.S=*.1.15*

IN *34 E**4** AFFE **0C01** 08 07 E8 01 80 00 FF 00 FD FE: A.R.S=3.*.1*

IN *34 E**4** AFFE **0420** 08 07 E8 01 80 00 FF 00 FD FE: A.R.S=1.2.**

Acceptance: BDUT does not accept the (partial) wildcard-addressed frames.

Expected reaction to all stimuli examples above: no response sent by BDUT

4.1.3.7 Testing of the Sniffer Functionality

Purpose: Check BDUT´s Sniffer behavior.

Note: The usage of the Sniffer functionality with LTE InfoReport is specified at Datapoint level in the Application Specifications and restricted to a defined set of (input) Datapoints. Sniffer functionality is only applicable with InfoReport and Response services. A “Response Sniffer” usually is used after a “Wildcard Read”. See vol. 10, part 1.

Procedure: Test shall be done for sample Datapoints that have Sniffer functionality enabled.

Use a telegram generator to send an InfoReport to BDUT with an LTE destination address that is in the range of the LTE-HEE client input Datapoint’s sniffed LTE addresses.

Stimuli (examples: GroupPropValueInfoReport; for a BDUT LTE-HEE client input Datapoint that sniffs on Room- and Subzone-Level):
IN 34 E4 AFFE 0411 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 0421 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 041F 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 042F 08 07 EB ... (no applicable example known yet)

Acceptance: BDUT does accept the frames. Check by BDUT’s expected reaction.

Procedure: Stimulate BDUT to send a wildcard addressed frame with A_GroupPropValue_Read service and use a telegram generator to send the response to BDUT with precise LTE address that is in the range of the LTE-HEE Datapoint’s sniffed LTE Group addresses.

Stimuli (examples: GroupPropValueRead; for BDUT that sniffs on Room- and Subzone-Level, response from telegram generator tool):
OUT 34 E4 1101 0400 08 07 E8 ...(no applicable example of a response-sniffing LTE-HEE Datapoint known yet)
IN 34 E4 AFFE 0411 08 07 E9 ... (Telegram Generator sends with 1.1.1)

Acceptance: BDUT does accept the responses. Check by BDUT’s expected reaction.

Procedure: Negative Test: Use a telegram generator to send LTE-addressed frames to the BDUT, where only one part of the LTE address is sniffed by the BDUT’s LTE-HEE input Datapoint and other part(s) of the LTE-address does not match (e.g. geographical tag 1.1.1 sent to a BDUT’s LTE-HEE input Datapoint sniffing on subzone-level but its LTE address set to 2.1.x;

Note: “Negative” tests for the sniffer functionality are applicable only for the structured LTE tag types; these are geographical tags and application specific tags, the unassigned tag type is not structured.

Stimuli (examples: GroupPropValueInfoReport; for a BDUT LTE-HEE client input Datapoint that sniffs on subzone-level only, and apartment A=2, room R=1):
IN 34 E4 AFFE 0411 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 0421 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 041F 08 07 EB ... (no applicable example known yet)
IN 34 E4 AFFE 042F 08 07 EB ... (no applicable example known yet)

Acceptance: BDUT does not accept the frames.

4.2 System Interface Objects

It’s intended for the future, that the Device Editor Tool (“DevEdit”, a property client tool, which is part of the ETS3 tools) is used to perform the interface object tests:

- Reading of all properties of all interface objects is very easy. The procedure of reading a property includes reading of the property description
- Writing to properties shall be done for some sample Datapoints, that have write access enabled for writing with the A_PropertyValue_Write-service.

Tests of the system interface objects will be performed this way as soon as the DevEdit Tool is released with the needed functionality.
The clauses 4.2.1 and following are informative. The tests in clause 4.2.2are examples for testing of properties of the LTE address table system interface objects.

See clauses 4.2.1 in [13] for an introduction to the general test case.

4.2.1 Device Object

- No (additional) tests.

4.2.2 Address Table Object

LTE Devices have more than one instance of the address table object: one for the standard mode group address table with the standard group addresses, and one for each used type of LTE addresses (Geographical Tag Types 1 & 2, Application Specific Tags, General Peripheral Tags).
Each instance of an address table object shall be tested according the following test sequences.

4.2.2.1 PID_OBJECT_TYPE (1)

The test sequence as described in [13], clause 4.2.2.1 applies, with adapted local Object Index and Object Type=1 (Address Table Object)

4.2.2.2 PID_OBJECT_NAME (2)

The test sequence as described in [13], clause 4.2.2.2 applies, with adapted local Object Index and Object Type=1 (Address Table Object)

4.2.2.3 PID_LOAD_STATE_CONTROL (5)

The load control property shall be tested according the test sequences described in [13], clause 5.4.2 (management server testing of Load State Machines).

4.2.2.4 PID_TABLE (23)

Property PID_TABLE is not present in LTE address tables. Therefore: no (additional) tests.

4.2.2.5 PID_EXT_FRAMEFORMAT (51)

Note: For the first generation of LTE Devices, the extended frame format property has read-only access. For later extension to write access, corresponding tests shall be added.

Purpose: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT
Test frame (IN): A_PropertyDescription_Read (ObjIndex=Index of Address Table Object; PropID=51d; PropIndex=0)

Acceptance: BDUT sends A_PropertyDescription_Response with correct data
Test frame (OUT): A_PropertyDescription_Response (ObjIndex= Index of Address Table Object; PropID=51d; PropIndex=0; property data type=PDT_Unsigned_Char; max. nr. of elements=1; write access=disabled; read access=enabled)

Purpose: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT
Test frame (IN): A_PropertyValue_Read (ObjIndex=Index of Address Table Object; PropID=51d; Count =1; Start=001)

Acceptance: BDUT sends A_PropertyValue_Response with correct data

Test frame (OUT): A_PropertyValue_Response (ObjIndex=Index of Address Table Object; PropID=51d; Count =1; Start=001; Property Data=extended frame format type which is valid for property 52 (AddrTab1))

4.2.2.6 PID_EXT_ADDRTAB1 (52)

Note: For the first generation of LTE Devices, the extended address table properties have read-only access. For later extension to write access, corresponding tests shall be added.

Purpose: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT
Test frame (IN): A_PropertyDescription_Read (ObjIndex=Index of Address Table Object; PropID=52 PropIndex=0)

Acceptance: BDUT sends A_PropertyDescription_Response with correct data
Test frame (OUT): A_PropertyDescription_Response (ObjIndex= Index of Address Table Object; PropID=52; PropIndex=0; property data type=PDT_Generic_04[]; max. nr. of elements; write access=disabled; read access=enabled)
Note: maximum number of elements is depending on implementation. The BDUT´s manufacturer shall declare the maximum number in the datasheet and/or PICS/PIXIT.

Purpose: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT to read number of valid array elements
Test frame (IN): A_PropertyValue_Read (ObjIndex=Index of Address Table Object; PropID=52d; Count =1; Start=000)

Acceptance: BDUT sends A_PropertyValue_Response with number of valid array elements

Test frame (OUT): A_PropertyValue_Response (ObjIndex=Index of Address Table Object; PropID=52h; Count =1; Start=000; Property Data=number of valid array elements)

FOR x=1 TO valid number of array elements DO:

Stimuli: send A_PropertyValue_Read to BDUT
Test frame (IN): A_PropertyValue_Read (ObjIndex=Index of Address Table Object; PropID=52d; Count =1; Start=x)

Note: Test can be done with adapted count (number of elements) and start index, according the number of valid array elements.

Acceptance: BDUT sends A_PropertyValue_Response with correct data:

Test frame (OUT): A_PropertyValue_Response (ObjIndex=Index of Address Table Object; PropID=52d; Count =1; Start=x; Property Data=(part of) Extended Address Table (AddrTab1)

5 Guidelines for Interworking Tests & Testing of Datapoint types

To be integrated into KNX Handbook Series, Test Suite Supplement to Volume 10, Part 1.

5.1 Testing of Datapoints of (LTE-HEE Application) Functional Blocks

5.1.1 Introduction

For complex devices (e.g. with a large amount of runtime interworking Datapoints and parameters and diagnostic property values), a complete testing according the following paragraphs (5.1.3.1to 5.1.3.4, 5.1.4, 5.1.5, 5.2) may result in an unreasonable certification effort.

In such a case, the following prioritization shall be taken into account during the tests

- high priority: test of runtime interworking Datapoints (group objects and than LTE-Datapoints)
- medium priority: test of properties available as ETS parameters
- low priority: test of other parameters & diagnostic value properties

5.1.2 Testing of Standard Mode Group Objects

Tests shall be done according to the test specifications & guidelines in chapter 8-3-7, “Interworking Tests”.

5.1.3 Testing of LTE-HEE Datapoints

For a better understanding of the test guidelines following in clauses 5.1.3.1to 5.1.3.4, the following figure shall illustrate the client/server process data (interworking) model used for runtime interworking between LTE-HEE-mode devices.

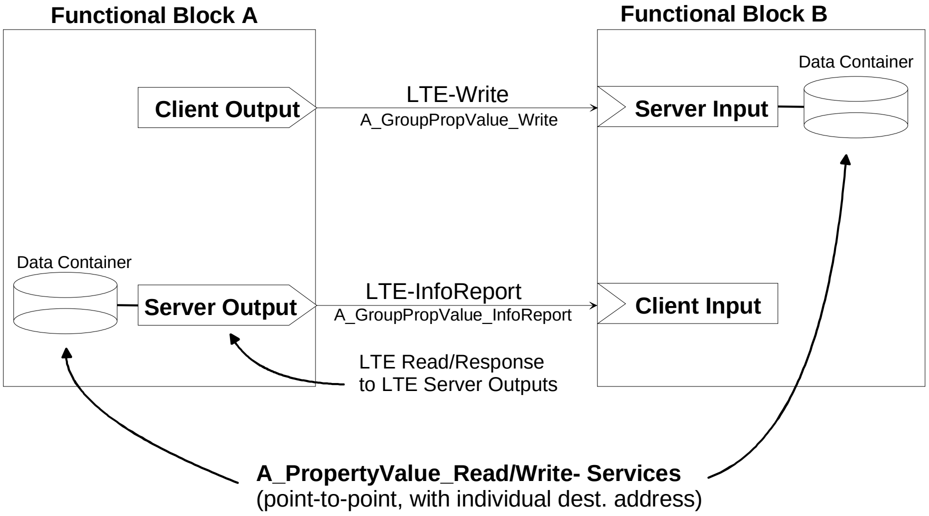

Figure 1: LTE-HEE client/server process data model

For Datapoint connections, where the receiving functional block (FB) is the content owner of the Datapoint value, the receiving FB is the server (data container). The Datapoint value is transmitted with LTE-Write service *A_GroupPropValue_Write* service. The sending FB is the client.

For Datapoint connections, where the sending functional block (FB) is the content owner of the Datapoint value, the sending FB is the server (data container). The Datapoint value is transmitted with LTE-InfoReport service *A_GroupPropValue_InfoReport* service.

An LTE-Read request (*A_GroupPropValue_Read.indication* and *A_GroupPropValue_Response.request*) is applicable only to server output Datapoints.

According volume 10, part 1, clauses 6.3.2 & 6.3.3, LTE Read access to LTE Write Inputs is not a general mechanism and is per default not supported. If LTE Write Inputs have LTE read-back capability, this feature shall be defined in the application specification. This LTE read-back capability is defined nowhere in the current [9] application specifications in Volume 7.

The value of server Datapoints (LTE Write Inputs and LTE InfoReport Outputs) is always accessible using Property Services and individual addressing.

5.1.3.1 LTE-HEE Client Input Datapoints

A test sequence for an LTE-HEE client input Datapoint shall contain:

- Test of allowed AL-Services: A_GroupPropValue_InfoReport & A_GroupPropValue_Read/ Response
- Test of Object Type, Object Instance and Property ID within the LTE AL-service(s).
  Note: the used LTE AL-services must contain the server Datapoint's Object Type, instance and property ID.
  Note: in read service from client to server, the object instance must be set to 0. The server answers with as much responses as object instances of the asked object type are implemented with object instance value set accordingly.
- Test of received Datapoint values: allowed values & values out of allowed range
- Test of status Z8: behavior after reception of supported and not supported Z8 status values (applicable only for Datapoints with Z8 status in their Datapoint type)
- Test of behavior after a receive timeout (no heartbeat from server Datapoint)
- Test of default values, if possible
- Test with changed LTE address: change LTE address at least once and use a telegram generator to send frames to the Datapoint under test.
- Test with illegal LTE address: use a telegram generator to send frames to the Datapoint under test, with LTE address that does not fit with that one of the tested Datapoint
- Test BDUT´s behavior after reception of wildcard-addressed frames
  Note: wildcard-addressed frames always must be accepted.
- Test of Sniffer functionality, if applicable
- Change at least once each parameter that has influence on BDUT´s behavior when receiving the LTE-HEE Datapoint under test, and then check BDUT´s behavior again according the tests above (as far as reasonable)
- Test of powerup behavior: check Datapoint value after powerup, if possible
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.1.3.2 LTE-HEE Server Input Datapoints

A test sequence for an LTE-HEE server input Datapoint shall contain:

- Test of property server: read property description (A_PropertyDescription_Read/Response services), read value from property if read enabled (A_PropertyValue_Read/Response services), (try to) write to property if (not) write enabled (A_PropertyValue_Write/Response services)
- Test of allowed LTE AL-Service(s): A_GroupPropValue_Write (& A_GroupPropValue_Read/ Response, if specified in Datapoint description to be enabled)
- Test of Object Type, Object Instance and Property ID within the LTE AL-service(s).

[9] Current state: Spring 2003

- Test of received Datapoint values: allowed values & values out of allowed range
- Test of command Z8: behavior after reception of supported and not supported Z8 command values (applicable only for Datapoints with Z8 command in their Datapoint type)
- Test of behavior after a receive timeout (no heartbeat from client Datapoint)
- Test of default values: use A_PropertyValue_Read/Response to read the value
- Test with changed LTE address: change LTE address at least once and use a telegram generator to send frames to the Datapoint under test.
- Test with illegal LTE address: use a telegram generator to send frames to the Datapoint under test, with LTE address that does not fit with that one of the tested Datapoint
- Test BDUT´s behavior after reception of wildcard-addressed frames
  Note: wildcard-addressed frames always must be accepted.
- Change at least once each parameter that has influence on BDUT´s behavior when receiving the LTE-HEE Datapoint under test, and then check BDUT´s behavior again according the tests above (as far as reasonable)
- Test of powerup behavior: read Datapoint value after powerup using A_PropertyValue_Read/ Response services
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.1.3.3 LTE-HEE Server Output Datapoints

A test sequence for an LTE-HEE server output Datapoint shall contain:

- Test of property server: read property description (A_PropertyDescription_Read/Response services), read value from property if read enabled (A_PropertyValue_Read/Response services), (try to) write to property if (not) write enabled (A_PropertyValue_Write/Response services)
- Test of allowed LTE AL-Services: A_GroupPropValue_InfoReport & A_GroupPropValue_Read/Response
- Mapping of the LTE address (tag) to EFF and destination group address field
- Test of Object Type, Object Instance and Property ID within the LTE AL-service(s).
  Note: in read service from client to server, the object instance must be 0. The server answers with as much responses as object instances of the asked object type are implemented.
- Test of transmission priority
- Test of COV condition/value, heartbeat and minimum repetition time
- Test of status Z8 (applicable only for Datapoints with Z8 status in their Datapoint type)
- Test of default settings: “per default communicating”, LTE address, Datapoint value
- Test with changed LTE address: change LTE address at least once and stimulate BDUT to send frames, additionally use wildcard address if applicable (as declared in Datapoint description)
- Change at least once each parameter that has influence on BDUT´s behavior on the LTE-HEE Datapoint under test, and then check BDUT´s behavior again according the tests above (as far as reasonable)
- Test of behavior after startup (and powerdown/restart): value according Datapoint description, time until Datapoint is sent 1st time.
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.1.3.4 LTE-HEE Client Output Datapoints

A test sequence for an LTE-HEE client output Datapoint shall contain:

- Test of allowed AL-Service(s): A_GroupPropValue_Write (& A_GroupPropValue_Read/ Response, if specified in Datapoint description to be enabled for corresponding server Datapoint)
- Mapping of the LTE address (tag) to EFF and destination group address field
- Test of Object Type, Object Instance and Property ID within the LTE AL-service(s).
  Note: client output Datapoints must set the server Datapoint's Object Type, instance and property ID within the LTE AL-Service
- Test of transmission priority
- Test of COV condition/value, heartbeat and minimum repetition time
- Test of command Z8 (applicable only for Datapoints with Z8 command in their Datapoint type)
- Test of default settings: "per default communicating", LTE address, Datapoint value
- Test with changed LTE address: change LTE address at least once and stimulate BDUT to send frames, additionally use wildcard address if applicable (as declared in Datapoint description)
- Change at least once each parameter that has influence on BDUT´s behavior on the LTE-HEE Datapoint under test, and then check BDUT´s behavior again according the tests above (as far as reasonable)
- Test of behavior after startup (and powerdown/restart): value according Datapoint description, time until Datapoint is sent 1st time.
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.1.4 Testing of parameter properties

Parameter properties normally are write-enabled and read-enabled.

A test sequence for a parameter property shall contain:

  - Test of property server:
    - Read property description (A_PropertyDescription_Read/Response services)
    - Read value from property if read enabled (A_PropertyValue_Read/Response services)
    - Write to property (A_PropertyValue_Write/Response services)
    - For array-structured properties:
      - read/write number of valid array elements (array element with start index 0)
- Test of property ID: implicitly done with the property server tests above
- The property server tests above shall be done in connectionless communication mode
  - Test of parameter's value range:
    - Write allowed values to property
    - Try to write values out of the allowed range
  - Test of Z8 status/command (applicable only for parameters with Z8 status/command in their Datapoint type):
    - Write to property with different (supported and not supported) command values
    - Read from property to check Z8 status
- For LTE address (LTE addresses) only: test if wildcard address is possible (according the Datapoint description)
- Test of parameter value after powerup and after powerdown/restart
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.1.5 Testing of diagnostic value properties

Diagnostic value properties normally have read-only access.

A test sequence for a diagnostic parameter property shall contain:

- Test of property server:
  - Read property description (A_PropertyDescription_Read/Response services)
  - Read property value (A_PropertyValue_Read/Response services)
  - Try to write to property (A_PropertyValue_Write/Response services)
  - For array-structured properties:
    - read number of valid array elements (array element with start index 0)
- Test of property ID: implicitly done with the property server tests above
- Test connectionless communication mode: the property server tests above shall be done in connectionless communication mode
  - Test of Z8 status (applicable only for diagnostic values with Z8 status in their Datapoint type):
    - Read from property to check Z8 status
    - Write to property with different (supported and not supported) command values
- Test of Datapoint value after powerup and after powerdown/restart
- Test of exception handling and special features, if applicable (according to Datapoint description)

5.2 Testing of Combinations of Functional Blocks in one Device

It is possible that in a certain configuration of a device, not all of the implemented functional blocks are accessible from the bus. That means no communication is possible from/to bus with Datapoints of such a non-accessible functional block.

In such a case, interworking tests shall be performed with different BDUT configuration(s):

- Each functional block is active to/from bus at least once so that each Datapoint can be tested according the guidelines described in clause 5.1in this document.
- BDUT configuration is changed (by device parameterisation as described by BDUT's manufacturer in the product documentation) as much times as necessary to fulfill the preceding test requirement.

It's also possible, that communication (of some Datapoints) between functional blocks is only BDUT-internal, but not necessary from/to bus. Such configuration(s) shall be described in the PIXIT for interworking testing by BDUT's manufacturer.

5.3 Testing of (Application) Functional Blocks - Specialties

5.3.1 Property PID_ObjectType

The Property PID_ObjectType (PID = 1) is mandatory for each implemented interface object. It shall also be tested for each implemented (application) interface object. The test is done as part of the interworking and Datapoint (type) tests.

Test shall be done with a test sequence as that one described in [13], clause 4.2.2.1 for the Device Object's Object Type Property, with adapted local Object Index and Object Type.