# System Conformance Testing

## Medium Dependant Layers Tests

## RF Physical and Data Link Layer Tests

8

2

5

Summary

This document contains the description of KNX RF conformance Tests.

Version 02.02.01 is a KNX Approved Standard.

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

# Document updates

| Version | Date | Modifications |
|---|---|---|
| 1 | 2003.01.16 | Document creation |
| 2 | 2003.04.07 | Restructuring of the document, Corrections and Modifications |
| 3 | 2003.05.19 | Additions and Modifications |
| 4 | 2003.06.27 | Modifications |
| 5 | 2003.10.01 | Modifications according to  KonCert meeting 27 |
| 6 | 2003.10.10 | Finalisation of draft for KonCert meeting October 2003 |
| 7 | 2003.12.03 | Modifications according to KonCert meeting 28 and some more editorial changes – preparation for release for voting |
| 8 | 2004.06 | Resolution of comments from release for voting – preparation for final voting |
| AS | 2004.09 | No comments in final voting – publication as AS |
| 1.1 | 2009.06 | Readying document for release of V2.0 of the KNX specifications |
| WD20091223 | 2009.12.23 | Update with KNX RF multi specification |
| WD20100309 | 2010.03.09 | Update after first review |
| WD20100719 | 2010.07.19 | TFRF review |
| WD20100923 | 2010.09.23 | TFRF review after Insta comments |
| WD20101004 | 2010.10.04 | TFRF update after Koncert group review. Addition of KNX RF AU tests |
| WD20101012 | 2010.10.12 | Resolution of Koncert comments |
| WD20111010a | 2011.10.10 | TFRF update after chapter 3-2-5 update. |
| WD20111010b | 2011.10.11 | Clause 3.2.1.1.4 value correction |
| WD20111019a | 2011.10.19 | Final TFRF review |
| WD20111027a | 2011.10.27 | Editorial comments KonCert |
| 2.1 DP | 2011.10.28 | Readying for Release for Voting |
| 2.2 AS | 2013.08 | No comments in Release for Voting – publication as AS and integration of AN 152 and AN 154 – positive check that AN 121 was covered. |
| 02.02.01 | 2013.10.24 | Editorial updates for the publication of KNX Specifications 2.1. |

# References

| | | |
|---|---|---|
| [01] | Chapter 3/2/5 | “Radio Frequency” |
| [02] | Chapter 3/5/3 | “Configuration Procedures” |
| [03] | Volume 7 | “Application Descriptions” |
| [04] | Volume 8 | “KNX System Conformance Testing” |
| [05] | Chapter 8/3/4 | “Transport Layer Tests” |

| | |
|---|---|
| Filename: | 08_02_05 System Conformace Testing - RF PhL and LL Tests v02.02.01 AS.docx |
| Version: | 02.02.01 |
| Status: | Approved Standard |
| Savedate: | 2013.10.24 |
| Number of pages: | 149 |

# Contents

**1 Introduction ..... 5**

**2 Abbreviations ..... 6**

**3 RF Common Parts ..... 7**

3.1 Physical Layer Tests ..... 7

3.1.1 EMC- and ERM related tests ..... 10

3.1.2 Bit Error Rate level ..... 10

3.1.3 Testing of Tx Centre Frequency Tolerance ..... 10

3.1.4 Testing of max Tx Effective Radiated Power ..... 11

3.1.5 Testing of Duty Cycle ..... 11

3.1.6 Testing of FSK Deviation ..... 11

3.1.7 Testing max allowed bandwitdh ..... 13

3.1.8 Testing of Tx Chip Rate ..... 13

3.1.9 Testing of Tx Jitter per transition ..... 14

3.1.10 Testing of Rx Blocking Performance ..... 15

3.1.11 Testing of Rx Frequency Tolerance ..... 15

3.1.12 Testing of Rx Chip Rate Tolerance ..... 15

3.1.13 Testing of Rx radiated sensitivity ..... 16

3.2 Data Link Layer Tests (Physical Part) ..... 16

3.2.1 Transmitter Tests ..... 16

3.2.2 Receiver – Tests ..... 18

3.3 Data Link Layer Tests (Data Part) ..... 20

3.3.1 Transmitter Tests for end devices ..... 20

3.3.2 Receiver Tests for end devices ..... 21

3.4 Set-up considerations for Testing System-Software ..... 23

3.4.1 Test Set-up: ..... 23

3.4.2 Format of Telegram sequences: ..... 23

3.5 Transport Layer tests ..... 23

3.5.1 System-Broadcast with different TL Services (UDT, NDT, UCD, NCD) ..... 24

3.5.2 Domain-Broadcast with different TL Services (UDT, NDT, UCD, NCD) ..... 24

3.5.3 Test for T_Disconnect-Response on individual connection-oriented requests. ..... 24

3.6 Management Layer tests ..... 25

3.6.1 DomainAddressSernr-Read/-Write/-Response ..... 25

3.6.2 DeviceDescriptor (type 0 / type 2) ..... 26

3.6.3 FunctionPropertyCommand ..... 26

**4 RF Retransmitters ..... 28**

4.1.1 Physical Layer ..... 28

4.1.2 Data Link Layer ..... 28

4.1.3 Network Layer ..... 34

4.1.4 Transport Layer, Application Layer, Management Layer ..... 35

**5 RF PB-Mode ..... 36**

5.1 Characteristics ..... 36

5.1.1 Individual Address ..... 36

5.1.2 Group Addresses ..... 36

5.1.3 Undirectional Sequences ..... 36

5.2 Channel Setup for Easy-Mode Tests ..... 36

5.2.1 E-Mode Channel Configuration for PB-Mode and E-Mode Tests ... 37
5.3 "Teach-In" and "Teach-Out" of non-generic sensor channel ... 38
5.4 "Teach-In" of generic sensor channel ... 38
5.5 "Teach-Out" of generic sensor channel ... 39
5.6 Linking generic sensor channel with no matching Datapoints ... 40
5.7 Linking actuator channel ... 40
5.8 Actuator with "single connect" - Datapoints ... 41
5.9 Timeout behaviour ... 43
5.10 Behaviour on unexpected Frames ... 44
5.11 Incorrect number of Group Objects ... 46
5.12 Actuator with full association table ... 47
5.13 Sensor with full association table ... 48
5.14 StopLink behaviour ... 49
5.15 Sub function ... 51
5.16 Unidirectional sensor (normal case) ... 52
5.17 Optional PB-Services ... 53
5.18 Unidirectional sensor (error case) ... 54
5.19 PB-Mode on unidirectional transmitters ... 55
5.19.1 Link sequence ... 55
5.19.2 Generic channel ... 55
5.19.3 Device Descriptor Type 2 ... 56

**6 RF Other EASY-Modes (and S-Mode) ... 57**
6.1 ObjectValue (PID_PB_OBJVALUE) ... 57
6.2 ObjectLink (PID_PB_OBJLINK ) ... 58
6.3 SetGetParameter (PID_PB_PARAMETERS) ... 59
6.4 GetObjectAddress (PID_PB_OBJADDR ) ... 59

**7 Multi Easy Configuration Tests ... 61**
7.1 Generalities ... 61
7.1.1 Channel configuration for RF Multi Easy tests ... 61
7.1.2 Frame description ... 61
7.2 Easy Controler ... 62
7.2.1 RF Multi Interface object existence ... 62
7.2.2 RF Multi Type (PID_RF_MULTI_TYPE) ... 62
7.2.3 RF Multi Physical Features (PID_RF_MULTI_PHYSICAL_FEATURES) ... 62
7.2.4 RF Multi Call Channel (PID_RF_MULTI_CALL_CHANNEL) ... 63
7.2.5 RF Multi Object Link (PID_RF_MULTI_OBJECT_LINK) ... 65
7.2.6 RF Multi Extended Group Address Repeated (PID_RF_MULTI_EXT_GA_REPEATED) ... 74
7.2.7 Compatibility between RF 1.1, RF1.R, RF1.M ... 78
7.2.8 Configuration with mixed links ... 82
7.3 Push-button ... 87
7.3.1 Linking actuator ... 87
7.3.2 Linking sensor ... 106
7.3.3 Linking 2 actuators ... 119
7.3.4 Linking 3 actuators ... 127
7.3.5 Linking sensor with multiple actuators ... 135
7.3.6 Linking actuator Fast Ack ... 141

# 1 Introduction

Scope of this document is a description of the different test procedures used to ensure the conformance of devices to the KNX RF specification.

This document only contains test specifications for the RF extensions RF1, RF2, RF3 (see [01] for further details) and modifications of the KNX standard system. For tests related to the aspects of RF devices that belong to the standard system, please refer to the existing documents in [04].

As there are different profiles possible for KNX RF devices, this document is separated in the following parts:

- clause 3 "RF Common Parts”
- clause 4 “RF Retransmitters
- clause 5 “RF PB-Mode”
- clause 6 “RF Other EASY-Modes (and S-Mode)”

Clause 2 is divided into several parts related to the RF chosen system KNX RF Ready, BiBat, BiBat 2 or KNX RF Multi.

Similar to other KNX media, RF implementations are in most cases dividable into a common system component (operating system) and a device specific application.

The conformity of the RF system component is hereby verified with the help of a test application. A detailed description of this test application follows in clause 3.4.

# 2 Abbreviations

| **Abbreviation** | **Description** |
|---|---|
| ACKREP | Frame transmitted by a KNX RF Multi Retransmitter after the repetition of a frame containing a Fast Ack request. |
| BDUT | Bus Device Under Test |
| BER | Bit Error Ratio |
| DUT | Device Under Test |
| EMC | ElectroMagnetic Compatibility |
| ERM | Electronic Radio Matters |
| FSK | Frequency Shift Keying |
| NPRM | No Permanent Receiver mod. |
| PRM | Permanent Receiver |
| RF | Radio Frequency |
| RX | Receiver |
| TRX | Transceiver |
| TX | Transmitter |

# 3 RF Common Parts

## 3.1 Physical Layer Tests

The KNX-specific Physical Layer requirements (see [01]) shall be fulfilled. As such tests shall preferably be performed “on the medium”, this requires the use of accurate RF measurement tools and analyzing equipment.

If such measuring equipment is not available, calibrated RF devices with well-known accuracy may also be used to perform the underneath tests.

**Table 1 - Conformance table with limits**

| **Measurement** | **KNX RF1 systems** | | | | **KNX RF2 systems** | **KNX RF3 systems** |
|---|---|---|---|---|---|---|
| | **KNX RF1 Ready** | **KNX RF1 BiBat**<br>**KNX RF1 BiBat 2** | **KNX RF1 Multi**<br>**Fast channels** | **KNX RF1 Multi**<br>**Slow channels** | **KNX RF2 Ready** | **KNX RF3 Ready** |
| EMC §3.1.1 | √ | √ | √ | √ | √ [for Europe]<br>Otherwise see specific local regulation | |
| BER §3.1.2 | √ | √ | √ | √ | √ | √ |
| TX center frequency<br>§3.1.3 | Fc = F1 = 868,300 MHz<br>±25ppm | Fc = F1 = 868,300 MHz<br>F4 = 869.525MHz<br>±25ppm | Fc = F1 = 868,300 MHz<br>F2 = 868.950MHz<br>F3 = 869.850MHz<br>±25ppm | S1 = 869.850MHz<br>S2 = 869.525MHz<br>±25ppm | Fc = 433.500MHz<br>±25ppm | Fc = 433.500MHz<br>±25ppm |
| TX radiated power §3.1.4 | Max : +14dBm<br>Min: -3 dBm | Max : +14dBm<br>Min : -3 dBm | Max : +14dBm<br>Min : -3 dBm | Max : +14dBm<br>Min : -3 dBm | Max : +10dBm<br>Min : -3 dBm | Max : +7dBm<br>Min : -3 dBm |

| **Measurement** | **KNX RF1 systems** | | | | **KNX RF2 systems** | **KNX RF3 systems** |
|---|---|---|---|---|---|---|
| | **KNX RF1 Ready** | **KNX RF1 BiBat**<br>**KNX RF1 BiBat 2** | **KNX RF1 Multi**<br>**Fast channels** | **KNX RF1 Multi**<br>**Slow channels** | **KNX RF2 Ready** | **KNX RF3 Ready** |
| Max TX duty cycle §3.1.5 | 1% | F 1 : 1%<br>F4 : 10% | F1 : 1%<br>F2 : 0.1%<br>F3 : 100% if erp ≤5mW else 1% | S1 : 100% if erp ≤5mW else 1%<br>S2 : 10% | 1% | 1% |
| FSK Deviation §3.1.6 | ±48kHz to ±80kHz | ±48kHz to ±80kHz | ±48kHz to ±80kHz | ±20kHz to ±65kHz | ±48kHz to ±80kHz | ±48kHz to ±80kHz |
| Max allowed bandwidth<br>§3.1.7 | 600kHz | F1 : 600kHz<br>F4 : 250kHz | F1 : 500kHz<br>F2 : 500kHz<br>F3 : 300kHz | S1 : 300kHz<br>S2 : 250kHz | 500kHz | 400kHz |
| TX chip rate §3.1.8 | 32768cps ±1.5% | 32768cps ±1.5% | 32768cps ±1.5% | 16384cps ±1.5% | 32768cps ±1.5% | 32768cps ±1.5% |
| TX jitter per transition §3.1.9 | ±5µs | ±5µs | ±5µs | ±5µs | ±5µs | ±5µs |
| RX blocking performance §3.1.10 | Category 2 | Category 2 | Category 2 | Category 2 | Category 2 | Category 2 |

| Measurement | KNX RF1 systems | | | | KNX RF2 systems | KNX RF3 systems |
|---|---|---|---|---|---|---|
| | KNX RF1 Ready | KNX RF1 BiBat<br>KNX RF1 BiBat 2 | KNX RF1 Multi<br>Fast channels | KNX RF1 Multi<br>Slow channels | KNX RF2 Ready | KNX RF3 Ready |
| RX frequency tolerance §3.1.11 | ±25ppm<br>±60 ppm in case of metering receiver on Fc | ±25ppm<br>±60 ppm in case of metering receiver on Fc | ±25ppm<br>±60 ppm in case of metering receiver on Fc | ±25ppm | ±25ppm | ±25ppm |
| RX chip rate and minimal tolerance §3.1.12 | 32768cps ±2% | 32768cps ±2% | 32768cps ±2% | 16384cps ±2% | 32768cps ±2% | 32768cps ±2% |
| Rx sensitivity §3.1.13 | Typ : -95dBm<br>Min : -80dBm | Typ : -95dBm<br>Min : -80dBm | Typ : -95dBm<br>Min : -80dBm | Typ : -95dBm<br>Min : -80dBm | Typ : -95dBm<br>Min : -80dBm | Typ : -95dBm<br>Min : -80dBm |

### 3.1.1 EMC- and ERM related tests

EMC- and ERM related issues are not part of this test specification. These tests shall be performed according to European Standards listed under the R&TTE directive.

Primarily the harmonised standards EN 300 220 and EN 301 489 shall be fulfilled. These standards cover principle RF requirements (power, bandwidth, modulation etc.) and EMC-specific requirements for RF short-range devices (SRD).

At any rate the product-specific (non RF related) European standards shall also be fulfilled.

### 3.1.2 Bit Error Rate level

Test setup
To meet the requirements regarding the minimum input signal level of the receiver, depending on the test setup, the signal may be transmitted by RF Cable (coax cable connection between transmitter and receiver to avoid external influences or may consist in transmitting the RF Signal via an antenna. However, this last method requires a lot of calibration effort and can only be carried out in an absorber room or similar environment.

- Testing:
  RF Telegrams are transmitted with a signal level according to the minimum allowed signal level of the DUT plus 3 dB. DUT shall allow to check whether it has received the RF-Telegram without an error e.g. by switching a relay or sending a reply Telegram. It is necessary to transmit enough RF Telegrams in order to get a sufficient number of tests for a statistical evaluation. A number of at least 1000 RF Telegrams shall be transmitted. During the tests the influences of the receiving software are also checked.
  The bit error rate is calculated with the following formula : Bit Error Rate = (Number of faulty Telegrams) / (Number of Telegrams*(Bits/Telegram))

  Requirements:
  Bit Error Rate ≤ $10^{-4}$ [For radiated measurement in the optimum antenna direction]

### 3.1.3 Testing of Tx Centre Frequency Tolerance

- Test Set-up:
  Test temperatures: 0 °C, 25 °C, 45 °C

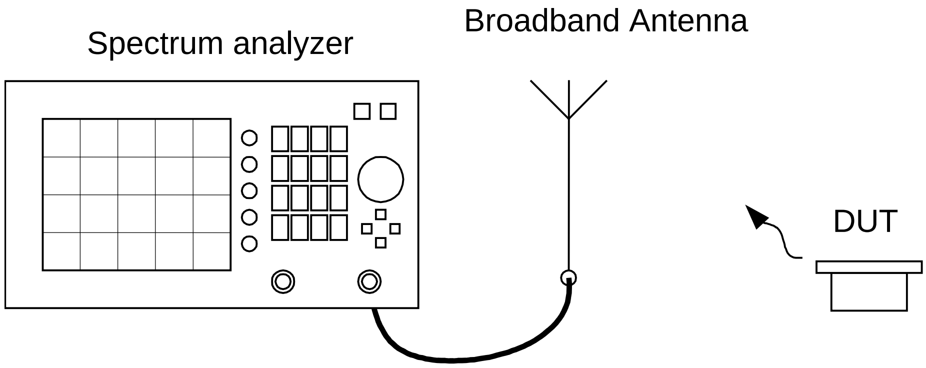

**Figure 1 - Example Test Set-up Tx Centre Frequency Tolerance**

- Testing:
  The centre frequency and its tolerance shall be measured with modulation switched off. The manufacturer shall state if the test mode provided allows the centre frequency to be measured directly, or if only the lower and upper modulation frequency can be measured. In the latter case the centre frequency is calculated as
  $f_c = (f0 + f1)/2$
  where f0 = frequency “0”, f1 = frequency “1”, fc = centre frequency

- Requirements:
  See Table 1 limits.

  If several RF channels are used, all nominal center frequencies shall be measured at 25 °C and the temperature drift can be done at one single frequency.

  KNX Physical Layer parameters shall be met on the entire product temperature range declared by the manufacturer in case it exceeds the range 0°C to 45°C (e.g. : -10°C to 70°C for outdoor usage).
- The manufacturer shall describe in a manufacturer test report how the measured tolerance is influenced by the frequency relevant components used (e.g. aging, temperature, initial tolerance of e.g. the used crystal) do not cause an overstepping of the limits.

### 3.1.4 Testing of max Tx Effective Radiated Power

- Test Setup:
  See EN 300 220-1
- Testing:
  See EN 300 220-1
- Requirements:
  See Table 1.
- EN 300-220 test report for the device may replace this test.

### 3.1.5 Testing of Duty Cycle

- Test Set-up:
  see EN 300 220-1; “Max Tx Duty Cycle”
- Testing:
  see EN 300 220-1; “Max Tx Duty Cycle”
- Requirements:
  See Table 1 limits.

  For repeater devices using RF channels with low Duty Cycle at 0,1 %, the duty cycle shall be software controlled.

### 3.1.6 Testing of FSK Deviation

- Test Set-up:
  see 3.1.3
- Testing:
  The DUT shall transmit a normal RF-Telegram. The demodulated FM-Signal shall be measured with a Spectrum Analyser (Option: FM-Demodulator)
- Requirements:
  - See Table 1 limits.
- For devices using RF channels Fc, F2, F3 or F4

  -
  The FSK-signal shape shall be within the allowed area. Amplitudes of spikes of maximum 5 µs up to 100 kHz are allowed.
  As owing to the baudrate and manchester coding the length of logical signal can change between 30,5 µs and 61 µs, both values shall be tested.
  w = 2 * π (1/61,02 µs)

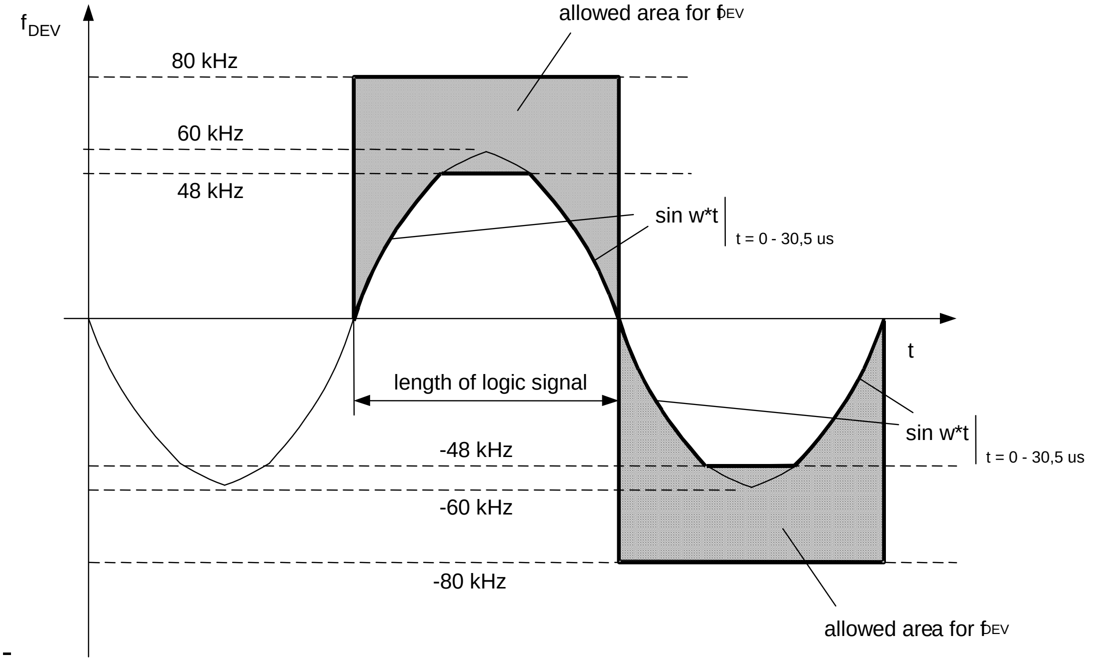

**Figure 2 - Allowed FSK-Deviation shape for Fc, F2 to F4**

- For devices using RF channels S1 and S2

  The FSK-signal shape shall be within the allowed area. Amplitudes of spikes of maximum 5 µs up to 80 kHz is allowed.
  As owing to the baudrate and manchester coding the length of logical signal can change between 61 µs and 122 µs, both values shall be tested.
  w = 2 * π (1/122 µs)

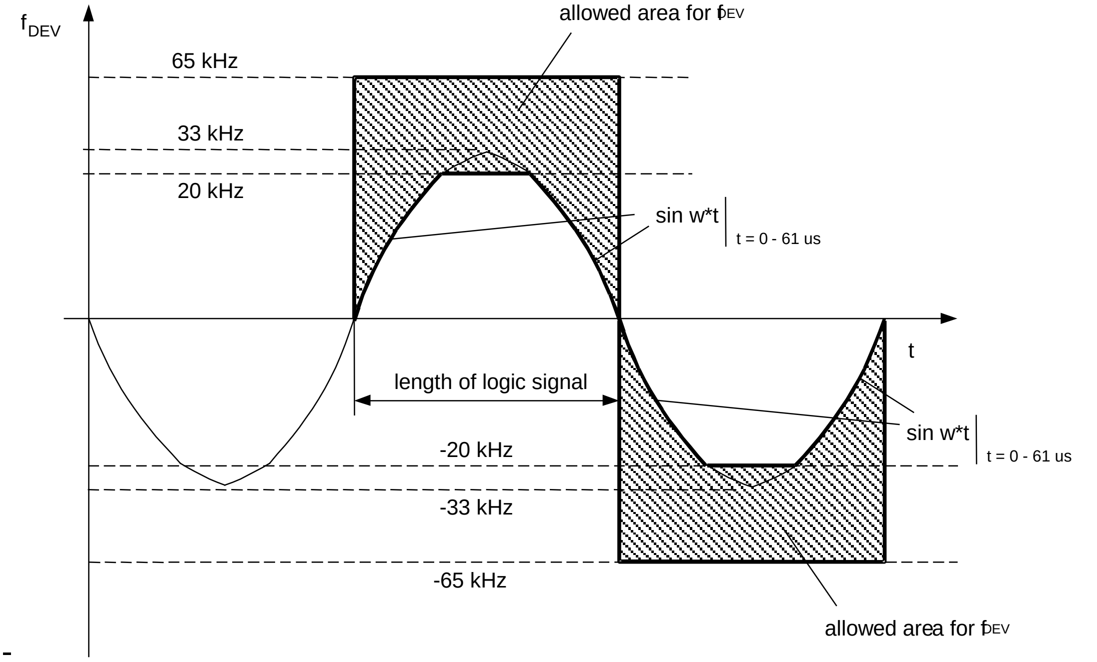

**Figure 3 - Allowed FSK-Deviation shape for S1 and S2**

### 3.1.7 Testing max allowed bandwitdh

- Test Setup:
  See modulation range test described in EN 300 220-1
- Testing:
  See modulation range test described in EN 300 220-1
- Requirements:
  See Table 1 limits.

  The modulation range including frequency drift over the whole temperature range shall be inside the designated RF channel bandwidth.

### 3.1.8 Testing of Tx Chip Rate

- Test Set-up:
  Ambient Temperature: 0 °C, 25 °C, 45 °C

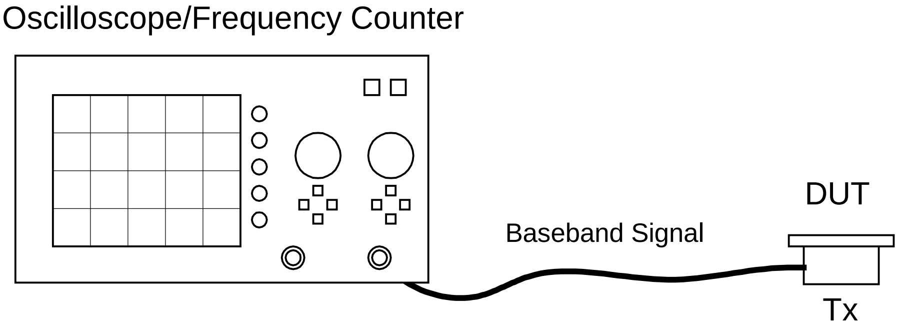

**Figure 4 - Example Test set up Tx chip rate / tolerance**

- Testing:
  An electrical access to baseband signal (digital chip stream) shall be provided and measured e.g. with an oscilloscope or frequency counter. In case the baseband signal is not accessible, a calculation demonstrating the compliance shall be provided.
- Requirements:
  See Table 1 limits.
  The manufacturer shall describe in a manufacturer test report how the measured tolerance is influenced by the frequency relevant components used (e.g. ageing, temperature, initial tolerance of e.g. the used crystal) do not cause an overstepping of the limits.

  KNX Physical Layer parameters shall be met on the entire product temperature range declared by the manufacturer in case it exceeds 0 °C – 45 °C range (e.g. : -10 °C to 70 °C for outdoor usage).

### 3.1.9 Testing of Tx Jitter per transition

- Test Set-up:

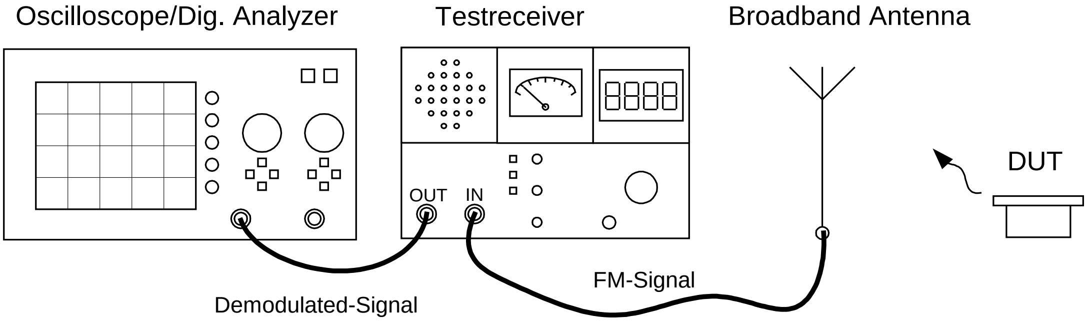

**Figure 5 - Example Test set up Tx Jitter per transition**

- Testing:
  Any FM-test receiver can be used with an appropriate accuracy regarding the demodulation of the FM-signal. The jitter generated by the FM-test receiver shall be ≤ 0,5 µs. The DUT shall transmit an RF-Telegram. The demodulated signal is subsequently evaluated by means of e.g. an oscilloscope or digital analyser.
- Requirements:

  See Table 1 limits.
  ***Fc, F2, F3 and F4***

  Owing to the baseband signal in the ideal case a stream of positive and negative pulses of 30,5 µs and 61,0 µs will be visible. Each pulse after the first preamble pulse [1] shall be tested and shall be in the following range.

| min. | typ. | max. | chips | comment |
|---|---|---|---|---|
| 25,5 µs | 30,5 µs | 35,5 µs | 1 | stream of logical “0” or “1” |
| 56,0 µs | 61,0 µs | 66,0 µs | 2 | change between logical “0” and “1” and vice versa |

**Figure 6 - Fc, F2, F3, F4 requirements Tx jitter per transition**

***S1 and S2***

Owing to the baseband signal in the ideal case a stream of positive and negative pulses of 61 µs and 122 µs will be visible. Each pulse after the first preamble pulse [2] shall be tested and shall be in the following range.

| min. | typ. | max. | chips | comment |
|---|---|---|---|---|
| 56,0 µs | 61,0 µs | 66 0µs | 1 | stream of logical “0” or “1” |
| 117,0 µs | 122,0 µs | 127.0 µs | 2 | change between logical “0” and “1” and vice versa |

**Figure 7 - S1 to S2 requirements Tx jitter per transition**

[1] It is allowed in that the first preamble pulse does not meet the above requirements due to the ramp up of power amplifiers.

[2] It is allowed in that the first preamble pulse does not meet the above requirements due to the ramp up of power amplifiers.

### 3.1.10 Testing of Rx Blocking Performance

- Test Setup:
  See EN 300 220-1
- Testing:
  See EN 300 220-1
- Requirements:
  See Table 1 limits.

### 3.1.11 Testing of Rx Frequency Tolerance

- Test Set-up:
  see 3.1.2 Bit Error Rate

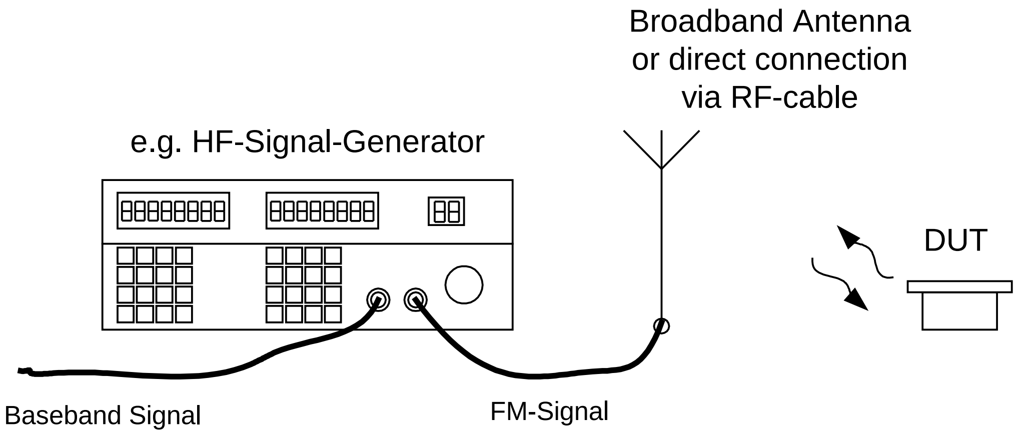

**Figure 8 - Example Test setup Rx Frequency Tolerance**

- Testing:
  The test shall be carried out in the same way as 3.1.2 (Bit Error Rate) with nominal Rx frequency (center frequency : Fc), Rx frequency +25 ppm and Rx frequency –25 ppm.
- Requirements:
  Bit Error Rate ≤ $10^{-4}$ [For radiated measurement, in the optimum antenna direction]

  See Table 1 limits.

  - For devices communicating with metering devices, the test shall also be performed at F1 = 868,300 MHz ± 60 ppm. [Not applicable to KNX RF AU]

  - For devices with RF slow channel feature, the test shall also be perfomed at Rx frequency S1 or S2. See limits in Table 1.

### 3.1.12 Testing of Rx Chip Rate Tolerance

- Test Set-up:
  Test Set-up regarding bit error rate see 3.1.2. at 25 °C

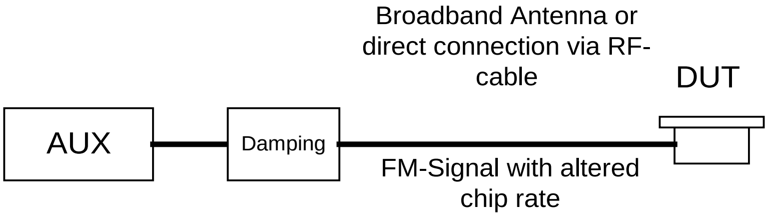

**Figure 9 - Example Test Setup Rx Chip Rate Tolerance**

- Testing:
  The test shall be carried out in the same way as 3.1.2 with an AUX device generating RF Telegrams with the nominal Tx chip rate, Tx chip rate +2 % and Tx chip rate –2 %.
- Requirements:
  Bit Error Rate ≤ $10^{-4}$

  For devices with Fast and Slow features, the test shall be performed at both datarates.

### 3.1.13 Testing of Rx radiated sensitivity

- Test Set-up:
  Test Set-up regarding bit error rate see 3.1.2. at 25 °C

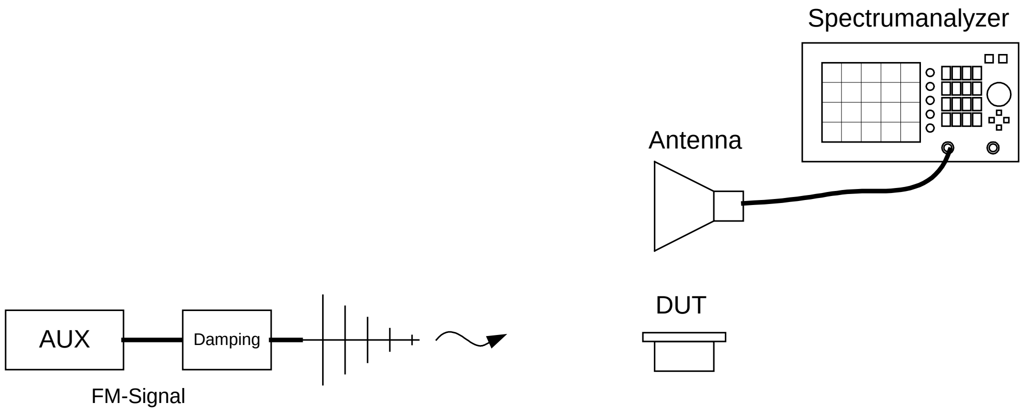

**Figure 10 - Example Test Setup Rx radiated Sensitivity**

- Testing:
  The test shall be carried out in the same way as 3.1.2 with an AUX device able to generate nominal RF Telegrams. The radiated signal shall be damped down to a level where the bit error rate is just achieved. Subsequently, the radiated signal level is measured with e.g. an antenna and spectrum analyser.
- Requirements:
  The radiated sensitivity level shall be below Table 1 limits in the optimum antenna direction.

## 3.2 Data Link Layer Tests (Physical Part)

### 3.2.1 Transmitter Tests

To perform these transmitter tests, an accurate measurement unit (with demodulator) or a calibrated receiver shall be used. Tests can be performed on only one single frequency.

### 3.2.1.1 Specific parameters

3.2.1.1.1 KNX RF Ready

| **Test** | **Test-Description** | **Expected Behaviour** |
|---|---|---|
| Preamble (Ready) | Count number of preamble bits | 76 ≤ number of preamble chip-sequences “01” shall be ≤ 81<br>Typical value = 79 |

#### 3.2.1.1.2 KNX RF BiBat and BiBat 2

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Preamble<br>(Bibat, Bibat2) | Count number of preamble bits | Number of preamble chip-sequences "01" ≥ 15 and not more than 32. |

#### 3.2.1.1.3 KNX RF Multi

Tests shall be done in relation with the device capability (fast only, slow only or fast & slow) and with the suitable data rate.

| Test for F1, F2 and F3 | Test-Description | Expected Behaviour |
|---|---|---|
| Preamble (Fast) | Count number of preamble bits | 242 ≤ Number of preamble chip-sequences "01" shall be ≤ 253<br>Typical value = 247 |
| Medium access with one RF channel blocked (Fast) | The default call channel shall be F1<br>Put a permanent carrier in F1 at 0dBm<br>Send Telegrams | DUT shall successfully transmit Frames in RF channel F2 without unexpected delay |
| Medium access with two RF channels blocked (Fast)<br>[only applicable to devices with F3 feature] | The default call channel shall be F1<br>Put two permanent carriers in F1 and in F2 at 0dBm<br>Send Telegrams | DUT shall successfully transmit Frames in RF channel F3 without unexpected delay |
| Medium access with an ongoing metering frame on F1 | The call channel shall be F1<br>Send a long KNX dummy frame of more than 600ms at F1<br>Send Telegrams | DUT shall successfully transmit Frames in RF channel F2 after 500ms |

| Test for S1 and S2 | Test-Description | Expected Behaviour |
|---|---|---|
| Preamble (Slow) | Count number of preamble bits | 4106 ≤ Number of preamble chip-sequences "01" shall be ≤ 4116<br>Typical value = 4111 according Volume 3/2/5 |
| Medium access with one RF channel blocked (Slow) | The default call channel shall be S1<br>Put a permanent carrier in S1 at 0dBm<br>Send Telegrams | DUT shall successfully transmit Frames in RF channel S2 without unexpected delay |

#### 3.2.1.1.4 Fast Ack service

This clause applies if the Fast Ack service is available in the device. Tests shall be done in relation with the device capability (fast only, slow only or fast & slow).

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Postamble with End of Ack | Inspect signal with the complete postamble sequence for Fast Ack | Number of postamble chips (from start to end) : 320 |

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Fast Ack Sync Word | Inspect signal with sync word for Fast Ack | Fast Ask Postamble sync word shall be sent.<br>(chip sequence 011011011011) |
| End Of Ack field | Send a Frame requesting 2 Fast Acks | DUT shall transmit the number of Fast Ack expected = “10” in the field EOA |
| Fast Ack End | Inspect signal with end sequence for Fast Ack | Number of End chip sequence: “01”: 4 |
| Fast Ack filling | Send a Frame requesting at least 3 Fast Ack. Send Fast Acks in answer but with one missing. | DUT shall transmit the filling sequence during the missing Fast Ack window : 8 to 10 times a sequence of “0011” chips |

#### 3.2.1.2 Other common parameters

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Postamble<br>[Not applicable when Fast Ack service is enabled] | Count number of postamble bits | Number of postamble chips = [2...8] |
| Manchester-Violation | Inspect signal after preamble | Chip-sequence „000111“ shall be sent after preamble |
| Syncword | Inspect signal after manchester violation | The correct syncword shall be sent: (chip sequence "011010010110") |

*Note: For slow RF channels, the data rate is divided by 2 compared to Fast RF channels.*

### 3.2.2 Receiver – Tests

To perform these receiver tests use a waveform generator or a calibrated RF device with modified Physical Layer.

#### 3.2.2.1 KNX RF Ready

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Preamble for KNX RF 1.1 devices | Send Telegrams with preamble chip-sequence number =15, as normally generated by a KNX RF 1.1 device | DUT (Device Under Test) shall successfully receive Frames with 15 preamble-chip-sequences “01” |
| Preamble for Ready devices | Send Telegrams with preamble chip-sequence number =79, as normally generated by any KNX RF device other than KNX RF 1.1 | DUT (Device Under Test) shall successfully receive Frames with 79 preamble-chip-sequences “01” |

#### 3.2.2.2 KNX RF BiBat, Bibat2

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Preamble | Send Telegrams with different amount of preamble-bits | DUT shall successfully receive Frames with ≥ 15 preamble-chip-sequences “01” |

### 3.2.2.3 KNX RF Multi

One single receiver with several behaviour KNX RF Multi Fast, KNX RF Multi Slow and KNX RF Fast & Slow might be used for these tests or 4 different devices.

If the NPRM (no permanent receive mode) is used on the DUT, the tests have to be performed with NPRM and PRM (permanent receive mode) behaviour if not specified.

| **Tests for F1, F2, F3** | **Test-Description** | **Expected Behaviour** |
|---|---|---|
| Multi Fast preamble | Send Telegrams with preamble chip-sequence number =247 successively on F1, F2 (and F3 if supported) | DUT (Device Under Test) shall successfully receive Frames with 247 preamble-chip-sequences “01” |
| Ready preamble | Use the device with PRM behaviour<br>Send Telegrams with preamble chip-sequence number =79 on F1 | DUT (Device Under Test) shall successfully receive Frames with 79 preamble-chip-sequences “01” |
| Fast Ack postamble (Fast) | Send Telegram with postamble sequence for Fast Ack | DUT shall successfully receive Frames with postamble sequence for Fast Ack |
| Medium access with one RF channel blocked (Fast) | Put a permanent carrier in F1<br>Send Telegrams on F2 | DUT shall successfully receive Frames in RF channel F2 without unexpected delay |
| Medium access with two RF channels blocked (Fast)<br>[only applicable to devices with F3 feature] | Put two permanent carriers in F1 an F2<br>Send Telegrams on F3 | DUT shall successfully receive Frames in RF channel F3 without unexpected delay |
| Medium access with a metering KNX frame (fast) | Send a long KNX Frame of more than 600ms at F1<br>At the same time send Telegrams | DUT shall successfully receive Frames in RF channel F2 after 500ms |

| **Tests for S1, S2** | **Test-Description** | **Expected Behaviour** |
|---|---|---|
| Multi Slow preamble | Send Telegrams with preamble chip-sequence number =4111 | DUT (Device Under Test) shall successfully receive Frames with 4111 preamble-chip-sequences “01” |
| Multi Fast preamble | Use the device with PRM behaviour<br>Send Telegrams with preamble chip-sequence number =247 on F1, F2 or F3 | DUT (Device Under Test) shall successfully receive Frames with 247 preamble-chip-sequences “01” |
| | | |
| Fast Ack postamble (Slow) | Send Telegram with postamble sequence for Fast Ack | DUT shall successfully receive Frames with postamble sequence for Fast Ack |
| Medium access with one RF channel blocked (Slow) | Put a permanent carrier in S1<br>Send Telegrams on S2 | DUT shall successfully receive Frames in RF channel S2 without unexpected delay |

#### 3.2.2.3.1 Fast Ack service

This paragraph applies if the Fast Ack Service is available in the device.

This service is independent of the used RF channel Fast or Slow.

| Tests | Test-Description | Expected Behaviour |
|---|---|---|
| Fast Ack preamble | Inspect preamble of Fast Ack | A preamble of 18 chip sequence "01" shall be sent |
| Fast Ack synchro word | Inspect signal of Fast Ack after preamble | The correct syncword shall be sent:<br>(chip sequence "011010010110") |
| Fast Ack KNX CTRL field | Sent Fast Ack with different KNX Ctrl values [bit7 to bit4] | DUT shall reject Fast Ack KNX CTRL values <> 0001eeee<br>eeee = EFF field see Vol 03_03_02 Data Link Layer General |

### 3.2.2.4 Other common parameters

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Postamble<br>[Not applicable when Fast Ack service is enabled] | Send Telegrams with different amount of postamble-bits | DUT shall successfully receive Frames with [2..8] postamble-chips |

## 3.3 Data Link Layer Tests (Data Part)

The data part of the Data Link Layer shall be tested with a device capable of receiving raw FT3 Frames.

### 3.3.1 Transmitter Tests for end devices

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| FT3 Length | count total numbers of user octets beginning with the C-Field but excluding the CRC's | The Length shall be correct |
| FT3 C-Byte | Inspect C-Byte in Frame | FT3 C-Byte = 44h |
| ESC-Byte | Inspect ESC-Byte in Frame | ESC-Byte = FFh |
| RF-Info-Byte | inspect RF-Info-Byte in Frame | Bid: 02h (in case battery good)<br>Uni: 03h (in case battery good) |
| Serial Number | inspect Serial Number in Frame | Pre-configured SerNr |
| KNX Ctrl-Byte | inspect KNX-Ctrl-Byte in Frame | KNX-Ctrl = 00h for standard<br>KNX-Ctrl = 04h .. 07h for LTE (EFF=01xxb for LTE)<br>KNX ctrl = 80h or 90h for Multi<br>other values of the KNX Ctrl field are reserved for RF extensions, e.g. BiBat |
| Indiv. Address | inspect Indiv. Address in Frame | Pre-configured IndAdr |
| CRC | Send Frames with DUT and check the CRC value on receiver or on demodulated signal | CRC correctly calculated (FT3-standard) |
| Telegram length | Send Telegrams of different lengths, check positions of | CRC shall appear on expected byte |

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| and –assembly | CRC (min. length – max. length) | positions |
| Medium access time after reception | Device A continuously sends Telegrams with shorter gap than allowed (<15 ms). Then trigger Telegram on DUT. Finally, stop the continuous transmission of device A. | DUT shall keep a total medium access time and wait until device A stops transmission |
| KNX RF Ready<br>Medium access time on transmission | Trigger several concurrent Telegrams on DUT at the same time (e.g. several transmit requests of communication objects) and measure interFrame gap. | DUT shall keep a total medium access time of :<br>Bidirectional: 15 ms minimum – 30 ms maximum between successive Telegrams<br>Unidirectional: 150 ms minimum – 160 ms maximum between successive Telegrams |
| KNX RF Multi<br>Fast channels<br>Medium access time on transmission | Trigger several concurrent Telegrams on DUT at the same time (e.g. several transmit requests of communication objects) and measure interFrame gap. | DUT shall keep a total medium access time of :<br>30 ms minimum – 50 ms maximum between successive Telegrams<br>Note: LBT time is not included. |
| KNX RF Multi<br>Slow channels<br>Medium access time on transmission | Trigger several concurrent Telegrams on DUT at the same time (e.g. several transmit requests of communication objects) and measure interFrame gap. | DUT shall keep a total medium access time of :<br>60 ms minimum – 100 ms maximum between successive Telegrams<br>Note: LBT time not included. |
| LFN | Trigger several Telegrams on DUT and inspect the LFN values | LFN shall loop over values 0..7 |
| Domain Bit | a) Send A_IndAdrRead to the DUT and inspect the A_IndAdrResponse<br>b) Send A_GroupValueWrite | a) Domain-bit shall be set and the SN fields contain the domain address<br>b) Domain-bit shall not be set, the SN fields shall contain the serial number |

## 3.3.2 Receiver Tests for end devices

### 3.3.2.1 KNX RF Ready

To perform these receiver tests use a transmitter device with modified Data Link Layer

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| FT3 C-Byte | Send Telegrams with different FT3 C-Bytes | DUT shall ignore values <> 44h |
| KNX CTRL-Byte | Send Telegrams with different KNX-CTRL-Bytes | DUT shall ignore values <> 00h<br>(standard), <> 04h .. 07h for LTE (EFF=01xxb for LTE) [other values of the KNX CTRL field are reserved for RF extensions, e.g. BiBat] |

### 3.3.2.2 KNX RF multi

To perform these receiver tests use a transmitter device with modified Data Link Layer.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| FT3 C-Byte | Send Telegrams with different FT3 C-Byte | DUT shall ignore values <> 44h and <>00h |
| KNX CTRL-Byte<br>RF Ready Asynchronous dataFrame L data | Send Telegrams with KNX CTRL = 00h | DUT shall successfully receive Frames and not discard them.<br>KNX-CTRL = 04h .. 07h for LTE (EFF=01xxb for LTE) |
| KNX CTRL-Byte<br>RF multi Asynchronous dataFrame L data | Send Telegrams with KNX CTRL= 10000000b = 80h | DUT shall successfully receive Frames and not discard them.<br>KNX-CTRL = 04h .. 07h for LTE (EFF=01xxb for LTE) |
| KNX CTRL-Byte<br>RF multi Asynchronous dataFrame L data with Fast Ack | Send Telegrams with KNX CTRL = 10010000b = 90h | DUT shall successfully receive Frames and not discard them.<br>KNX-CTRL = 04h .. 07h for LTE (EFF=01xxb for LTE) |
| KNX CTRL-Byte | Send Telegrams with different KNX-CTRL-Bytes | DUT shall ignore values <> 00h, 80h or 90h<br>(standard), <> 04h .. 07h for LTE (EFF=01xxb for LTE) [other values of the KNX CTRL field are reserved for RF extensions, e.g. BiBat] |

### 3.3.2.3 Other common tests

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Undefined Codings | Send Telegrams with other coding than FT3 (e.g. NRZ) | DUT shall ignore Telegrams with incorrect coding |
| CRC | Send Telegrams with invalid CRC values and check if receiver ignores them. For this, use different CRC positions. | DUT shall ignore Telegrams with incorrect CRC (FT3-standard) |
| Telegram length | Send Telegrams of different lengths and check if receiver is able to receive them. | DUT shall successfully receive all valid Telegram lengths and ignore Telegrams with invalid length |
| ESC-Byte | Send Telegrams with different FT3 ESC-Bytes | DUT shall ignore non-KNX values (<> FFh) |
| LFN | Use transmitter that doesn’t increase LFN | DUT shall only accept first transmission. Subsequent transmissons with wrong LFN shall be ignored. |
| Domain-Bit | a) Send A_PropertyRead to DUT with Domain-Bit not set<br>b) Send A_GroupValueWrite to DUT with Domain-Bit set | a) DUT shall ignore Telegram<br>b) DUT shall ignore Telegram |
| RF Info-Byte | Send Telegrams with different RF-Info-Bytes | Receivers shall accept all values, as this byte is only used for information purposes. |

## 3.4 Set-up considerations for Testing System-Software

### 3.4.1 Test Set-up:

- A Personal Computer running EITT4 or higher), connected to the bus by an RF interface with the serial number "Master" and the Individual Address AFFEh,
- the BDUT (Bus Device Under Test) with a test-specific application software. The BDUT shall have the serial number "_BDUT_" and the Individual Address 1000h.

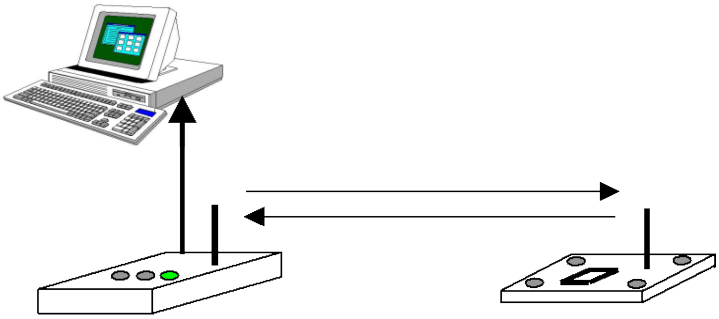

**Figure 11 - Test Set-up**

Moreover, the test application may provide means to:

- delete all links in the BDUT (i.e. delete the association table). This can e.g. be ensured via a user function property,
- trigger the sensor channel in the BDUT to send the link sequence. This can e.g. be ensured via a user function property,
- activate the programming mode of an actuator channel of the BDUT. This can e.g. be ensured via a user function property.

### 3.4.2 Format of Telegram sequences:

The Telegram sequences depicted in the following tests are reduced to the KNX related fields, i.e. the FT3-specific fields (Length, C-Byte, ESC-Byte and CRC's) are not included:

| RF-INFO<br>(1 octet) | Serial Number / Domain ID<br>(6 octets) | CTRL<br>(1 octet) | SRC<br>(2 octets) | DST<br>(2 octets) | L/NPCI<br>(1 octet) | TPCI<br>(1 octet) | APCI<br>(1 octet) | Data....<br>(x octets) |
|---|---|---|---|---|---|---|---|---|

Since the RF-Info has an informative nature, the values of this field shall not influence the behaviour of the BDUT. Hence , the value 00h ("Battery weak, "Bidirectional device") has been chosen for Telegrams to be transmitted. Any other value can be chosen instead.

As the RF-Info field for received Telegrams moreover includes unpredictable values like the "received signal strength", the values used in the underneath sequences are just examples. Any other value shall be accepted by the BDUT as well.

## 3.5 Transport Layer tests

The following tests should be used if the implemented KNX RF Transport Layer only supports connectionless services (profiles). If the device supports a connection-oriented Transport Layer, the corresponding tests of the KNX core system shall be carried out (see [05]).

The following test ensures that

- in sending direction the TL-service is correctly selected, and
- in receiving direction the correct TL-service is checked and split into the different AL-services.

- All the following tests can be performed on a native RF Ready software stack or a RF Multi software stack compiled in Ready type.

### 3.5.1 System-Broadcast with different TL Services (UDT, NDT, UCD, NCD)

Test description: seven IndividualAdrSernrRead Telegrams (SerNr = "_BDUT_") are sent. The BDUT shall accept all Telegrams with the TL – Service "Unnumbered Data" (UDT) and reject arbitrary sequence numbers (1..l5). The BDUT shall answer with IndividualAdrSernrResponse Telegrams.

Every other sent TL-service (NDT, UCD, NCD) shall be ignored by the BDUT. The response Telegrams of the BDUT shall always be sent as TL – Service UDT with SequenceNr = 0000.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=0* |
| ► 32 5F424455545F 00 1000 0000 E0 03 DD 5F424455545F 00000000 | *answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 07 DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=1* |
| ►---------- no *answer from BDUT* | |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 3F DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=15* |
| ►---------- no *answer from BDUT* | |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 43 DC 5F424455545F | *IASerNoRead (_BDUT_), NDT, S=0* |
| ► ---------- | *no answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 83 DC 5F424455545F | *IASerNoRead (_BDUT_), UCD S=0* |
| ► ---------- | *no answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 C3 DC 5F424455545F | *IASerNoRead (_BDUT_), NCD S=0* |
| ► ---------- | *no answer from BDUT* |

### 3.5.2 Domain-Broadcast with different TL Services (UDT, NDT, UCD, NCD)

Test description: seven IndividualAdrSernrRead Telegrams (SerNr = "_BDUT_") are sent. The BDUT shall accept all Telegrams with the TL – Service "Unnumbered Data" (UDT) and reject arbitrary sequence numbers (1..l5) The BDUT shall answer with IndividualAdrSernrResponse Telegrams. Every other sent TL – service (NDT, UCD, NCD) shall be ignored by the BDUT. The response Telegrams of the BDUT shall always be sent as TL – Service UDT with SequenceNr = 0000.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E1 03 DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=0* |
| ► 32 4D4153544552 00 1000 0000 E7 03 DD 5F424455545F 00000000 | *answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E1 07 DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=1* |
| ►---------- no *answer from BDUT* | |
| ◄ 00 4D4153544552 00 AFFE 0000 E1 3F DC 5F424455545F | *IASerNoRead (_BDUT_), UDT, S=15* |
| ►---------- no *answer from BDUT* | |
| ◄ 00 4D4153544552 00 AFFE 0000 E1 43 DC 5F424455545F | *IASerNoRead (_BDUT_), NDT, S=0* |
| ► ---------- | *no answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E1 83 DC 5F424455545F | *IASerNoRead (_BDUT_), UCD, S=0* |
| ► ---------- | *no answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E1 C3 DC 5F424455545F | *IASerNoRead (_BDUT_), NCD, S=0* |
| ► ---------- | *no answer from BDUT* |

### 3.5.3 Test for T_Disconnect-Response on individual connection-oriented requests.

When receiving an individual request connection oriented, the BDUT shall answer with a T_Disconnect.req Service to close the connection immediately.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 00 | *DDRead Typ 0* |
| ► 32 4D4153544552 00 1000 AFFE 6D 03 40 20 10 | *answer from BDUT* |

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 43 00 | *DDRead Typ 0 connection oriented* |
| ► 32 4D4153544552 00 1000 AFFE 6F 81 | *T_Disconnect* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 83 00 | *DDT0 (TPCI <> UDT)* |
| ► 32 4D4153544552 00 1000 AFFE 61 81 | *T_Disconnect* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 C3 00 | *DDT0 (TPCI <> UDT)* |
| ► 32 4D4153544552 00 1000 AFFE 63 81 | *T_Disconnect* |

## 3.6 Management Layer tests

### 3.6.1 DomainAddressSernr-Read/-Write/-Response

Purpose: DomainAdressSerNr services are sent as system broadcast whereby the addressed serial number is included in the data part of the Telegram. The BDUT shall only react if the received serial number matches its own.

#### 3.6.1.1 DoAddrSerNoRead

Check if BDUT does not send a response if it receives a DomAdrSernrRead service with the wrong SerNo.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EC 112233445566 | *DoAddrSerNoRead (wrong SerNr)* |
| ► ----------------- | *no answer from BDUT* |

Check if BDUT does not send a response if it receives a DomAdrSernrResponse service with the wrong SerNo.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 ED 112233445566 | *DoAddrSerNoResponse (wrong SerNr)* |
| ► ----------------- | *no answer from BDUT* |

Check if BDUT sends the correct response if it receives a DomAdrSernrRead service containing the SerNo of the BDUT.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EC 5F424455545F | *DoAddrSerNoRead (correct SerNr)* |
| ► 32 5F424455545F 00 1000 0000 E8 03 ED 5F424455545F 4D4153544552 | *expected response* |

Check if BDUT does not send a response if it receives a DomAdrSernrResponse containing the SerNo of the BDUT.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 ED 5F424455545F | *DoAddrSerNoResponse (correct SerNr)* |
| ► ----------------- | *no answer from BDUT* |

#### 3.6.1.2 DoAddrSerNoWrite

Check if BDUT changes its domain address correctly if it receives a DoAddrSerNoWrite service containing the serial number of the BDUT.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EE 5F424455545F 434845464645 | *DoAddrSerNoWrite ok* |
| ► ----------------- | *no answer from BDUT* |
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EC 5F424455545F | *DoAddrSerNoRead (correct SerNr)* |
| ► 32 5F424455545F 00 1000 0000 EA 03 ED 5F424455545F 434845464645 | *expected response* |

Check if BDUT does not change its domain address if it receives a DoAddrSerNoWrite service with the wrong serial number.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EE 112233445566 4D4153544552 | *DoAddrSerNoWrite (wrong SerNr)* |
| ► ----------------- | *no answer from BDUT* |

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 0000 E0 03 EC  5F424455545F | *DoAddrSerNoRead (previous SerNr)* |
| ► 32 5F424455545F 00 1000 0000 EC 03 ED  5F424455545F 434845464645 | *expected response* |

### 3.6.2 DeviceDescriptor (type 0 / type 2)

The Device-Descriptor type 0 returns the Mask version, type 2 returns the Easy channel information.

Check if BDUT sends the correct response to a DDRead type 0:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 00 | *DDRead type 0* |
| ► 32 4D4153544552 00 1000 AFFE 63 03 40  20 10 | *type 0 response* |

Check if BDUT sends the correct error response to a DDRead type 1:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 01 | *DDRead type 1* |
| ► 32 4D4153544552 00 1000 AFFE 65 03 7F | *type 1 response error* |

Check if BDUT sends the correct response to a DDRead type 2:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 02 | *DDRead type 2* |
| ► 32 4D4153544552 00 1000 AFFE 67 03 42 0001ABCD003F0123020104010000 | *type 2 response* |

Check if BDUT sends the correct error response to a DDRead type 3:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 03 | *DDRead Type 3* |
| ► 32 4D4153544552 00 1000 AFFE 69 03 7F | *type 3 response error* |

### 3.6.3 FunctionPropertyCommand

Check that the BDUT executes a FuncPrCmd on the function ObjectValue correctly

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E100 01 | *FuncPrCmd ObjValue: E100 to 0x01* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 3E 00 | *Response ok (=00h)* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E100 | *FuncPrStRead ObjValue E100* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9  00 3E 00 01 | *Response ok, value = 0x01* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E100 00 | *FuncPrCmd ObjValue: E100 to 00h* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9  00 3E 00 | *response ok (=00h)* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E100 | *FuncPrStRead ObjValue E100* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9  00 3E 00 00 | *response ok, value = 00h* |

Check that BDUT reacts on a PropertyValueRead on a property, which is of type PDT_Function, with an error response  (start index in response may have any value).

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 D5  00 3E 10 01 | *PropertyRead (not allowed)* |
| ► 32 4D4153544552 00 1000 AFFE 6B 03 D6  00 3E 00 xx | *response error (NoE = 00h)* |

Check that BDUT reacts on a PropertyValueWrite on a property, which is of type PDT_Function, with an error response (start index in response may have any value).

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 03 D7  00 3E 10 01 | *PropertyWrite (not allowed)* |
| ► 32 4D4153544552 00 1000 AFFE 6D 03 D6  00 3E 00 xx | *response error (NoE = 00h)* |

Check that BDUT reacts on a FuncPrCmd on a property, which is not of type PDT_Function, with an error response.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 0B 10 01 | *FuncPrCmd (invalid)* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9  00 0B | *response error (no data)* |

Check that BDUT reacts on a FuncPrStRead on a property, which is not of type PDT_Function, with an error response.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 0B 10 01 | *FuncPrStRead (invalid)* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 0B | *response error (no data)* |

# 4 RF Retransmitters

Retransmitters are used to expand the range of RF transmitters. Incoming RF Telegrams are repeated transparently (with a shorter interFrame time than end devices have), i.e. the original KNX Telegram remains unchanged, except the routing counter, which shall be decremented by Retransmitters. (Further information is given in the RF System specification [01]).

### 4.1.1 Physical Layer

Requirements for the Physical Layer of Retransmitters are identical to those for end devices. (see 3.1).

### 4.1.2 Data Link Layer

#### 4.1.2.1 KNX RF Ready and KNX RF BiBat

The Data Link Layer part of Retransmitters shall be tested using a transmitter device and a tool capable of receiving raw FT3 Frames. The expected behaviour for Retransmitters is identical to end devices (see 3.2), with the following exceptions.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Medium access time after reception | Use a device continuously sending Telegrams with an interFrame gap <5ms. | The Retransmitter shall keep medium access time [5..15 ms] and wait until the device stops transmission. |
| Medium access time random portion | Use a bidirectional device continuously sending Telegrams. | Medium access time between original and repeated transmission should vary between [5ms – 15 ms] |

#### 4.1.2.2 KNX RF Multi

##### 4.1.2.2.1 Telegrams retransmission (except ACKREP)

The Data Link Layer part of Retransmitters shall be tested using a transmitter device and a tool capable of receiving Frames. The expected behaviour for Retransmitter is identical to end devices with the following exceptions:

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| InterFrame-time after reception (Ready frame) | Use a device continuously sending Ready Telegrams with an interFrame gap < 5ms | The Retransmitter shall keep interFrame time [5.. 15ms[ and wait until the device stops transmission. |
| InterFrame-time after reception (Multi frame Fast) | Use a device continuously sending Multi Fast Telegrams with an interFrame gap < 5ms | The Retransmitter shall keep interFrame time [5.. 10ms[ and wait until the device stops transmission. |
| InterFrame-time after reception (Multi frame Slow) | Use a device continuously sending Multi Slow Telegrams with an interFrame gap < 10ms | The Retransmitter shall keep interFrame time [10.. 20ms[ and wait until the device stops transmission. |
| InterFrame-time random portion (Ready frame frame) | Use a bidirectional device sending Ready Telegrams | InterFrame time between original and repeated transmission should vary between [5..15ms] . |
| InterFrame-time random portion (Multi frame Fast) | Use a bidirectional device sending Multi Fast Telegrams | InterFrame time between original and repeated transmission should vary between [5..10ms] . |
| InterFrame-time random portion | Use a bidirectional device sending Multi Slow Telegrams | InterFrame time between original and repeated transmission should |

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| (Multi frame Slow) | | vary between [10..20ms] . |
| RF-Info-Byte (signal strength).<br>(Ready frame, Multi frame Fast and Multi frame Slow) | Use a device sending Telegrams with different Retransmitter-signal-strength values in the RF-Info-Byte. Hereby try to lower the received signal strength for the Retransmitter. In all cases, inspect RF-Info-Byte of original and repeated Frame. | If supported, the Retransmitter should insert its measured signal strength into the RF-Info-Byte, but only if the received value is 0 (normal sender) or the measured value is lower than the one from the incoming Telegram (from other Retransmiiters). |
| Serial Number, Individual Address.<br>(Ready frame, Multi frame Fast and Multi frame Slow) | Send different valid KNX RF Telegrams and compare them with the repeated ones. | The Serial/Domain address and Individual address shall be unchanged by the Retransmitter. |
| History buffer.<br>(Ready frame, Multi frame Fast and Multi frame Slow) | Use 2 Retransmitters and a device continuously sending Telegrams. | Each Telegram shall be transmitted only once per Retransmitter. (3 transmissions: original, Retransmitter 1, Retransmitter 2) |

#### 4.1.2.2.2 ACKREP frame generation in retransmitter

These tests use a KNX RF Multi retransmitter (DUT).

The goal is to check the ACKREP frame that is generated by a retransmitter, after the retransmission of a frame containing a Fast Ack request.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| C-Byte | Transmit a frame with a Fast Ack request, and inspect C-Byte in the ACKREP frame. | C-Byte = 00h |
| KNX Ctrl-Byte | Transmit a frame with a Fast Ack request, and inspect KNX Ctrl-Byte in the ACKREP frame. | KNX Ctrl-Byte = A0h |
| RC | Transmit a frame with a Fast Ack request, and inspect RC in the ACKREP frame.<br>(For Multi, RC <= 2) | Verify for different generated values of RC that :<br>RC(generated AckRep frame) =<br>2 – RC(received frame request) |
| Identical fields (Esc, RF-info, SN/DoA, SA, DA, LPCI (except RC), TPCI, Seq. number, APCI) | Transmit a frame with a Fast Ack request, and compare the fields from the ACKREP frame with the initial frame sent by the sensor. | The fields shall be identical. |
| 64 Fast Ack | Transmit a frame with a 64 Fast Ack request, and check if the ACKREP frame is also generated with 64 Fast Ack status bytes. | An ACKREP frame with 64 bytes of data. |

#### 4.1.2.2.3 Medium access time for ACKREP frame

To run the next tests, a retransmitter (DUT), that is able to transmit an ACKREP on user action, shall be used.

The goal is to check the medium access times for an ACKREP frame generated by a retransmitter.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Total medium access time after reception (Fast) | Use an auxiliary device continuously sending Telegrams with an interframe gap < 10 ms , and request the transmission of an ACKREP on the DUT. | The DUT shall keep medium access time [10..15ms] after the auxiliary device stops transmission. |
| Random medium access time (Fast) | Trigger several concurrent ACKREP transmissions on DUT at the same time. | DUT shall keep a total medium access time of 10ms minimum – 15ms maximum between successive Telegrams. |

#### 4.1.2.2.4 ACKREP management in sender

In the next test, the DUT is a KNX RF Multi device (linked with two receivers) that transmits frames with Fast Ack request. A KNX RF Multi retransmitter shall be used to transmit the ACKREP frame, and the 2 receivers shall be configured to answer to the Fast Ack request. They shall also be out of RF range of the sender.

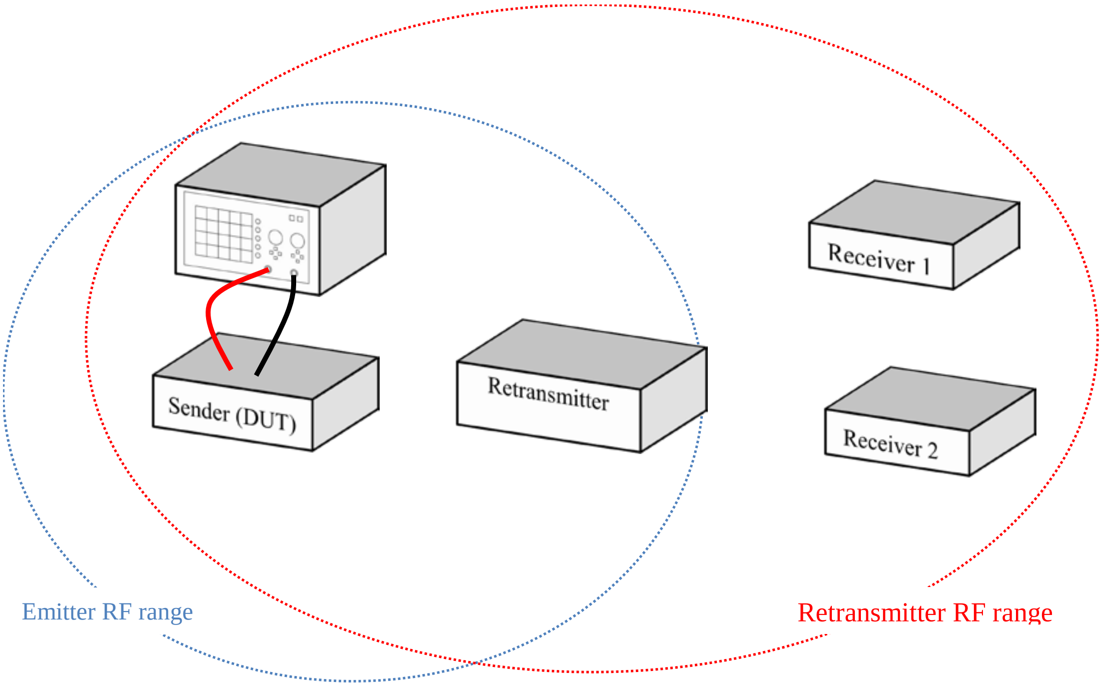

**Figure 12 Test setup**

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| Echo Frame time-out (fast) | The DUT transmits a frame with Fast Ack request, and no retransmitter repeats it. The two linked devices are switched off, so that they do not send Fast Ack when the frame is transmitted. | The DUT shall wait at least 75 ms before sending the frame again. |
| Echo detection (fast) | The DUT transmits a frame with Fast Ack request, and the retransmitter repeats it. The two linked devices are switched off, so that they do not send Fast Ack when the frame is transmitted. | The DUT shall wait at least 1000 ms before sending the frame again. |
| Echo Frame time-out (slow) | The DUT transmits a frame with Fast Ack request, and no retransmitter repeats it. The two linked devices are switched off, so that they do not send Fast Ack when the frame is transmitted. | The DUT shall wait at least 600 ms before sending the frame again. |
| Echo detection (slow) | The DUT transmits a frame with Fast Ack request, and the retransmitter repeats it. The two linked devices are switched off, so that they do not send Fast Ack when the frame is transmitted. | The DUT shall wait at least 4000 ms before sending the frame again. |

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| ACKREP handling (fast) | The DUT shall transmit a frame with Fast Ack request to the linked receivers that are out of direct RF range. The repeater shall repeat this frame, and send back the ACKREP to the DUT. | Fast Ack status is correctly handled by the DUT (DUT does not resend), even though it was generated by the retransmitter. |
| ACKREP handling | The DUT shall transmit a frame with Fast Ack | Fast Ack status is correctly handled |

| | | |
|---|---|---|
| (slow) | request to the linked receivers that are out of direct RF range. The repeater shall repeat this frame, and send back the ACKREP to the DUT. | by the DUT (DUT does not resend), even though it was generated by the retransmitter. |

#### 4.1.2.2.5 ACKREP after reception of Fast Ack

In the next test, the DUT is a KNX RF Multi device (linked with two receivers) that transmits frames with Fast Ack request. A KNX RF Multi retransmitter shall be used to transmit the ACKREP frame, and the 2 receivers shall be configured to answer to the Fast Ack request. They shall also be in the RF range of the sender.

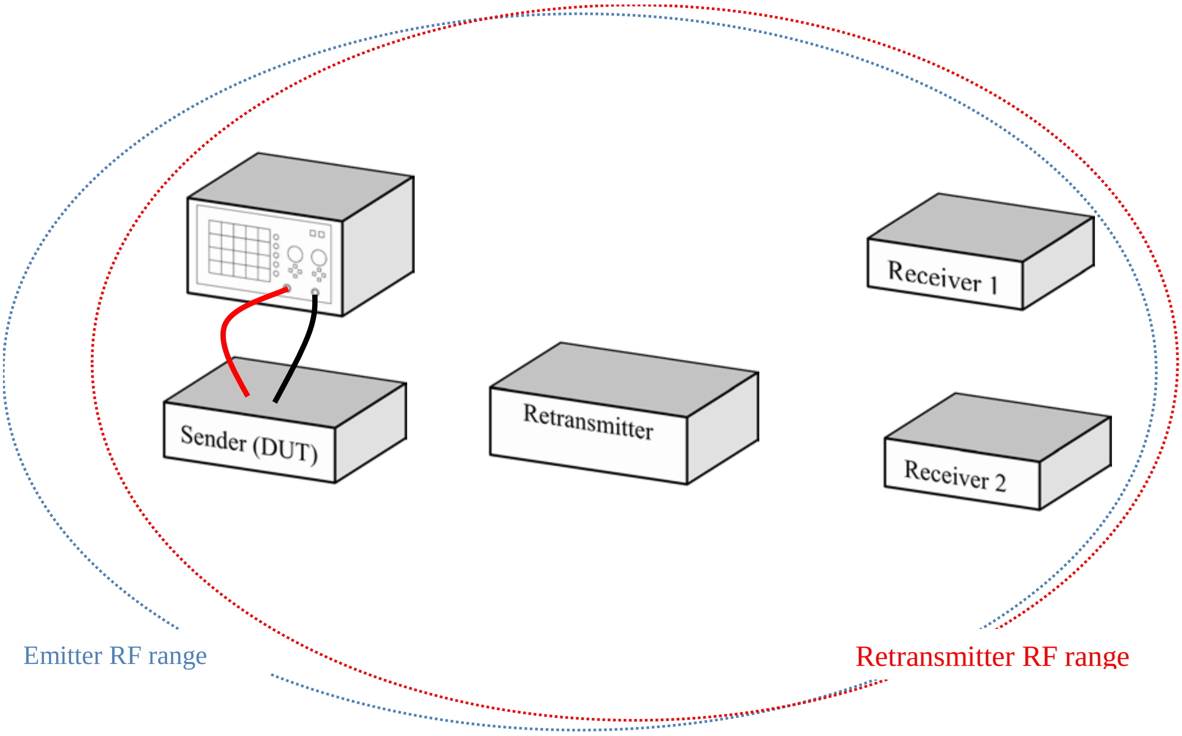

**Figure 13 Test setup**

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| ACKREP received after Fast Ack (fast). | The DUT shall transmit a frame with Fast Ack request to the linked receivers that are in the RF range of the sender. The receivers successfully respond to the request. The retransmitter repeats the frame. | The ACKREP frame shall be received after reception of all Fast Acks by the sender. The ACKREP frame shall be discarded by the sender. |
| ACKREP received after Fast Ack (slow). | The DUT shall transmit a frame with Fast Ack request to the linked receivers that are in the RF range of the sender. The receivers successfully respond to the request. The retransmitter repeats the frame. | The ACKREP frame shall be received after reception of all Fast Acks by the sender. The ACKREP frame shall be discarded by the sender. |

### 4.1.2.2.6 Maximum number of Fast Ack

In the next test, the DUT is a KNX RF Multi device that transmits frames with 64 Fast Ack request. A KNX RF Multi retransmitter shall be used to transmit the ACKREP frame, and a dedicated device (out of the RF range of sender) shall be able to answer to the 64 Fast Ack request.

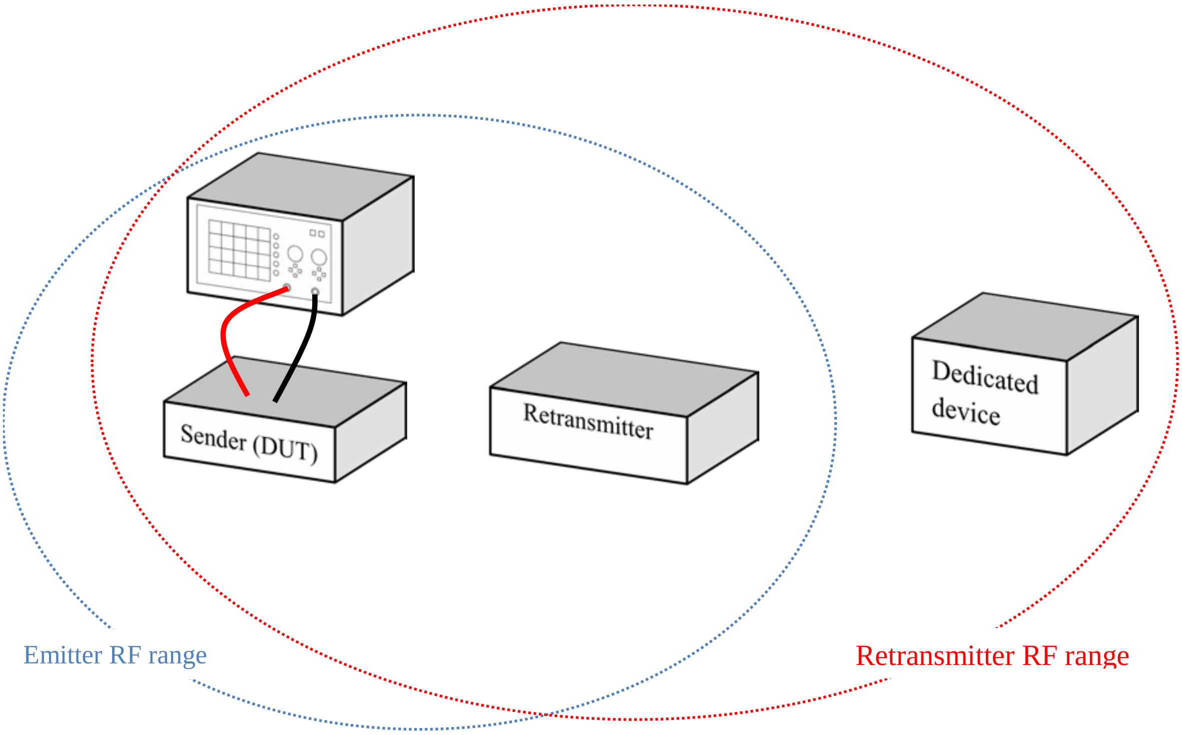

**Figure 14 Test setup**

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| ACKREP management (fast) | The DUT shall transmit a frame with 64 Fast Ack requested, and verify that the emitter correctly understands the ACKREP frame. | The DUT does not resend the original frame. |
| ACKREP management (slow) | The DUT shall transmit a frame with 64 Fast Ack requested, and verify that the emitter correctly understands the ACKREP frame. | The DUT does not resend the original frame. |

### 4.1.2.2.7 RF Multi frame retransmission

These tests use a KNX RF Multi retransmitter as DUT.

For all the following test cases, no RF disturbance is supposed on any channel.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| RF Ready frame | Transmit an RF Ready frame and inspect the retransmitted frame. | The same channel (RF Ready) is used for the retransmission.<br>The preamble is left unchanged. |
| RF Multi Fast F1 frame | Transmit an RF Multi Fast frame and inspect the retransmitted frame. | The same channel (RF Multi Fast F1) is used for the retransmission.<br>The preamble is left unchanged. |
| RF Multi Fast F2 frame | Transmit an RF Multi Fast frame and inspect the retransmitted frame. | The same channel (RF Multi Fast F2) is used for the retransmission.<br>The preamble is left unchanged. |
| RF Multi Fast F3 frame | Transmit an RF Multi Fast frame and inspect the retransmitted frame. | The same channel (RF Multi Fast F3) is used for the retransmission.<br>The preamble is left unchanged. |
| RF Multi Slow S1 frame | Transmit an RF Multi Slow frame and inspect the retransmitted frame. | The same channel (RF Multi Slow S1) is used for the retransmission.<br>The preamble is left unchanged. |
| RF Multi Slow S2 frame | Transmit an RF Multi Slow frame and inspect the retransmitted frame. | The same channel (RF Multi Slow S2) is used for the retransmission.<br>The preamble is left unchanged. |

#### 4.1.2.3 Other common parameters

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| RF-Info-Byte<br>(signal strength) | Use a device sending Telegrams with different Retransmitter-signal-strength values in the RF-Info-Byte. Hereby try to lower the received signal strength for the Retransmitter. In all cases inspect RF-Info-Byte of original and repeated Frame. | If supported, the Retransmitter should insert its measured signal strength into the RF Info-Byte, but only if the received value is 0 (normal sender) or the measured value is lower than the one from the incoming Telegram (from other Retransmitters). |
| Serial Number, Indiv. Address | Send different valid KNX RF Telegrams and compare them with the repeated ones. | The Serial Number/DomainID and Indiv.Addr. shall be unchanged by the Retransmitter . |
| History buffer | Use 2 Retransmitters and a device continuously sending Telegrams. | Each Telegram shall be transmitted only once per Retransmitter. (In all 3 transmissions: original, Retransmitter 1, Retransmitter 2) |

### 4.1.3 Network Layer

#### 4.1.3.1 KNX RF Ready, KNX RF Multi (Fast/Slow), KNX RF BiBat and BiBat 2

Retransmitters use the Routing-Counter field of the NPCI to ensure a maximum number of retransmissions and avoid an avalanche of Frames. When transmitting, end devices use a routing counter with a default value of 6, which is then decremented by every Retransmitter repeating the Telegram.

The retransmission of an incomimg KNX RF depends on a limit (minimum value) defined for the Retransmitter. This limit (MIN_ROUTING_COUNTER >= 0) can be defined by the manufacturer, thereby influencing the maximum level of retransmission per Telegram.

| Test | Test-Description | Expected Behaviour |
|---|---|---|
| NPCI (Routing Counter) | Use a device sending Telegrams with different routing counter values. Check if the original Telegram is retransmitted and if the routing counter is decremented. | Retransmitter decrements the routing counter, and doesn't retransmit Frames with a routing counter <= MIN_ROUTING_COUNTER |

## 4.1.4 Transport Layer, Application Layer, Management Layer

### 4.1.4.1 KNX RF Ready, KNX RF BiBat and BiBat 2

Management functionality or Applications are optional for Retransmitters, and if supported, shall fulfil the same requirements as end devices have to (see clause 3).

# 5 RF PB-Mode

## 5.1 Characteristics

The RF PB-Mode is part of the “E-Modes” and is based on the PB-Mode (see [02]). It is defined to allow configuration of KNX devices without any external tool or controller.

In spite of the fact that most parts of [02] are also valid for RF PB-Mode, some adaptations are introduced together with the RF system specification, mainly as a consequence of requirements for unidirectional transmitters.

### 5.1.1 Individual Address

The Individual Address of RF PB-Mode devices is preconfigured by manufacturers (and therefore is shared between devices). This is acceptable, since no point-to-point communication is used in PB-Mode, neither during configuration nor in application runtime.

### 5.1.2 Group Addresses

Group Addresses (also called object handles) are also pre-assigned by manufacturers. As identical devices share the same Group Addresses, the ‘Extended Group Address’ is defined for RF devices to ensure the uniqueness of Group Objects during runtime. This Extended Group Address is defined as the combination of the KNX Serial Number of the device and the Group Address of the Group Object.

### 5.1.3 Undirectional Sequences

As unidirectional sensors are not capable of receiving any Frames, the PB link sequence for such devices is a unidirectional sequence, i.e. the actuators do not transmit any responses to PB sensor Telegrams.

To ensure detection of errors during the link procedure, some additional information is transmitted from unidirectional sensors, allowing the actuators to detect and ignore incomplete or faulty sequences.

For further information on these RF specific adaptations, please refer to the RF system specification (see [01]).

## 5.2 Channel Setup for Easy-Mode Tests

The test sequences for PB Mode are based on a sample application including some E-Mode Channels with different Datapoints and parameters. Note that these E-Mode Channels and Datapoints are completely fictituous and not standardised.

In some of the sequences the Property Functions for E-Mode Management (see clause 6) are used to obtain the results of simulated link procedures. If these functions are not implemented, use another method to ensure the expected results.

### 5.2.1 E-Mode Channel Configuration for PB-Mode and E-Mode Tests

| **Description** | Non-generic sensor-channel (comparable to switch) | | | |
|---|---|---|---|---|
| **Channel-Index** | 0 | | | |
| **ChannelCode** | 0123h | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | E000 | I | 1 Bit | 11h (Status Info) |
| | E001 | O | 1 Bit | 22h (Switching) |
| **Parameters** | (none) | | | |

| **Description** | generic sensor-channel (Generic Push Button) | | | |
|---|---|---|---|---|
| **Channel-Index** | 0 | | | |
| **ChannelCode** | 0x0ABC | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | E000 | I | 1 Bit | 11h |
| | E001 | O | 1 Bit | 22h |
| | E002 | O | 1 Bit | 33h |
| | E001 | O | 1 Bit | 44h |
| **Parameters** | 1 | AJS (Adjustable Selection) | | |
| | 2 | Sample Parameter | | |

| **Description** | Sample Actuator Channel | | | |
|---|---|---|---|---|
| **Channel-Index** | 1 | | | |
| **ChannelCode** | 0x2001 | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | E100 | I | 1 Bit | 11h |
| | E101 | I | 1 Byte | 51h, 52h, 53h |
| | E102 | I | 8 Bytes | 61h |
| | E103 | O | 1 Bit | 11h (Info) |
| **Parameters** | (none) | | | |

| **Description** | Sample “Logic”Actuator Channel | | | |
|---|---|---|---|---|
| **Channel-Index** | 2 | | | |
| **ChannelCode** | 0x4001 | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | E200 | I,X | 1 Bit | 22h, 88h |
| | E201 | I,X | 1 Bit | 22h, 88h |
| | E202 | I,X | 1 Bit | 22h, 88h |
| | E203 | O | 1 Bit | 22h, 88h |
| **Parameters** | 1 | Logical Function (0 = AND, 1 = OR) | | |

## 5.3 “Teach-In” and “Teach-Out” of non-generic sensor channel

Purpose: check if sensor channel (CH0) configured as non-generic channel (Chn.Code 0123h) is able to set and delete links correctly.

Preparation: Delete Association Table;

Set Channel 0 to non-generic configuration;

trigger BDUT to start the link sequence.

```
► 32 5F424455545F 00 1000 0000 E8 03 E4  00 00 3B 20 00 01 02      StartLink (MANU=0001,OBJ=2)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 30 11 22 00      ChnFuncAct (CHC=1122)
► 32 5F424455545F 00 1000 0000 EA 03 E4  00 00 3B 40 01 23 00      ChnFuncSen (CHC=0123)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 70 00 00 00      begin connection
► 32 5F424455545F 00 1000 0000 EC 03 E4  00 00 3B 80 11 00 00      SetDelLink (CC=11, Obj is Input)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 91 11 1000       LinkResp (use address 1000)
► 32 5F424455545F 00 1000 0000 EE 03 E4  00 00 3B 80 22 E001       SetDelLink (CC=22, Obj E001)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 90 22 E001       LinkResp (Link added)
► 32 5F424455545F 00 1000 0000 E0 03 E4  00 00 3B A0 00 00 00      StopLink (Link successful)
```

Check if BDUT has created the links correctly:

```
◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3F 00 00               read Association 0
► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 3F 00 00 4D4153544552 1000 E000
                                                                    return (MASTER,1000,E000)
◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3F 00 01               read Association 1
► 32 4D4153544552 00 1000 AFFE 65 02 C9  00 3F FF                  return FF => no more Associations
```

Preparation: trigger BDUT to start the link sequence.

```
► 32 5F424455545F 00 1000 0000 E8 03 E4  00 00 3B 20 00 01 02      StartLink (MANU=0001,OBJ=2)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 30 11 22 00      ChnFuncAct (CHC=1122)
► 32 5F424455545F 00 1000 0000 EA 03 E4  00 00 3B 40 01 23 00      ChnFuncSen (CHC=0123)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 70 00 00 00      begin connection
► 32 5F424455545F 00 1000 0000 EC 03 E4  00 00 3B 80 11 00 00      SetDelLink (CC=11, Obj is Input)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 91 11 10 00      LinkResp (use address 1000)
► 32 5F424455545F 00 1000 0000 EE 03 E4  00 00 3B 80 22 E0 01      SetDelLink (CC=22, Obj E001)
◄ 00 4D4153544552 00 AFFE 0000 E0 03 E4  00 00 3B 92 22 E0 01      LinkResp (Link deleted)
► 32 5F424455545F 00 1000 0000 E0 03 E4  00 00 3B A0 00 00 00      StopLink (Link successful)
```

Check if BDUT has deleted the links correctly:

```
◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3F 00 00               read Association 0
► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 3F FF                  return FF => no more Associations
```

## 5.4 “Teach-In” of generic sensor channel

Purpose: check if sensor channel (CH0) configured as generic subchannel (Chn.Code 0x0ABC) is able to set links and adjust the AJS correctly (AJS 2 should be default for actuator channel code 02h15).

Preparation: Set Channel 0 to generic configuration;

set AJS of channel 0 to value 3,

trigger BDUT to start the link sequence of channel 0.

```
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 20 00 01 04   StartLink (MANU=0001,OBJ=4)
```

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 02 15 00 | *ChnFuncAct (CHC=0215)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 40 0A BC 00 | *ChnFuncSen (CHC=0ABC)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj is Input)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use my address)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj E001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 22 E0 01 | *LinkResp (Link added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 33 E0 02 | *SetDelLink (CC=33, Obj E002)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 33 E0 02 | *LinkResp (Link added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 80 44 E0 03 | *SetDelLink (CC=44, Obj E003)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 44 E0 03 | *LinkResp (Link added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT has changed AJS correctly:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 41 00 01 | *Read AJS* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 41 00 00 02 | *Return AJS = 2* |

## 5.5 “Teach-Out” of generic sensor channel

Purpose: check if sensor channel (CH0) configured as generic channel (Chn.Code 0x0ABC) is able to delete links and does not adjust the AJS due to a “Teach-Out” sequence.

Preparation: clear association table,

activate subchannel 1 of channel 0,

set AJS of channel 0 to value 3

Trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 20 00 01 04 | *StartLink (MANU=0001,OBJ=4)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 02 15 00 | *ChnFuncAct (CHC=0215) => AJS2* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 40 0A BC 00 | *ChnFuncSen (CHC=0ABC)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj E000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use my address)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj E001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 92 22 E0 01 | *LinkResp (Link deleted)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 80 33 E0 02 | *SetDelLink (CC=33, Obj E002)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 92 33 E0 02 | *LinkResp (Link deleted)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 44 E0 03 | *SetDelLink (CC=44, Obj E003)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 92 44 E0 03 | *LinkResp (Link deleted)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT did not change the AJS (because actuator has deleted the links).

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 41 00 01 | *Read AJS* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 41 00 00 03 | *Return AJS = 3* ***(hasn’t changed !!)*** |

Check if the association table is still empty.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F FF | *Return FF => no Assocs* |

## 5.6 Linking generic sensor channel with no matching Datapoints

Purpose: check if sensor channel (CH0) configured as generic channel (Chn.Code 0x0ABC) does not adjust the AJS due to the fact that no Datapoints could be linked during sequence.

Preparation: Trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 20 00 01 04 | *StartLink (MANU=0001,OBJ=4)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 01 35 00 | *ChnFuncAct (CHC=0135) => AJS 1* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 40 0A BC 00 | *ChnFuncSen (CHC=0ABC)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj E000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use my adress)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj E001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 93 22 E0 01 | *LinkResp (Link not added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 80 33 E0 02 | *SetDelLink (CC=33, Obj E002)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 93 33 E0 02 | *LinkResp (Link not added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 80 44 E0 03 | *SetDelLink (CC=44, Obj E003)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 93 44 E0 03 | *LinkResp (Link not added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT did not change the AJS (because the actuator could not add the links).

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 41 00 01 | *Read AJS* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 41 00 00 03 | *Return AJS = 3* ***(hasn't changed !!)*** |

Check if the association table is still empty.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F FF | *Return FF => no Assocs* |

## 5.7 Linking actuator channel

Purpose: Check if actuator channel (CH1) is able to set and delete links correctly. The same link sequence is sent twice to the actuator channel. The first time the actuator shall insert all links, the second time it shall remove the links again. Additionally, it is tested that an actuator also accepts linking with manufacturer code other than the own.

Preparation: clear association table,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 04 | *StartLink (MANU=**1234**,OBJ=4)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT has generated the links correctly.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 00 E1 00 | *Return (MASTER,1000,E100)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01 | *Read assoc 1* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F 00 01 4D 41 53 54 45 52 10 01 E1 01 | *Return (MASTER,1001,E101)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02 | *Read assoc 2* |
| ►32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 02 4D 41 53 54 45 52 10 02 E1 02 | *Return (MASTER,1002,E102)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 03 | *Read assoc 3* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F FF | *Return FF => no more Assocs* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT,

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 04 | *StartLink (MANU=1234,OBJ=4)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 92 22 10 00 | *LinkResp (link deleted)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 92 51 10 01 | *LinkResp (link deleted)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 92 61 10 02 | *LinkResp (link deleted)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT has deleted the links correctly.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F FF | *Return FF => no Assocs* |

## 5.8 Actuator with “single connect” - Datapoints

Purpose: First the objects, which are "connectable only once" (X flag, see Supplement 12 Channel Codes), with the addresses E200h, E201h, E202h of the logical actuator channel in BDUT (channel 2) are linked to three different devices (A, B, C). A fourth device (D) is unable to link to any of these three objects. B is then unlinked, and D can link to the second object.

Sensor device A (Ser.Nr. “ANDREA”) gets linked

Preparation: delete association table,

activate programming mode of channel 2 (logical actuator) in BDUT.

| | |
|---|---|
| ◄ 00 41 4E 44 52 45 41 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01 | *StartLink (MANU=1234,OBJ=1)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 30 04 01 00 | *ChnFuncAct (CHC=0401)* |
| ◄ 00 41 4E 44 52 45 41 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 41 4E 44 52 45 41 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 30 00 | *SetDelLink (CC=22, Obj 3000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 90 22 30 00 | *LinkResp (link added)* |
| ◄ 00 41 4E 44 52 45 41 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 10 00 | *StopLink (Link successful)* |

Sensor device B (Ser.Nr. “JOSEPH”) gets linked

Preparation: activate programming mode of channel 2 (logical actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01 | *StartLink (MANU=1234,OBJ=1)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 30 04 01 00 | *ChnFuncAct (CHC=0401)* |
| ◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 40 00 | *SetDelLink (CC=22, Obj 4000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 90 22 40 00 | *LinkResp (link added)* |
| ◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Sensor device C (Ser.Nr. “THOMAS”) gets linked

Preparation: activate programming mode of channel 2 (logical actuator) in BDUT.

| | |
|---|---|
| ◄ 00 54 48 4F 4D 41 53 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01 | *StartLink (MANU=1234,OBJ=1)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 30 04 01 00 | *ChnFuncAct (CHC=0401)* |
| ◄ 00 54 48 4F 4D 41 53 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 54 48 4F 4D 41 53 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 50 00 | *SetDelLink (CC=22, Obj 5000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 90 22 50 00 | *LinkResp (link added)* |
| ◄ 00 54 48 4F 4D 41 53 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Sensor device D (Ser.Nr. “HELMUT”) gets linked, but now no connectable Datapoints exists.

Preparation: activate programming mode of channel 2 (logical actuator) in BDUT.

| | |
|---|---|
| ◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01 | *StartLink (MANU=1234,OBJ=1)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 30 04 01 00 | *ChnFuncAct (CHC=0401)* |
| ◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 01 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 60 00 | *SetDelLink (CC=22, Obj 6000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 93 22 60 00 | *LinkResp (link not added)* |
| ◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check if BDUT has generated the links correctly.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 00 41 4E 44 52 45 41 30 00 E2 00 | *Return (ANDREA,3000,E200)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01 | *Read assoc 1* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F 00 01 4A 4F 53 45 50 48 40 00 E2 01 | *Return (JOSEPH,4000,E201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02 | *Read assoc 2* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 02 54 48 4F 4D 41 53 50 00 E2 02 | *Return (THOMAS,5000,E202)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 03 | *Read assoc 3* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F FF | *Return FF => no more Assocs* |

Sensor device B (Ser.Nr. “JOSEPH”) gets linked, actuator should teach it out.

Preparation: activate programming mode of channel 2 (logical actuator) in BDUT.

```
◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01        StartLink (MANU=1234,OBJ=1)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 30 04 01 00        ChnFuncAct (CHC=0401)
◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 00        ChnFuncSen (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 70 00 00 00        BeginConnection
◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 40 00        SetDelLink (CC=22, Obj 4000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 92 22 40 00        LinkResp (link deleted)
◄ 00 4A 4F 53 45 50 48 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 00 00        StopLink (Link successful)
```

Check if BDUT has generated the links correctly.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00                 Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 00 41 4E 44 52 45 41 30 00 E2 00
                                                                            Return (ANDREA,3000,E200)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01                 Read assoc 1
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F 00 01 54 48 4F 4D 41 53 50 00 E2 02
                                                                            Return (THOMAS,5000,E202)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02                 Read assoc 2
► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F FF                     Return FF => no more Assocs
```

Sensor device D (Ser.Nr. “HELMUT”) gets linked again, now it should be connected to E201.

Preparation: activate programming mode of channel 2 (logical actuator) in BDUT.

```
◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 20 12 34 01        StartLink (MANU=1234,OBJ=1)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 30 04 01 00        ChnFuncAct (CHC=0401)
◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 40 11 22 01        ChnFuncSen (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 70 00 00 00        BeginConnection
◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B 80 22 60 00        SetDelLink (CC=22, Obj 6000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 22 60 00        LinkResp (link added)
◄ 00 48 45 4C 4D 55 54 00 05 FF 00 00 E0 03 E4 00 00 3B A0 00 00 00        StopLink (Link successful)
```

Check if BDUT has generated the links correctly.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00                 Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6B 02 C9 00 3F 00 00 41 4E 44 52 45 41 30 00 E2 00
                                                                            Return (ANDREA,3000,E200)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01                 Read assoc 1
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 01 48 45 4C 4D 55 54 60 00 E2 01
                                                                            Return (HELMUT,6000,E201)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02                 Read assoc 2
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F 00 02 54 48 4F 4D 41 53 50 00 E2 02
                                                                            Return (THOMAS,5000,E202)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 03                 Read assoc 3
► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F FF                     Return FF => no more Assocs
```

## 5.9 Timeout behaviour

Purpose: With BDUT, a sensor channel (CH0) and an actuator channel (CH1) are linked. During the link sequence a Frame is delayed for a specific time (see PIXIT supplied by the manufacturer). Due to this timeout the BDUT shall leave the programming mode automatically and stop to respond.

Preparation: clear association table in BDUT,

choose subchannel 0 of channel 0,

trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj. is Input)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use address 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj E001)* |
| ..................................... *(delay)*..................................... | |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 22 E0 01 | *LnkResp delayed=> no answer* |

Check that no links have been generated in BDUT

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F FF | *Return FF => no links* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 04 | *StartLink (MANU=1234,OBJ=4)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |
| ..................................... *(delay)*..................................... | |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *delayed => no answer* |

Check that no links have been generated in BDUT

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F FF | *Return FF => no links* |

## 5.10 Behaviour on unexpected Frames

Purpose: In the BDUT, the sensor channel (CH0) and the actuator channel (CH1) are linked. During the link sequence "unexpected" Frames are sent to the BDUT. BDUT shall ignore these and continue executing the link sequence.

Preparation: clear association table in BDUT,

activate programming mode of channel 0 (sensor) in BDUT.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |

```
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 80 11 00 00     SetDelLink (CC=11, Obj INPUT)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00
      ChnFuncAct(CHC=1122h)(unexpected)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00     LinkResp (use address 1000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 22 E0 01     SetDelLink (CC=22, Obj E001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00     BeginConnection (unexpected)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 22 E0 01     LinkResp (link added)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B A0 00 00 00     Stop Link (successful)
```

Check that the input Datapoints of sensor has been correctly linked

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00              Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 00 E0 00
                                                                         Return (MASTER;1000;E000)
```

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 04     StartLink (MANU=1234,OBJ=4)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 30 02 01 00     ChnFuncAct (CHC=0201)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00     ChnFuncAct (CHC=1122) (unexp.)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00     ChnFuncSen (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 70 00 00 00     BeginConnection
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00     BeginConnection (unexp.)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00     SetDelLink (CC=22, Obj 1000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 22 10 00     LinkResp (link added)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00     ChnFuncSen (CHC=1122) (unexp.)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01     SetDelLink (CC=51, Obj 1001)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 90 51 10 01     LinkResp (link added)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00     LinkResp (use address) (unexp.)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02     SetDelLink (CC=61, Obj 1002)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 90 61 10 02     LinkResp (link added)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 00 12 34     ChnParaResp (unexp.)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00     SetDelLink (CC=11, Obj INPUT)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 91 11 E1 03     LinkResp (use address E103)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00     StopLink
```

Check that links have been generated correctly in BDUT

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00              Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 00 E0 00
                                                                         Return (MASTER;1000;E000)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01              Read assoc 1
► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F 00 01 4D 41 53 54 45 52 10 00 E1 00
                                                                         Return (MASTER;1000;E100)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02              Read assoc 2
► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 3F 00 02 4D 41 53 54 45 52 10 01 E1 01
                                                                         Return (MASTER;1001;E101)
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 03              Read assoc 3
► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F 00 03 4D 41 53 54 45 52 10 02 E1 02
                                                                         Return (MASTER;1002;E102)
```

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 04 | *Read assoc 4* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 69 02 C9 00 3F FF | *Return FF => no more assocs* |

## 5.11 Incorrect number of Group Objects

Purpose: Actuator channel (CH1) in BDUT is linked by a link sequence, with a sensor announcing an incorrect number of Group Objects. In this case, the actuator shall declare the link sequence invalid and shall not insert the links.

Preparation: clear association table in BDUT,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 05 | *StartLink (MANU=1234,OBJ=5)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check that no links have been generated in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6B 02 C9 00 3F FF | *Return FF => no assocs* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check that no links have been generated in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 69 02 C9 00 3F FF | *Return FF => no assocs* |

## 5.12 Actuator with full association table

Purpose: The association table (max. 5 entries) is filled with 4 entries. The actuator channel (CH1) in BDUT is linked by a link sequence, which contains more compatible Group Objects than free entries in the association table. In a second step, two entries are deleted from the association table, and the same link sequence shall then be successfully accepted. A second repetition of this link sequence is again not successful, since the association table is full again.

Preparation: clear association table in BDUT,

enter 4 links in association table:

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 00 E1 00
                                                                   SetLink (MASTER,A100,E100)
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00                    Return OK
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 01 E1 01
                                                                   SetLink (MASTER,A101,E101)
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F 00                    Return OK
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 02 E1 02
                                                                   SetLink (MASTER,A102,E102)
► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00                    Return OK
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 AA AA E0 00
                                                                   SetLink (MASTER,A103,E103)
► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F 00                    Return OK
```

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02      StartLink (MANU=1234,OBJ=2)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 30 02 01 00      ChnFuncAct (CHC=0201)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00      ChnFuncSen (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 70 00 00 00      BeginConnection
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00      SetDelLink (CC=22, Obj 1000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 90 22 10 00      LinkResp (link added)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01      SetDelLink (CC=51, Obj 1001)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 94 51 10 01      LinkResp (link error)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A4 00 00 00      StopLink (Link aborted)
```

Check that no more than 4 entries are present in the association table of the BDUT:

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 04                Read assoc 4
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F FF                    Return FF => no assoc
```

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02      StartLink (MANU=1234,OBJ=2)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 30 02 01 00      ChnFuncAct (CHC=0201)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00      ChnFuncSen (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 70 00 00 00      BeginConnection
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 A1 00      SetDelLink (CC=22, Obj A100)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 92 22 A1 00      LinkResp (link deleted)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 A1 01      SetDelLink (CC=51, Obj 1001)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 92 51 A1 01      LinkResp (link deleted)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00      StopLink (Link ok)
```

Check that no more than 2 entries are present in the association table of the BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02 | *Read assoc 2* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6B 02 C9 00 3F FF | *Return FF => no assoc* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link ok)* |

Check that more than 2 entries are present in the association table of the BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02 | *Read assoc 2* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F 00 02 4D 41 53 54 45 52 A1 02 E1 02 | *Return assoc A102,E102* |

## 5.13 Sensor with full association table

Purpose: The association table (max. 5 entries) is filled with 5 entries. The sensor channel (CH0) in BDUT is linked by a link sequence, which contains one status object, so that the sensor shall quit the link sequence (association table full). In a second step, one entry is deleted from the association table, and the same link sequence shall then be successfully when repeated.

Preparation: clear association table in BDUT,

enter 5 links in association table:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 00 E1 00 | *SetLink (MASTER,A100,E100)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 3F 00 | *Return OK* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 01 E1 01 | *SetLink (MASTER,A101,E101)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F 00 | *Return OK* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 A1 02 E1 02 | *SetLink (MASTER,A102,E102)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 69 02 C9 00 3F 00 | *Return OK* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 AA AA E0 00 | *SetLink (MASTER,AAAA,E000)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6B 02 C9 00 3F 00 | *Return OK* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 3F 00 00 4D 41 53 54 45 52 10 00 E0 00 | *SetLink (MASTER,1000,E103)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 | *Return OK* |

Preparation: trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj is Input)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 20 00 | *LinkResp (use address 2000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B A4 00 00 00 | *StopLink (Link **aborted**)* |

Preparation: trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj is Input)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use address 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj A100)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 92 22 E0 01 | *LinkResp (link deleted)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link ok)* |

Preparation: trigger BDUT to start the link sequence of channel 0.

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj E000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 20 00 | *LinkResp (use address 2000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj A100)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 22 E0 01 | *LinkResp (link added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link ok)* |

Check that link has been generated correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 04 | *Read assoc 4* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 04 4D 41 53 54 45 52 20 00 E0 00 | *assoc (Master;2000;E000)* |

## 5.14 StopLink behaviour

Purpose: During linking of the actuator channel (CH1), in various positions of the link sequence a StopLink is sent. The BDUT shall abort the link sequence. It is further checked, that the BDUT reacts correctly on the StopLink flags (error/success), and that the BDUT correctly checks the number of objects.

Preparation: clear association table in BDUT,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A4 00 00 00 | *StopLink (Link aborted)* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A4 00 00 00 | *StopLink (Link aborted)* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A4 00 00 00 | *StopLink (Link aborted)* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A4 00 00 00 | *StopLink (Link aborted)* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link ok, too many Obj=3)* |

Check that no associations have been generated in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F FF | *Return FF => no assoc* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 90 22 10 00 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 90 51 10 01 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link ok)* |

Check that association has been generated correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6B 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 00 E1 00 | *Return assoc 0* |

## 5.15 Sub function

Purpose: the actuator channel (CH1) in BDUT receives a link sequence of a sensor twice: the sensor tries to configure the scene number of the actuator via sub function 1 of the SetDelLink service. If this sub-function is not supported, the actuator shall respond with "link not added" and ignore this service when counting offered objects.

Preparation: clear association table,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=2)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 81 22 00 02 | *SetDelLink (CC=22, SubFunc 01)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 93 22 00 02 | *LinkResp (link not added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 81 22 00 04 | *SetDelLink (CC=22, SubFunc 01)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 93 22 00 04 | *LinkResp (link not added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 90 61 10 02 | *LinkResp (link added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link succesful)* |

Check that link has been generated correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 02 E1 02 | *Return (MASTER,1002,E102)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01 | *Read assoc 1* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F FF | *Return FF => no more assocs* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 20 12 34 02 | *StartLink (MANU=1234,OBJ=4)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B 30 02 01 00 | *ChnFuncAct (CHC=0201)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EC 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 81 22 00 02 | *SetDelLink (CC=22, SubFunc 01)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EE 03 E4 00 00 3B 93 22 00 02 | *LinkResp (link not added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 81 22 00 04 | *SetDelLink (CC=22, SubFunc 01)* |

| | |
|---|---|
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E0 03 E4 00 00 3B 93 22 00 04 | *LinkResp (link not added)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02 | *SetDelLink (CC=61, Obj 1002)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 92 61 10 02 | *LinkResp (link deleted)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj=INPUT)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 91 11 E1 03 | *LinkResp (use my address)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |

Check that link has been deleted correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 67 02 C9 00 3F FF | *Return FF => no more assocs* |

## 5.16 Unidirectional sensor (normal case)

Purpose: the actuator channel (CH1) in BDUT receives the link sequence of a unidirectional sensor twice: after the first time the links shall have been created correctly, after the second time the links shall have been deleted again.

Preparation: clear association table,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 2C 12 34 02 | *StartLink (UNI, MANU=1234,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 40 20 10 | *DeviceDecriptorResp Type 0* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 42 00 01 12 34 01 3F 23 00 0F FF 00 00 00 00 | *DeviceDecriptorResp Type 2* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 00 01 03 | *ChnParaResp (P=0,CH=1,C=3)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 01 00 01 | *ChnParaResp (P=1, V=0001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 02 00 01 | *ChnParaResp (P=2, V=0001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 03 00 02 | *ChnParaResp (P=3, V=0002)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Sequence ok)* |

Check that links have been created correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 00 3F 00 00 4D 41 53 54 45 52 10 00 E1 00 | *Return (MASTER;1000;E100)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01 | *Read assoc 1* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F 00 01 4D 41 53 54 45 52 10 01 E1 01 | *Return (MASTER;1000;E100)* |
| 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 02 | *Read assoc 2* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 3F FF | *Return FF => no more assocs* |

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 2C 12 34 02 | *StartLink (UNI, MANU=1234,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 40 20 10 | *DeviceDecriptorResp Type 0* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 42 00 01 12 34 01 3F 23 00 0F FF 00 00 00 00 | *DeviceDecriptorResp Type 2* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00 | *ChnFuncSen (CHC=1122)* |

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 00 01 03 | *ChnParaResp (P=0,CH=1,C=3)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 01 00 01 | *ChnParaResp (P=1, V=0001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 02 00 01 | *ChnParaResp (P=2, V=0001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 03 00 02 | *ChnParaResp (P=3, V=0002)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00 | *SetDelLink (CC=22, Obj 1000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01 | *SetDelLink (CC=51, Obj 1001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00 | *StopLink (Sequence ok)* |

Check that links have been deleted correctly in BDUT:

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 69 02 C9 00 3F FF | *Return FF => no assoc* |

## 5.17 Optional PB-Services

Purpose: simulate link sequence for sensor channel (CH0) with optional (not mandatory) PB-services. If the BDUT doesn’t support these optional services, it shall silently discard them.

Preparation: clear association table,

activate programming mode of channel 1 (actuator) in BDUT.

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 01 04 | *Delete AssocTab* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 01 04 00 | *Return OK (gelöscht)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 01 05 00 00 01 23 | *Select SubChannel 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 01 05 00 00 01 23 | *Return SubChannel CHC=0123h* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 10 00 00 00 | ***Enter Config Mode*** |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 01 03 01 01 00 | *Start LinkSequence Sensor CH0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 01 03 01 01 00 | *Return OK* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00 | *ChnFuncAct (CHC=1122)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 40 01 23 00 | *ChnFuncSen (CHC=0123)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00 | *BeginConnection* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 80 11 00 00 | *SetDelLink (CC=11, Obj E000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00 | *LinkResp (use address 1000)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 80 22 E0 01 | *SetDelLink (CC=22, Obj E001)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 90 22 E0 01 | *LinkResp (Link added)* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B A0 00 00 00 | *StopLink (Link successful)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B B0 00 00 00 | ***Quit Config Mode*** |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B C0 00 00 00 | ***Reset Installation*** |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00 | *Read Assoc 0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F 00 00 4D4153544552 1000 E000 | *Return (MASTER,1000,E000)* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 01 | *Read Assoc 1* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F FF | *Return FF => no Associations left* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 10 00 00 00 | ***Enter Config Mode*** |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 01 03 01 01 00 | *Start LinkSequence Sensor CH0* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 61 02 C9 01 03 01 01 00 | *Return OK* |
| ► 32 5F 42 44 55 54 5F 00 10 00 00 00 E2 03 E4 00 00 3B 20 00 01 02 | *StartLink (MANU=0001,OBJ=2)* |

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 30 11 22 00    ChnFuncAct (CHC=1122)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E4 03 E4 00 00 3B 40 01 23 00    ChnFuncSen (CHC=0123)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 70 00 00 00    BeginConnection
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E6 03 E4 00 00 3B 80 11 00 00    SetDelLink (CC=11, Obj E000)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 91 11 10 00    LinkResp (use address 1000)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 E8 03 E4 00 00 3B 80 22 E0 01    SetDelLink (CC=22, Obj E001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 92 22 E0 01    LinkResp (Link deleted)
► 32 5F 42 44 55 54 5F 00 10 00 00 00 EA 03 E4 00 00 3B A0 00 00 00    StopLink (Link successfull)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B B0 00 00 00    Quit Config Mode
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B C0 00 00 00    Reset Installation
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00             Read Assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6D 02 C9 00 3F FF                Return FF => no associations left
```

## 5.18 Unidirectional sensor (error case)

Purpose: the actuator channel (CH1) in BDUT receives an incorrect link sequence of a unidirectional sensor twice: the number of objects given in the StartLink Frame is different from the actual number of objects. In both cases the BDUT shall not generate any links in the association table (the entire link sequence sent to the BDUT is fictitious).

Preparation: clear association table,

activate programming mode of channel 1 (actuator) in BDUT.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 2C 12 34 02    StartLink (UNI, MANU=1234,OBJ=2)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 40 20 10                   DeviceDecriptorResp Type 0
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 42 00 01 12 34 01 3F 23 00 0F FF 00 00 00 00
                                                               DeviceDecriptorResp Type 2
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00    ChnFuncSen (CHC=1122)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 00 01 03    ChnParaResp (P=0,CH=1,C=3)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 01 00 01    ChnParaResp (P=1, V=0001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 02 00 01    ChnParaResp (P=2, V=0001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 03 00 02    ChnParaResp (P=3, V=0002)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00    SetDelLink (CC=22, Obj 1000)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00    StopLink (Sequence ok)
```

Check that no links have been generated in BDUT:

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00             Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 6F 02 C9 00 3F FF                Return FF => no assoc
```

Preparation: activate programming mode of channel 1 (actuator) in BDUT.

```
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 2C 12 34 02    StartLink (UNI, MANU=1234,OBJ=2)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 40 20 10                   DeviceDecriptorResp Type 0
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 42 00 01 12 34 01 3F 23 00 0F FF 00 00 00 00
                                                               DeviceDecriptorResp Type 2
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 40 11 22 00    ChnFuncSen (CHC=1122)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 00 01 03    ChnParaResp (P=0,CH=1,C=3)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 01 00 01    ChnParaResp (P=1, V=0001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 02 00 01    ChnParaResp (P=2, V=0001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 60 03 00 02    ChnParaResp (P=3, V=0002)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 22 10 00    SetDelLink (CC=22, Obj 1000)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 51 10 01    SetDelLink (CC=51, Obj 1001)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B 80 61 10 02    SetDelLink (CC=61, Obj 1002)
◄ 00 4D 41 53 54 45 52 00 AF FE 00 00 E0 03 E4 00 00 3B A0 00 00 00    StopLink (Sequence ok)
```

Check that no links have been generated in BDUT:

```
◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C8 00 3F 00 00             Read assoc 0
► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 3F FF                 Return FF => no assoc
```

## 5.19 PB-Mode on unidirectional transmitters

### 5.19.1 Link sequence

Check that the BDUT sends the link sequence correctly for all channels of the device:

Example:

```
► 23 555042413336 00 05FF 0000 E8 03 E4  00 00 3B 2C 00 01 05   Start Link
► 23 555042413336 00 05FF 0000 EA 03 40  21 10                  DD type0
► 23 555042413336 00 05FF 0000 EC 03 42  00 01 A2 01 11 3F 23 11 00 21 00 00 00 00
                                                                     DD type2
► 23 555042413336 00 05FF 0000 EE 03 E4  00 00 3B 40 03 11 00   ChFctSensor (0x311)
► 23 555042413336 00 05FF 0000 E0 03 E4  00 00 3B 60 00 00 03   ChParamResp (announce nr.of.param)
► 23 555042413336 00 05FF 0000 E2 03 E4  00 00 3B 60 01 00 01   ChParamResp (value of Param 1)
► 23 555042413336 00 05FF 0000 E4 03 E4  00 00 3B 60 02 00 00   Ch param resp
► 23 555042413336 00 05FF 0000 E6 03 E4  00 00 3B 60 03 00 01   Ch param resp
► 23 555042413336 00 05FF 0000 E8 03 E4  00 00 3B 80 01 00 01   Set/ delete Link (00h1)
► 23 555042413336 00 05FF 0000 EA 03 E4  00 00 3B 80 0C 00 02   Set/ delete Link (00h2)
► 23 555042413336 00 05FF 0000 EC 03 E4  00 00 3B 80 0B 00 03   Set/ delete Link (00h3)
► 23 555042413336 00 05FF 0000 EE 03 E4  00 00 3B 80 05 00 04   Set/ delete Link (00h4)
► 23 555042413336 00 05FF 0000 E0 03 E4  00 00 3B 80 08 00 05   Set/ delete Link (00h5)
► 23 555042413336 00 05FF 0000 E2 03 E4  00 00 3B A0 00 00 00   Stop Link
```

### 5.19.2 Generic channel

If the BDUT supports generic channels, check that the AJS (adjustable selection) is sent correctly in the link sequence when the local setting is changed.

Example (AJS is locally set to 2)

```
► 33 555042413336 00 05FF 0000 E4 03 E4  00 00 3B 2C 00 01 05   Start Link
► 33 555042413336 00 05FF 0000 E6 03 40  21 10                  DD type0
► 33 555042413336 00 05FF 0000 E8 03 42  00 01 A2 01 11 3F 23 11 00 21 00 00 00 00
                                                                     DD type2
► 33 555042413336 00 05FF 0000 EA 03 E4  00 00 3B 40 03 11 0    0          ChFctSensor (0x311)
► 33 555042413336 00 05FF 0000 EC 03 E4  00 00 3B 60 00 00 03   ChParamResp
► 33 555042413336 00 05FF 0000 EE 03 E4  00 00 3B 60 01 00 02   Ch param resp (AJS = 2)
► 33 555042413336 00 05FF 0000 E0 03 E4  00 00 3B 60 02 00 00   ChParamResp
► 33 555042413336 00 05FF 0000 E2 03 E4  00 00 3B 60 03 00 01   ChParamResp
► 33 555042413336 00 05FF 0000 E4 03 E4  00 00 3B 80 01 00 01   Set/ delete Link (00h1)
► 33 555042413336 00 05FF 0000 E6 03 E4  00 00 3B 80 0C 00 02   Set/ delete Link (00h2)
► 33 555042413336 00 05FF 0000 E8 03 E4  00 00 3B 80 0B 00 03   Set/ delete Link (00h3)
► 33 555042413336 00 05FF 0000 EA 03 E4  00 00 3B 80 05 00 04   Set/ delete Link (00h4)
► 33 555042413336 00 05FF 0000 EC 03 E4  00 00 3B 80 08 00 05   Set/ delete Link (00h5)
► 33 555042413336 00 05FF 0000 EE 03 E4  00 00 3B A0 00 00 00   Stop Link
```

### 5.19.3 Device Descriptor Type 2

Check that the Group Addresses are assigned to the Group Objects in the order as E-Mode Channels appear in DD2 with respect to the Group Object order in the E-Mode channel definition. (Please refer to the various Chapters with E-Mode Channel definitions in [03].)

(compare to the example sequences above)

# 6 RF Other EASY-Modes (and S-Mode)

The following test sequences shall be used with RF devices supporting other E-Modes (e.g. Ctrl-Mode) and S-Mode. They are used to approve the implementation of the RF specific management Property-Functions.

Note: These tests refer to a sample test application as described in clause 5.2.

## 6.1 ObjectValue (PID_PB_OBJVALUE)

The Property – function "ObjectValue" (Property ID = PID_PB_OBJVALUE) is tested, which can be used for reading/writing a Group Object. The addressed object and the length of the data part shall be correct, otherwise the property shall return an error (ReturnCode = FFh).

Check that the BDUT returns an error if it receives a FuncPrStRead to an invalid object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E1FF | *FuncPrStRead get E1FF* |
| ► 32 4D4153544552 00 1000 AFFE 6B 02 C9  00 3E FF | *error (inv. Obj.)* |

Check that the BDUT reads and writes correctly to a 1 byte object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E100 01 | *set E100 (1 byte)* |
| ► 32 4D4153544552 00 1000 AFFE 6D 02 C9  00 3E 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E100 | *get E100* |
| ► 32 4D4153544552 00 1000 AFFE 6F 02 C9  00 3E 00 01 | *return = 01* |

Check that the BDUT rejects writing 1 byte object with 3 bytes data

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E100 01 02 03 | *set E100 (3 Bytes)* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9  00 3E FF | *error (length)* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E100 | *get E100* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 3E 00 01 | *return = 01* |

Check that the BDUT reads and writes correctly to an 8 byte object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E102 0000000000000000 | *set E102 (8 bytes)* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9  00 3E 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E102 | *get E102* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9  00 3E 00 0000000000000000 | *return = 8 x 00* |

Check that the BDUT rejects writing 8 byte object with 9 bytes data (data too long):

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7 003E E102 010203040506070809 | *set E102 (9 bytes)* |
| ► 32 4D4153544552 00 1000 AFFE 6D 02 C9  00 3E FF | *error (length)* |

Check that the BDUT rejects writing 8 byte object with 7 bytes data (data too short):

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7  00 3E E102 01020304050607 | set E102 (7 bytes) |
| ► 32 4D4153544552 00 1000 AFFE 6F 02 C9  00 3E FF | *error (length)* |

Check that the value of the 8 byte object did not change:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 3E E1 02 | *get E102* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9  00 3E 00 0000000000000000 | *return = 8 x 00* |

## 6.2 ObjectLink (PID_PB_OBJLINK )

The Property – function “ObjectLink” (Property ID = PID_PB_OBJLINK) is tested. This one is used for inserting and removing links. In case of success the function returns with ReturnCode 00h, on errors FFh or 0xFE (=invalid object).

Preparation: clear association table.

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 4D4153544552 1000 E000 | *SetLink (MASTER,1000,E000)* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 4D4153544552 1100 E100 | *SetLink (MASTER,1100,E100)* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 4D4153544552 1101 E101 | *SetLink (MASTER,1101,E101)* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 4D4153544552 1200 E200 | *SetLink (MASTER,1200,E200)* |
| ► 32 4D4153544552 00 1000 AFFE 6B 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 4D4153544552 1201 E201 | *SetLink (MASTER,1201,E201)* |
| ► 32 4D4153544552 00 1000 AFFE 6D 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 112233445566 1000 E000 | *add Link* |
| ► 32 4D4153544552 00 1000 AFFE 6F 02 C9 00 3F FF | *FFh => error (full)* |

Delete one link and check if it is free:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 01 00 4D4153544552 1201 E201 | *delete Link 4* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9 00 3F 00 | *return = ok* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 3F 00 04 | *read Link 4* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9 00 3F FF | *FFh => free* |

Check that another link can be entered:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 112233445566 1000 E000 | *add Link* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9 00 3F 00 | *return = ok* |

Delete this link again:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F01004D 4153544552 1000 E000 | *delete Link 0* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9 00 3F 00 | *return = ok* |

Check if an error is returned on trying to insert a link to an invalid object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 00 00 112233445566 1000 ABCD | *invalid insertion* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9 00 3F FE | *0xFE => invalid object* |

Check if an error is returned on trying to delete a link from an invalid object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 01 00 112233445566 1000 ABCD | *invalid deletion* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9 00 3F FE | *0xFE => invalid object* |

Check if ok is returned on trying to delete a non existing link from a valid object:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02C7 00 3F 01 00 112233445566 7777 E000 | *(Link not existing)* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9 00 3F 00 | *return = ok* |

## 6.3 SetGetParameter (PID_PB_PARAMETERS)

The Property – function "Parameters" (Property ID = PID_PB_PARAMETERS) is tested. It is used for reading / writing parameters of a selected KNX-Easy channel.

Read existing parameter in existing channel:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 41 02 01 | *read Ch 2 / Par 1* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9 00 41 00 0000 | *return val = 0000h* |

Read non-existing parameter in existing channel:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 41 02 02 | *read Ch 2 / Par 2* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9 00 41 FF | *FFh => error* |

Read non-existing parameter in non-existing channel:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 41 03 01 | *read Ch 3 / Par 2* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9 00 41 FF | *FFh => error* |

Write existing parameter in existing channel, value too long:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7 00 41 02 01 01 02 | *write Ch 2, Par 1 (too long)* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9 00 41 FF | *FFh => error* |

Write existing parameter in existing channel, correct length:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C7 00 41 02 01 01 | *write Ch 2, Par 1 ok (01h)* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9 00 41 00 | *return code = 00h (ok)* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 41 02 01 | *read Ch 2, Par 1* |
| ► 32 4D4153544552 00 1000 AFFE 6B 02 C9 00 41 00 00 01 | *return val = 01h* |

## 6.4 GetObjectAddress (PID_PB_OBJADDR )

The Property – function "Object Address" (Property ID = PID_PB_OBJADDR) is tested. It is used for reading the channels and objects and their addresses in the active configuration of the device.

Read objects until the return code indicates the end of the object list:

| | |
|---|---|
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 00 | *read with iterator 0* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9 00 42 00 00 00 0123 00 E000 | *CH: 0 / 0123h, OBJ: 0 / E000* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 01 | *read with iterator 1* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9 00 42 00 01 00 0123 01 E001 | *CH: 0 / 0123h, OBJ: 1 / E001* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 02 | *read with iterator 2* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9 00 42 00 02 01 0201 00 E100 | *CH: 1 / 0201h, OBJ: 0 / E100* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 03 | *read with iterator 3* |
| ► 32 4D4153544552 00 1000 AFFE 67 02 C9 00 42 00 03 01 0201 01 E101 | *CH: 1 / 0201h, OBJ: 1 / E101* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 04 | *read with iterator 4* |
| ► 32 4D4153544552 00 1000 AFFE 69 02 C9 00 42 00 04 01 0201 02 E102 | *CH: 1 / 0201h, OBJ: 2 / E102* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 05 | *read with iterator 5* |
| ► 32 4D4153544552 00 1000 AFFE 6B 02 C9 00 42 00 05 01 0201 03 E103 | *CH: 1 / 0201h, OBJ: 3 / E103* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8 00 42 00 06 | *read with iterator 6* |

| | |
|---|---|
| ► 32 4D4153544552 00 1000 AFFE 6D 02 C9  00 42 00 06 02 0401 00 E200 | *CH: 2 / 0401h, OBJ: 0 / E200* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 42 00 07 | *read with iterator 7* |
| ► 32 4D4153544552 00 1000 AFFE 6F 02 C9  00 42 00 07 02 0401 01 E201 | *CH: 2 / 0401h, OBJ: 1 / E201* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 42 00 08 | *read with iterator 8* |
| ► 32 4D4153544552 00 1000 AFFE 61 02 C9  00 42 00 08 02 0401 02 E202 | *CH: 2 / 0401h, OBJ: 2 / E202* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 42 00 09 | *read with iterator 9* |
| ► 32 4D4153544552 00 1000 AFFE 63 02 C9  00 42 00 09 02 0401 03 E203 | *CH: 2 / 0401h, OBJ: 3 / E203* |
| ◄ 00 4D4153544552 00 AFFE 1000 61 02 C8  00 42 00 0A | *read with iterator 10* |
| ► 32 4D4153544552 00 1000 AFFE 65 02 C9  00 42 FF | *FF => no more Ojects* |

| | |
|---|---|
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 42 00 00 | *FuncPropCmd (iterator 0)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 63 02 C9 00 42 FF | *FF => not allowed* |
| ◄ 00 4D 41 53 54 45 52 00 AF FE 10 00 61 02 C7 00 42 00 0A | *FuncPropCmd (iterator 10)* |
| ► 32 4D 41 53 54 45 52 00 10 00 AF FE 65 02 C9 00 42 FF | *FF => not allowed* |

# 7 Multi Easy Configuration Tests

## 7.1 Generalities

### 7.1.1 Channel configuration for RF Multi Easy tests

#### 7.1.1.1 Channel configuration for RF Multi Easy Controler tests

| | | | | |
|---|---|---|---|---|
| **Description** | Light Actuator Scene | | | |
| **Channel-Index** | 1 | | | |
| **ChannelCode** | 0x0103 | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | 0001 | I | 1 Bit | 0x01 |
| | 0002 | I | 1 Bit | 0x03 |
| | 0003 | I | 2 Bits | 0x04 |
| | 0004 | I | 1 Byte | 0x08 |
| | 0005 | O | 1 Bit | 0x02 (Info) |
| **Parameters** | (none) | | | |

#### 7.1.1.2 Channel configuration for Push-Button tests

The above Light Actuator Scene channel and the following channel are used for the Push-Button tests.

| | | | | |
|---|---|---|---|---|
| **Description** | Generic Push-Button ½ info 5 | | | |
| **Channel-Index** | 1 | | | |
| **ChannelCode** | 0x0316 | | | |
| **Datapoints** | **Handle** | **Flags** | **Length** | **Conn.-Codes** |
| | 0001 | I | 1 Bit | 0x02 (Info) |
| | 0002 | O | 1 Bit | 0x01 |
| | 0003 | O | 1 Bit | 0x0B |
| | 0004 | O | 2 Bits | 0x04 |
| | 0005 | O | 4 Bits | 0x05 |
| | 0006 | O | 1 Byte | 0x08 |
| **Parameters** | (none) | | | |

### 7.1.2 Frame description

The tool used for RF test is EITT4 together with a cEMI interface able to send and receive RF frames.

The IN and OUT frames listed in this RF Multi section are taken from the EITT4 trace window.

The bytes displayed between square brackets represent the cEMI additionnal information. This information is defined in the document AN151.

FA means Fast Acknowledge in the comment of the scripts.

Example :

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 03  D5 02 01 10 01
```

The cEMI additional info ([08 04 01 00 00 01]) gives the following information:

- 08: KNX RF Multi additional information
- 04 : length = 4 bytes
- 01 : emission frequency = F1 ready
- 00 : fast and slow call channel = F1, S1
- 00 : no fast ack management

- 01 : reception frequency = F1 ready

## 7.2 Easy Controler

### 7.2.1 RF Multi Interface object existence

The existence of the KNX RF Multi object is checked by  reading the object type of the interface object 2.

Check that the BDUT returns the value 0x13 when reading the PID_OBJECT_TYPE on object index 02.

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 03  D5 02 01 10 01  :
PropertyValueRead(Obj=02, Prop=01, Count=1, Start=001)
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 07 03  D6 02 01 10 01 00 13  :
PropertyValueResponse(Obj=02, Prop=01, Count=1, Start=001, Data=00 13 )
```

### 7.2.2 RF Multi Type (PID_RF_MULTI_TYPE)

The Property “RF Multi Type” (Property ID = PID_RF_MULTI_TYPE) is tested, which can be used for configuring the KNX RF multi device as “KNX RF Ready” or as “KNX RF Multi” type.

The reading and writing of this property shall be correct.

Reading the value of KNX RF type => shall be Ready (0x00) (by default as set in the factory state)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 03  D5 02 33 10 01  :
PropertyValueRead(Obj=02, Prop=33, Count=1, Start=001)
```

```
►[RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 00  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
```

Writing the value of KNX RF type to Multi (0x01)

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 01  : PropertyValueWrite(Obj=02,
Prop=33, Count=1, Start=001, Data=01 )
```

```
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 01  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=01 )
```

Reading the value of KNX RF type => shall be Multi (0x01)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 05 03  D5 02 33 10 01  : PropertyValueRead(Obj=02, Prop=33,
Count=1, Start=001)
```

```
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 01  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=01 )
```

Writing the value of KNX RF type to Ready (0x00)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 00  : PropertyValueWrite(Obj=02,
Prop=33, Count=1, Start=001, Data=00 )
```

```
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 00  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
```

Reading the value of KNX RF type => shall be Ready (0x00)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 05 03  D5 02 33 10 01  : PropertyValueRead(Obj=02, Prop=33,
Count=1, Start=001)
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 00  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
```

### 7.2.3 RF Multi  Physical Features (PID_RF_MULTI_PHYSICAL_FEATURES)

Set the DUT physical feature to all -> 0x1F

Reading the value of flags of KNX RF Multi device => TxFx and RxFx

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 03  D5 02 34 10 01  : PropertyValueRead(Obj=02, Prop=34,
Count=1, Start=001)
```

```
 ► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 34 10 01 05  :
PropertyValueResponse(Obj=02, Prop=34, Count=1, Start=001, Data=05 )
```

Writing the value of flags of KNX RF Multi device => Error

◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 06 03  D7 02 34 10 01 00  : PropertyValueWrite(Obj=02, Prop=34, Count=1, Start=001, Data=00 )

► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 05 03  D6 02 34 00 00  : PropertyValueResponse(Obj=02, Prop=34, Count=0, Start=000, Data=)

Reading the value of flags of KNX RF Multi device => TxFx and RxFx

◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 05 03  D5 02 34 10 01  : PropertyValueRead(Obj=02, Prop=34, Count=1, Start=001)

► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 34 10 01 05  : PropertyValueResponse(Obj=02, Prop=34, Count=1, Start=001, Data=05 )

### 7.2.4 RF Multi Call Channel (PID_RF_MULTI_CALL_CHANNEL)

Preparation: Set the BDUT physical features to all features (value 0x1F).

Functional test : Add first a link on output datapoint (in transmission mode)

A RF Multi link is added on Fast frequency without FA  (GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x01, FA = 0x00)

◄ [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 02 00 05 01 00

► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  : *OK*

A RF Multi link is added on Slow frequency without FA  (GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x02, FA = 0x00)

◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 02 00 05 02 00

► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  : *OK*

Reading the value of Call Channel by default : F1 and S1

◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 03  D5 02 35 10 01  : *Read*

► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 00  : *Response=00*

Writing the value of Call Channel => F1 and S1

◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 06 03  D7 02 35 10 01 00  : *Write 00*

► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 00  : *Response=00*

Reading the value of Call Channel => F1 and S1

◄ [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 09] 3C 60 AFFE 05FF 05 03  D5 02 35 10 01  : PropertyValueRead(Obj=02, Prop=35, Count=1, Start=001)

► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 00  : PropertyValueResponse(Obj=02, Prop=35, Count=1, Start=001, Data=00 )

Make the interaction on the product to send the datapoint

Check that DUT transmits RF frame (ON) on the F1 channel

► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 02] 0C A0 05FF 0002 01 00  81  :

Check that DUT transmits RF frame (ON) on the S1 channel

► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 05] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)

Writing the value of Call Channel => F2 and S2

◄ [RF: 02 SN: 00090040FF03 2][08 04 00 00 00 09] 3C 60 AFFE 05FF 06 03  D7 02 35 10 01 05  : PropertyValueWrite(Obj=02, Prop=35, Count=1, Start=001, Data=05 )

```
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 05  :
PropertyValueResponse(Obj=02, Prop=35, Count=1, Start=001, Data=05 )
```

Reading the value of Call Channel => F2 and S2

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 03  D5 02 35 10 01  : PropertyValueRead(Obj=02, Prop=35,
Count=1, Start=001)
```

```
► [RF: 02 SN: 00090040FF03 0] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 05  : PropertyValueResponse(Obj=02,
Prop=35, Count=1, Start=001, Data=05 )
```

Make the interaction on the product to send the datapoint

Check that DUT transmits RF frame (ON) on the F2 channel

```
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 03] 0C A0 05FF 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

Check that DUT transmits RF frame (ON) on the S2 channel

```
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 06] 0C A0 05FF 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

Writing the value of Call Channel => F3 and S1

```
◄ [RF: 02 SN: 00090040FF03 4][08 04 00 00 00 09] 3C 60 AFFE 05FF 06 03  D7 02 35 10 01 08  :
PropertyValueWrite(Obj=02, Prop=35, Count=1, Start=001, Data=08 )
```

```
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 08  :
PropertyValueResponse(Obj=02, Prop=35, Count=1, Start=001, Data=08 )
```

Reading the value of Call Channel => F3 and S1

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 05 03  D5 02 35 10 01  : PropertyValueRead(Obj=02, Prop=35,
Count=1, Start=001)
```

```
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 35 10 01 08  :
PropertyValueResponse(Obj=02, Prop=35, Count=1, Start=001, Data=08 )
```

Make the interaction on the product to send the datapoint

Check that DUT transmits RF frame (ON) on the F2 channel

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 04] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Check that DUT transmits RF frame (ON) on the S1 channel

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 05] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

### 7.2.5 RF Multi Object Link (PID_RF_MULTI_OBJECT_LINK)

The following tables give the compatibility between features of the device and features requested by the object link function property when adding a link on an output datapoint of a device.

F1r = F1 ready ; Fx = Fast frequency ; Sx = Slow frequency; FA = Fast Ack (Physical Acknowledge)

🟥 No link possible (FDh = physical error)

🟦 Link ok

| | | | Link addition | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Object_ link | RF_Multi_Object_Link | | | | | | | | | | | | | |
| | | | | Fx | Sx | F1r | Sx, Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | Fx (FA) | Sx (FA) | F1r (FA) | Sx(FA), Fx(FA) | F1r(FA Fx(FA) | F1r(FA), Sx(FA | F1r(FA), Fx(FA), Sx(FA) |
| output datapoint | RF Ready | | F1r | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | RF Multi Tx features | - | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | TX Sx | 🟥 | 🟥 | Sx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | TX Fx | F1r | Fx | 🟥 | F1r | 🟥 | F1r, Fx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | TX Sx, TX Fx | F1r | Fx | Sx | F1r | Sx, Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, TX Sx | 🟥 | 🟥 | Sx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | Sx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, TX Fx | F1r | Fx | 🟥 | F1r | 🟥 | F1r, Fx | 🟥 | 🟥 | Fx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, TX Sx, TX Fx | F1r | Fx | Sx | F1r | Sx,Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | Fx (FA) | Sx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |

The following tables give the compatibility between features of the device and features requested by the object link function property when adding a link on an input datapoint of a device.

| | | | Link addition | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Object_ link | RF_Multi_Object_Link | | | | | | | | | | | | | |
| | | | | Fx | Sx | F1r | Sx, Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | Fx (FA) | Sx (FA) | F1r (FA) | Sx(FA), Fx(FA) | F1r(FA Fx(FA) | F1r(FA), Sx(FA | F1r(FA), Fx(FA), Sx(FA) |
| input datapoint | RF Ready | | F1r | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | RF Multi Rx features | - | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | RX Sx | 🟥 | 🟥 | Sx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | RX Fx | F1r | Fx | 🟥 | F1r | 🟥 | F1r, Fx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | RX Sx, RX Fx | F1r | Fx | Sx | F1r | Sx, Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, RX Sx | 🟥 | 🟥 | Sx | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | Sx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, RX Fx | F1r | Fx | 🟥 | F1r | 🟥 | F1r, Fx | 🟥 | 🟥 | Fx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |
| | | FA, RX Sx, RX Fx | F1r | Fx | Sx | F1r | Sx,Fx | F1r, Fx | F1r, Sx | F1r, Sx, Fx | Fx (FA) | Sx (FA) | 🟥 | 🟥 | 🟥 | 🟥 | 🟥 |

Add / Remove link

Pre-conditions :

DUT => reset to factory

DUT Serial Number = 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

PID_RF_MULTI_CALL_CHANNEL is set with F1 and S1 capacities

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

DUT has a link table of size 5.

Purpose of the test :

Add again a new link on an input datapoint.

Check the returned value : 00 (OK)

Read the link using the iterator.

Check that the link is present and the bytes 22 and 23.

PID_RF_MULTI_OBJETC_LINK => Set RF Multi link on OnOff datapoint on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 00] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Read 1 link

Property Function command Read( Object Link Read iterator 0) --> OnOff link

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 01
00 )
```

Property Function command Read( Object Link Read iterator 1) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Check that DUT correctly understand the command for switching ON

```
◄ [RF: 00 SN: 112233445566 0][08 04 02 00 00 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

REMOVE A SIMPLE LINK

Purpose of the test :

Remove the link on an input datapoint.

Check the returned value : 00 (OK)

Read the link using the iterator.

Check that the link is no more present and the bytes 22 and 23.

PID_RF_MULTI_OBJETC_LINK => Delete RF Multi link on OnOff datapoint on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 00] 3C 60 AFFE 05FF 11 02  C7 02 36 02 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=02 00 11 22 33 44 55 66 00 02 00 01 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Check that DUT do not understand any more the command for switching OFF

```
◄ [RF: 00 SN: 112233445566 1][08 04 02 FF 00 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

#### 7.2.5.1 Add link on non existing datapoint

Pre-conditions :

DUT => reset to factory

DUT serial number = 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

PID_RF_MULTI_CALL_CHANNEL is set with F1 and S1 capacities

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

DUT has a link table of size 5.

Purpose of the test :

Add a link on a non existing object handle.

Check the returned value : 0xFE (ERR_OBJECT)

Read the link using the iterator.

Check that the link is not present

PID_RF_MULTI_OBJETC_LINK => Set RF Multi link on a non existing object handle on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0012, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 00] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00
02 00 12 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 12 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FE  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FE, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

#### 7.2.5.2 Add a link when the table is already full

Pre-conditions :

DUT => reset to factory

DUT serial number = 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

DUT has a link table of size 3.

Purpose of the test :

Fill the link table (3 entries) with 3 links.

Add a link on an input datapoint

Check the returned value : 0xFF (ERR_TABLE FULL)

Read the link using the iterator.

Check that the link is not present

Fill the link table with three new links

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on an existing object handle on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on an existing object handle on Fast without FA (SN =0x112233445577, GA = 0x0003, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 77 00 03 00 01 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 77 00 03 00 01 01 00 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on an existing object handle on Fast without FA (SN =0x112233445588, GA = 0x0004, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 88 00 04 00 01 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 88 00 04 00 01 01 00 )
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Add another link that must fail

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on an existing object handle on Fast without FA (SN =0x112233445599, GA = 0x0005, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 99 00 05 00 01 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 99 00 05 00 01 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Read 3 links

Property Function command Read( Object Link Read iterator 0) --> Link with GA 0x0002

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00 02 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 01 00 )
```

Property Function command Read( Object Link Read iterator 1) --> Link with GA 0x0003

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 77 00 03 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 77 00 03 00 01 01 00 )
```

Property Function command Read( Object Link Read iterator 2) --> Link with GA 0x0004

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 88 00 04 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 88 00 04 00 01 01 00 )
```

Property Function command Read( Object Link Read iterator 3) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 05 02  C8 02 36 00 03  : FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.2.5.3 Add link with incompatible features

#### 7.2.5.3.1 Physical acknowledge incompatibility

Pre-conditions :

DUT => reset to factory

DUT SERIAL Number = 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities except FA : FA:no, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x0F

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

Description of the test :

Add a link on an input datapoint with Rx Sx with Fast Ack

Check the returned value : 0xFD (ERR_RF_PHYSICAL)

Read the link using the iterator.

Check that the link is not present.

Add RF Multi link on an existing object handle on Slow frequency with FA  (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x02, FA = 0x40)

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00 02 00 01 02 40  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 02 40 )
```

```
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 36 FD  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FD, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.2.5.3.2 Fast frequency transmission incompatibility

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities except TX Fx : FA:yes, TX Fx:no, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1B

DUT contains a light actuator channel scene containing the following datapoints : OnOff (1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

Description of the test :

Add a link on an output datapoint on Fast, without FA

Check the returned value : 0xFD (ERR_RF_PHYSICAL)

Read the link using the iterator.

Check that the link is not present

PID_RF_MULTI_OBJETC_LINK => Set RF Multi link on an existing object handle on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 11 22 33 44 55 66 00 02 00 05 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 11 22 33 44 55 66 00 02 00 05 01 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 36 FD  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FD, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.2.5.3.3 Slow frequency reception incompatibility

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities except RX Sx : ACK:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : no ==> 0x1D

DUT contains a light actuator channel scene containing the following datapoints : OnOff (1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

Description of the test :

Add a link on an INPUT datapoint on SLOW, without FA

Check the returned value : 0xFD (ERR_RF_PHYSICAL)

Read the link using the iterator.

Check that the link is not present

Add RF Multi link on an existing object handle on slow without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0002, Physical features = 0x02, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00 02 00 02 02 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 02 02 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 36 FD  : FctPropState_Rsp(ObjIdx=02, PropId=36,
RetCode=FD, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36,
RetCode=FF, Data=)
```

#### 7.2.5.3.4 Fast and Slow frequency transmission with fast physical acknowledge

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities except RX Sx : ACK:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : no ==> 0x1D

DUT contains a light actuator channel scene containing the following datapoints : OnOff (1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

Description of the test :

Add a link on an OUTPUT datapoint on SLOW, AND FAST with FA

Check the returned value : 0xFD (ERR_RF_PHYSICAL)

Read the link using the iterator.

Check that the link is not present

Add RF Multi link on an existing object handle on slow and fast with FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x03, FA = 0x40)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 11 22 33 44 55 66 00 02 00 05 03 40  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 11 22 33 44 55 66 00 02 00 05 03 40 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 36 FD  : FctPropState_Rsp(ObjIdx=02, PropId=36,
RetCode=FD, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36,
RetCode=FF, Data=)
```

#### 7.2.5.3.5 Attempt to modify ack slot number of an existing link with physical acknowledge

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities except RX Sx : ACK:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : no ==> 0x1D

DUT contains a light actuator channel scene containing the following datapoints : OnOff (1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

Description of the test :

Add a link on an OUTPUT datapoint on SLOW with FA (ack slot number 3)

Check the returned value : 0x00 (OK)

Add a link on an OUTPUT datapoint on SLOW with FA (ack slot number 4)

Check the returned value : 0xFD (ERR_RF_PHYSICAL)

Read the link using the iterator.

Check that the second link is not present

PID_RF_MULTI_OBJETC_LINK => Set RF Multi link on an existing object handle on Slow with FA, ack slot number 3 (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x02, FA = 0x42)

```
◄ [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 11 22 33 44 55 66 00
02 00 05 02 42  : FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 11 22 33 44 55 66 00 02 00 05 02 42 )
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJETC_LINK => Set RF Multi link on an existing object handle on Slow with FA , ack slot number 4 (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0005, Physical features = 0x02, FA = 0x43)

```
◄ [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 11 22 33 44 55 66 00
02 00 05 02 43  : FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 11 22 33 44 55 66 00 02 00 05 02 43 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FD  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FD, Data=)
```

Read 1 link

Property Function command Read( Object Link Read iterator 1) --> Link in Sx with Fast ack (ack slot number = 3)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
02 00 05 02 42  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 02 00 05 02
42 )
```

Property Function command Read( Object Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

#### 7.2.5.4 Remove non existing link

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

DUT has a link table of size 5.

Purpose of the test :

Remove a non exiting link on an input datapoint.

Check the returned value : 00 (OK)

Read the link using the iterator.

Check that the link is no more present and the bytes 22 and 23.

PID_RF_MULTI_OBJETC_LINK => Delete a non existing RF Multi link on OnOff datapoint on Fast without FA  (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 11 02  C7 02 36 02 00 11 22 33 44 55 66 00 02 00 01 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=02 00 11 22 33 44 55 66 00 02 00 01 01 00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.2.5.5 Remove link on non existing datapoint

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

DUT contains a light actuator channel scene containing the following datapoints : OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

Purpose of the test :

Remove a link on a non existing datapoint.

Check the returned value : 0xFE (ERR_OBJECT)

Read the link using the iterator.

Check that the link is no more present and the bytes 22 and 23.

PID_RF_MULTI_OBJETC_LINK => Delete RF Multi link on non existing datapoint on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0012, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 11 02  C7 02 36 02 00 11 22 33 44 55 66 00 02 00 12 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=02 00 11 22 33 44 55 66 00 02 00 12 01 00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FE  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FE, Data=)
```

Read 0 link

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.2.5.6 Read the number of links in the device

Pre-conditions :

DUT => reset to factory

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

DUT contains a light actuator channel scene containing the following datapoints: OnOff, TimedStartStop, Forced, Scene Number, InfoOnOff

Purpose of the test :

Read the numbers of links and check them.

Add a link on an input datapoint.

Read the numbers of links and check them.

Remove the link.

Read the numbers of links and check them.

Property Function command Read (with code = 01) --> Read the numbers of links in the device and check them

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 01 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 01 00 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00 00 05 00 05  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 05 00 05)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on an input datapoint on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 00] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 01 00 )
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Property Function command Read (with code = 01) --> Read the numbers of links in the device and check them

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 01 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 01 00 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00 00 04 00 05  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 04 00 05)
```

PID_RF_MULTI_OBJETC_LINK => Delete RF Multi link on input datapoint on Fast without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 11 02  C7 02 36 02 00 11 22 33 44 55 66 00 02 00 01 01 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=02 00 11 22 33 44 55 66 00 02 00 01 01 00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Property Function command Read (with code = 01) --> Read the numbers of links in the device and check them

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 01 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 01 00 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00 00 05 00 05  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 05 00 05)
```

### 7.2.6 RF Multi Extended Group Address Repeated (PID_RF_MULTI_EXT_GA_REPEATED)

#### 7.2.6.1 Add and remove an extended group address in the repetition table

Pre-conditions :

DUT => Reset to factory

KNX RF type = Ready

Purpose of the test :

Add an extended group address in the repetition table

Check the returned value : 0xFF (OK)

Read back the group address using the iterator.

Check that the extended group address is in the table

Remove the extended group address from the repetition table

Check the return code : 00 (ok)

Read back the repetition table using the iterator

Check that the extended group address is no more in the repetition table

PID_RF_MULTI_EXT_GA_REPEATED => Add an extended group address in the repetition table (SN =0x112233445566, GA = 0x4321)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 00 00 00 00] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 66 43 21  : FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 66 43 21 )
 ► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=)
```

Read the repetition table (1 entry)

Property Function command Read( Object Link Read iterator 0) --> first extended group address

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02, PropId=37, 00 00 )
► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 66 43 21  : FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 66 43 21 )
```

Property Function command Read( Object Link Read iterator 1) --> Shall answer: no more extended group address in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 05 02  C8 02 37 00 01  : FctPropState_Read(ObjIdx=02, PropId=37, 00 01 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=FF, Data=)
```

REMOVE THE EXTENDED GROUP ADDRESS

PID_RF_MULTI_EXT_GA_REPEATED => Remove the added extended group address in the repetition table (SN =0x112233445566, GA = 0x4321)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 0D 02  C7 02 37 01 00 11 22 33 44 55 66 43 21  : FctPropCmd(ObjIdx=02, PropId=37, Data=01 00 11 22 33 44 55 66 43 21 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=)
```

Read the repetition table : 0 entry

Property Function command Read( Object Link Read iterator 0) --> Shall answer: no more extended group address in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02, PropId=37, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=FF, Data=)
```

#### 7.2.6.2 Add and remove a serial number in the repetition table

DUT => Reset to factory

KNX RF type = Ready

Purpose of the test :

Add a serial number in the repetition table

Check the returned value : 0xFF (OK)

Read back the serial number using the iterator.

Check that the serial number is in the table

Remove the serial number from the repetition table

Check the return code : 00 (ok)

Read back the serial number using the iterator

Check that the serial number is no more in the repetition table

PID_RF_MULTI_EXT_GA_REPEATED => Add a serial number in the repetition table  (SN =0x112233445577, GA = 0x0000)

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 77 00 00  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 77 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

Property Function command Read( Object Link Read iterator 0) --> first serial number

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 00 )
 ► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 77 00 00  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 77 00 00 )
```

Property Function command Read( Object Link Read iterator 1) --> Shall answer: no more serial number in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 05 02  C8 02 37 00 01  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 01 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

REMOVE THE EXTENDED GROUP ADDRESS

PID_RF_MULTI_EXT_GA_REPEATED => Remove the added serial number in the repetition table (SN =0x112233445577, GA = 0x0000)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 0D 02  C7 02 37 01 00 11 22 33 44 55 77 00 00  :
FctPropCmd(ObjIdx=02, PropId=37, Data=01 00 11 22 33 44 55 77 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

Read the repetition table  : 0 entry

Property Function command Read( Object Link Read iterator 1) --> Shall answer: no more serial number in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

### 7.2.6.3 Add / remove several elements in the repetition table

DUT => Reset to factory

The SIZE of the repetition table is set to 5.

Purpose of the test :

Add 5 elements (extended group address or serial number) in the repetition table

Check the returned value : 0x00 (OK)

Add another element to the repetition table and check that the returned code is 0xFF (ERR_TABLEFULL)

Read back the 5 elements using the iterator.

Check that the last added element is not in the table

Remove all the elements from the repetition table

Check the return code : 00 (ok)

Read back the elements using the iterator

Check that the repetition table is empty

PID_RF_MULTI_EXT_GA_REPEATED => Add a serial number in the repetition table  (SN =0x112233445577, GA = 0x0000)

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 00] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 77 00
00  : FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 77 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

PID_RF_MULTI_EXT_GA_REPEATED => Add an extended group address in the repetition table  (SN =0x112233445566, GA = 0x4321)

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 66 43 21  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 66 43 21 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

PID_RF_MULTI_EXT_GA_REPEATED => Add a serial number in the repetition table  (SN =0x112233445588, GA = 0x0000)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 88 00 00  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 88 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

PID_RF_MULTI_EXT_GA_REPEATED => Add an extended group address in the repetition table  (SN =0x112233445544, GA = 0x5555)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 44 55 55  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 44 55 55 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

PID_RF_MULTI_EXT_GA_REPEATED => Add a serial number in the repetition table  (SN =0x112233445599, GA = 0x0000)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 11 22 33 44 55 99 00 00  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 11 22 33 44 55 99 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 00  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=00, Data=)
```

PID_RF_MULTI_EXT_GA_REPEATED => Add an extended group address in the repetition table  (SN =0x010203040506, GA = 0x1234)

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 0D 02  C7 02 37 00 00 01 02 03 04 05 06 12 34  :
FctPropCmd(ObjIdx=02, PropId=37, Data=00 00 01 02 03 04 05 06 12 34 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

Read the repetition table (5 entry)

Property Function command Read( Object Link Read iterator 0) --> first serial number

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 00 )
► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 77 00 00  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 77 00 00 )
```

Property Function command Read( Object Link Read iterator 1) --> second entry of the repetition table

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 37 00 01  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 01 )
► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 66 43 21  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 66 43 21 )
```

Property Function command Read( Object Link Read iterator 2) --> 3rd entry of the repetition table

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 05 02  C8 02 37 00 02  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 02 )
► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 88 00 00  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 88 00 00 )
```

Property Function command Read( Object Link Read iterator 3) --> 4th entry of the repetition table

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 05 02  C8 02 37 00 03  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 03 )
 ► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 44 55 55  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 44 55 55 )
```

Property Function command Read( Object Link Read iterator 4) --> 5th entry of the repetition table

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 05 02  C8 02 37 00 04  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 04 )
► [RF: 02 SN: 00090040FF03 4] 1C 60 05FF AFFE 0D 02  C9 02 37 00 00 11 22 33 44 55 99 00 00  :
FctPropState_Rsp(ObjIdx=02, PropId=37, RetCode=00, Data=00 11 22 33 44 55 99 00 00 )
```

Property Function command Read( Object Link Read iterator 5) --> Shall answer: no more serial number in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 37 00 05  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 05 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

REMOVE THE EXTENDED GROUP ADDRESS

PID_RF_MULTI_EXT_GA_REPEATED => Remove all elements from the repetition table

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 0F 02  C7 02 37 02 00 00 00 00 00 00 00 00 00 00 00  :
FctPropCmd(ObjIdx=02, PropId=37, Data=02 00 00 00 00 00 00 00 00 00 00 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

Read the repetition table  : 0 entry

Property Function command Read( Object Link Read iterator 1) --> Shall answer: no more serial number in the repetition table

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 05 02  C8 02 37 00 00  : FctPropState_Read(ObjIdx=02,
PropId=37, 00 00 )
► [RF: 02 SN: 00090040FF03 1] 1C 60 05FF AFFE 04 02  C9 02 37 FF  : FctPropState_Rsp(ObjIdx=02, PropId=37,
RetCode=FF, Data=)
```

## 7.2.7 Compatibility between RF 1.1, RF1.R, RF1.M

### 7.2.7.1 Compatibility between RF 1.1 and RF1.R

Pre-conditions :

DUT => reset to factory

Serial Number of DUT : 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

PID_RF_MULTI_CALL_CHANNEL is set with F1 and S1 capacities

DUT contains a light actuator channel scene containing the following datapoints : OnOff(1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

DUT has a link table of size 5.

=================== TEST SCRIPT ====================

Purpose of the test :

Set the RF Multi device in RF Ready mode.

Add one link on input datapoint using the PID_OBJECT_LINK function property command : check that the result is OK (0x00).

Add one link on output datapoint using the PID_OBJECT_LINK function property command : check that the result is OK (0x00).

Check that an emitted frame on RF 1.1 is correctly understood by DUT (in RF Ready mode).

Check that the DUT sends the correct frame using RF Ready format.

Add one link on input datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Check that an emitted frame on RF 1.1 is correctly understood by DUT (in RF Ready mode).

Check that the DUT sends the correct frame using RF Ready format.

Writing the value of KNX RF type => Ready

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 FF 00 01] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 00  :
PropertyValueWrite(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 00  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
```

PID_OBJECT_LINK => Set RF link on OnOff datapoint  (SN =0x112233445566, GA = 0x0008, Obj Handle = 0x0001)

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 0F 02  C7 00 3F 00 00 11 22 33 44 55 66 00 08 00 01  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=00 00 11 22 33 44 55 66 00 08 00 01 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

PID_OBJECT_LINK => Set RF link on InfoOnOff datapoint  (SN =00090040ff03, GA = 0x0009, Obj Handle = 0x0005)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 0F 02  C7 00 3F 00 00 00 09 00 40 FF 03 00 09 00 05  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=00 00 00 09 00 40 FF 03 00 09 00 05 )
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

Check that DUT correctly understand the command for switching ON using RF 1.1 (Preamble with 1 ms = 16 number of sequences of "01" chips)

And correctly sends the InfoOnOff datapoint value

```
◄ [RF: 00 SN: 112233445566 0][08 04 01 FF 00 00  09 03 00 10 00] 2C E0 AFFE 0008 01 00  81  : DPT 1.xxx
Boolean (switch on)
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 0C E0 05FF 0009 01 00  81  : DPT 1.xxx Boolean (switch on)
```

remove links

PID_OBJECT_LINK => Set RF link on OnOff datapoint  (SN =0x112233445566, GA = 0x0008, Obj Handle = 0x0001)

```
◄ [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 3C 60 AFFE 05FF 0F 02  C7 00 3F 01 00 11 22 33 44 55 66 00
08 00 01  : FctPropCmd(ObjIdx=00, PropId=3F, Data=01 00 11 22 33 44 55 66 00 08 00 01 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 00 3F 00  : FctPropState_Rsp(ObjIdx=00, PropId=3F,
RetCode=00, Data=)
```

PID_OBJECT_LINK => Set RF link on InfoOnOff datapoint  (SN =00090040ff03, GA = 0x0009, Obj Handle = 0x0005)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 0F 02  C7 00 3F 01 00 00 09 00 40 FF 03 00 09 00 05  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=01 00 00 09 00 40 FF 03 00 09 00 05 )
```

```
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on OnOff datapoint on RF Ready Fast without FA  (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x04, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00
02 00 01 04 00  : FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 04 00 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on InfoOnOff datapoint on RF Ready Fast without FA  (SN =00090040ff03, GA = 0x0003, Obj Handle = 0x0005, Physical features = 0x04, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 03 00 05 04 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 00 40 FF 03 00 03 00 05 04 00 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Check that DUT correctly understand the command for switching ON using RF 1.1 (Preamble with 1 ms = 16 number of sequences of "01" chips)

And correctly sends the InfoOnOff datapoint value

```
◄ [RF: 00 SN: 112233445566 1][08 04 01 00 00 01  09 03 00 10 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx
Boolean (switch off)
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 0C E0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
```

#### 7.2.7.2 Compatibility between RF1.R and RF1.M

The tests would also apply for the other RFx.R and RFx.M devices according to vol. 3.2.5 clause 9.

Pre-conditions :

DUT => reset to factory

Serial Number of DUT : 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

PID_RF_MULTI_CALL_CHANNEL is set with F1 and S1 capacities

DUT contains a light actuator channel scene containing the following datapoints : OnOff(1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

DUT has a link table of size 5.

Purpose of the test :

Set the RF Multi device in RF Multi mode.

Add one link on input datapoint using the PID_OBJECT_LINK function property command : check that the result is OK (0x00).

Add one link on output datapoint using the PID_OBJECT_LINK function property command : check that the result is OK (0x00).

Check that an emitted frame on RF Ready is correctly understood by DUT.

Check that the DUT sends the correct frame using RF Ready format.

Add one link on input datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Check that an emitted frame on RF Ready is correctly understood by DUT.

Check that the DUT sends the correct frame using RF Ready format.

Writing the value of KNX RF type => Ready

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 FF 00 01] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 00  :
PropertyValueWrite(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 00  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=00 )
```

PID_OBJECT_LINK => Set RF link on OnOff datapoint  (SN =0x112233445566, GA = 0x0008, Obj Handle = 0x0001)

```
◄ [RF: 02 SN: 00090040FF03 1] 3C 60 AFFE 05FF 0F 02  C7 00 3F 00 00 11 22 33 44 55 66 00 08 00 01  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=00 00 11 22 33 44 55 66 00 08 00 01 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

PID_OBJECT_LINK => Set RF link on InfoOnOff datapoint  (SN =DUT SN, GA = 0x0009, Obj Handle = 0x0005)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 0F 02  C7 00 3F 00 00 00 09 11 22 33 44 00 09 00 05  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=00 00 00 09 11 22 33 44 00 09 00 05 )
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

Check that DUT correctly understand the command for switching ON using RF Ready

And correctly sends the InfoOnOff datapoint value

```
◄ [RF: 00 SN: 112233445566 0][08 04 01 FF 00 01] 2C E0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 0C E0 05FF 0009 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Remove the two links

PID_OBJECT_LINK => Set RF link on OnOff datapoint  (SN =0x112233445566, GA = 0x0008, Obj Handle = 0x0001)

```
◄ [RF: 02 SN: 00090040FF03 3][08 04 01 FF 00 01] 3C 60 AFFE 05FF 0F 02  C7 00 3F 01 00 11 22 33 44 55 66 00
08 00 01  : FctPropCmd(ObjIdx=00, PropId=3F, Data=01 00 11 22 33 44 55 66 00 08 00 01 )
► [RF: 02 SN: 00090040FF03 3] 1C 60 05FF AFFE 04 02  C9 00 3F 00  : FctPropState_Rsp(ObjIdx=00, PropId=3F,
RetCode=00, Data=)
```

PID_OBJECT_LINK => Set RF link on InfoOnOff datapoint  (SN =DUT SN, GA = 0x0009, Obj Handle = 0x0005)

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 0F 02  C7 00 3F 01 00 00 09 11 22 33 44 00 09 00 05  :
FctPropCmd(ObjIdx=00, PropId=3F, Data=01 00 00 09 11 22 33 44 00 09 00 05 )
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F 00  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=)
```

RF MULTI

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on OnOff datapoint on RF Ready Fast without FA  (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x04, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00 02 00 01 04 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 04 00 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on InfoOnOff datapoint on RF Ready Fast without FA  (SN =DUT SN, GA = 0x0003, Obj Handle = 0x0005, Physical features = 0x04, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 11 22 33 44 00 03 00 05 04 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 11 22 33 44 00 03 00 05 04 00 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Check that DUT correctly understand the command for switching ON using RF Ready.

And correctly sends the InfoOnOff datapoint value

```
◄ [RF: 00 SN: 112233445566 1][08 04 01 FF 00 07] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 0C E0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
```

#### 7.2.7.3 Compatibility between RF1.M and RF1.M

The tests would also apply for the other RFx.M and RFx.M devices according to vol. 3.2.5 clause 9.

Purpose of the test :

Set the RF Multi device in RF Multi mode.

Add one link on input datapoint in Slow mode using the PID_RF_MULTI_LINK function property command (with Sx bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint in Slow mode using the PID_RF_MULTI_LINK function property command (with Sx bit set in Physical Features) : check that the result is also ok (00).

Check that an emitted frame on Multi with Sx is correctly understood by DUT.

Check that the DUT sends the correct frame using RF Multi Sx format.

Writing the value of KNX RF type => Multi

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 01  : PropertyValueWrite(Obj=02,
Prop=33, Count=1, Start=001, Data=01 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 01  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=01 )
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on OnOff datapoint on Slow mode without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x02, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00 02 00 01 02 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 02 00 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on InfoOnOff datapoint on Slow mode without FA  (SN = 00090040ff03, GA = 0x0003, Obj Handle = 0x0005, Physical features = 0x02, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 03 00 05 02 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 00 40 FF 03 00 03 00 05 02 00 )
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Check that DUT correctly understand the command for switching ON using RF Sx

And correctly sends the InfoOnOff datapoint value using Sx

```
◄ [RF: 00 SN: 112233445566 0][08 04 05 00 00 05] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 00 05] 0C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
```

### 7.2.8 Configuration with mixed links

Pre-conditions :

DUT => reset to factory

Serial Number of DUT : 00090040ff03

PID_RF_MULTI_TYPE is set to RF Multi

PID_PHYSICAL_FEATURES is set with all capabilities : FA:yes, TX Fx:yes, TX Sx:yes, RX Fx : yes, RX Sx : yes ==> 0x1F

PID_RF_MULTI_CALL_CHANNEL is set with F1 and S1 capacities

DUT contains a light actuator channel scene containing the following datapoints : OnOff(1), TimedStartStop(2), Forced(3), Scene Number(4), InfoOnOff(5)

DUT has a link table of size 5.

Purpose of the test :

Set the RF Multi device in RF Multi mode.

Ready :

Add one link on input datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint using the PID_RF_MULTI_LINK function property command (with RF Ready bit set in Physical Features) : check that the result is also ok (00).

Check that an emitted frame on RF Ready is correctly understood by DUT.

Check that the DUT sends the correct frame using RF Ready format.

Multi Fast :

Add one link on input datapoint in Fast mode using the PID_RF_MULTI_LINK function property command (with Fx bit set in Physical Features) : check that the result is also ok (00).

Add one link on input datapoint in Fast mode using the PID_RF_MULTI_LINK function property command with FA slot n°10 (with Fx bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint in Fast mode using the PID_RF_MULTI_LINK function property command (with Fx bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint in Fast mode using the PID_RF_MULTI_LINK function property command with FA slot n° 10(with Fx bit set in Physical Features) : check that the result is also ok (00).

Check that emitted frame on Multi with Fx are correctly understood by DUT.

Check that the DUT sends the correct frame using RF Multi Fx format.

Multi Slow

Add one link on input datapoint in Slow mode using the PID_RF_MULTI_LINK function property command (with Sx bit set in Physical Features) : check that the result is also ok (00).

Add one link on input datapoint in Slow mode using the PID_RF_MULTI_LINK function property command with FA slot n° 10 (with Sx bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint in Slow mode using the PID_RF_MULTI_LINK function property command (with Sx bit set in Physical Features) : check that the result is also ok (00).

Add one link on output datapoint in Slow mode using the PID_RF_MULTI_LINK function property command with FA slot n° 10(with Sx bit set in Physical Features) : check that the result is also ok (00).

Check that emitted frame on Multi with Sx are correctly understood by DUT.

Check that the DUT sends the correct frame using RF Multi Sx format.

Writing the value of KNX RF type => Multi

```
◄ [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 00] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 01  :
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 01  :
```

Check that the DUT is able to do everything

Reading the value of flags of KNX RF Multi device =>1F

```
◄ [RF: 02 SN: 00090040FF03 5] 3C 60 AFFE 05FF 05 03  D5 02 34 10 01  :
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 34 10 01 1F  :
```

Add RF Multi link on OnOff datapoint on Fast, Slow and Ready mode without FA (SN =0x112233445566, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x07, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 66 00 02 00 01 07 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 66 00 02 00 01 07 00 )
```

```
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Add RF Multi link on InfoOnOff datapoint on Fast, Slow and Ready mode without FA  (SN =00090040ff03, GA = 0x0003, Obj Handle = 0x0005, Physical features = 0x07, FA = 0x00)

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 03 00 05 07 00  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 00 40 FF 03 00 03 00 05 07 00 )
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Check the links : read 2 links using iterators

Property Function command Read( Object Link Read iterator 0) --> OnOff link without fast ack

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 07 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 07
00 )
```

Property Function command Read( Object Link Read iterator 1) --> InfoOnOff link without fast ack

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 03 00
03 00 05 07 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 03 00 03 00 05
07 00 )
```

Property Function command Read( Object Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 4] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

WITHOUT FAST ACK

Check that DUT correctly understand the command for switching ON using RF Sx

And correctly sends the InfoOnOff datapoint value

F1 Ready frame is send at first.

F1 (multi) frame is then sent.

Last frame is Sx frame.

```
◄ [RF: 00 SN: 112233445566 0][08 04 05 00 00 09] 2C A0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 5][08 04 05 00 00 01] 0C E0 05FF 0003 01 00  81  :
► [RF: 02 SN: 00090040FF03 5][08 04 05 00 00 02] 0C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 5][08 04 05 00 00 05] 0C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Check that DUT correctly understand the command for switching OFF using RF Fx

And correctly sends the InfoOnOff datapoint value

F1 Ready frame is send at first

F1 (multi) frame is then sent

Last frame is Sx frame.

```
◄ [RF: 00 SN: 112233445566 1][08 04 02 00 00 09] 2C A0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 6][08 04 02 00 00 01] 0C E0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 6][08 04 02 00 00 02] 0C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 6][08 04 02 00 00 05] 0C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
```

Check that DUT correctly understand the command for switching ON using RF Ready

And correctly sends the InfoOnOff datapoint value

F1 Ready frame is send at first

F1 (multi) frame is then sent

Last frame is Sx frame.

```
◄ [RF: 00 SN: 112233445566 2][08 04 01 00 00 09] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 01] 0C E0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 02] 0C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 7][08 04 01 00 00 05] 0C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
```

WITH FAST ACK

Writing the value of KNX RF type => Multi

```
◄ [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 00] 3C 60 AFFE 05FF 06 03  D7 02 33 10 01 01  :
PropertyValueWrite(Obj=02, Prop=33, Count=1, Start=001, Data=01 )
► [RF: 02 SN: 00090040FF03 3][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 33 10 01 01  :
PropertyValueResponse(Obj=02, Prop=33, Count=1, Start=001, Data=01 )
```

Check that the DUT is able to do everything

Reading the value of flags of KNX RF Multi device =>1F

```
◄ [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 3C 60 AFFE 05FF 05 03  D5 02 34 10 01  :
PropertyValueRead(Obj=02, Prop=34, Count=1, Start=001)
► [RF: 02 SN: 00090040FF03 4][08 04 01 00 00 01] 1C 60 05FF AFFE 06 03  D6 02 34 10 01 1F  :
PropertyValueResponse(Obj=02, Prop=34, Count=1, Start=001, Data=1F )
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on OnOff datapoint on Fast mode with FA  (SN =0x112233445577, GA = 0x0002, Obj Handle = 0x0001, Physical features = 0x01, FA = 0x49)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 77 00 02 00 01 01 49  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 77 00 02 00 01 01 49 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on OnOff datapoint on Slow mode with FA  (SN =0x112233445577, GA = 0x0004, Obj Handle = 0x0001, Physical features = 0x02, FA = 0x49)

```
◄ [RF: 02 SN: 00090040FF03 2] 3C 60 AFFE 05FF 11 02  C7 02 36 00 00 11 22 33 44 55 77 00 04 00 01 02 49  :
FctPropCmd(ObjIdx=02, PropId=36, Data=00 00 11 22 33 44 55 77 00 04 00 01 02 49 )
► [RF: 02 SN: 00090040FF03 5][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on InfoOnOff datapoint on Fast mode with FA (SN =00090040ff03, GA = 0x0003, Obj Handle = 0x0005, Physical features = 0x01, FA = 0x49)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 03 00 05 01 49  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 00 40 FF 03 00 03 00 05 01 49 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

PID_RF_MULTI_OBJECT_LINK => Set RF Multi link on InfoOnOff datapoint on Slow mode with FA (SN =00090040ff03, GA = 0x0005, Obj Handle = 0x0003, Physical features = 0x02, FA = 0x49)

```
◄ [RF: 02 SN: 00090040FF03 3] 3C 60 AFFE 05FF 11 02  C7 02 36 01 00 00 09 00 40 FF 03 00 03 00 05 02 49  :
FctPropCmd(ObjIdx=02, PropId=36, Data=01 00 00 09 00 40 FF 03 00 03 00 05 02 49 )
► [RF: 02 SN: 00090040FF03 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 00  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=)
```

Check the links : read 4 links

Property Function command Read( Object Link Read iterator 0) --> OnOff link with fast ack

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 00 )
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 77 00
02 00 01 01 49  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 77 00 02 00 01 01
49 )
```

Property Function command Read( Object Link Read iterator 1) --> OnOff link with fast ack

```
◄ [RF: 02 SN: 00090040FF03 6] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 77 00
04 00 01 02 49  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 77 00 04 00 01 02
49 )
```

Property Function command Read( Object Link Read iterator 2) --> InfoOnOff link with fast ack

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 03 00
03 00 05 01 49  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 03 00 03 00 05
01 49 )
```

Property Function command Read( Object Link Read iterator 3) --> InfoOnOff link with fast ack

```
◄ [RF: 02 SN: 00090040FF03 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 03  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 03 )
► [RF: 02 SN: 00090040FF03 1][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 03 00
03 00 05 02 49  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 03 00 03 00 05
02 49 )
```

Property Function command Read( Object Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF03 0] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 04 )
► [RF: 02 SN: 00090040FF03 2][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Check that DUT correctly understand the command for switching ON using RF Sx

And correctly sends the InfoOnOff datapoint value

F1 (multi) frame is then sent

Last frame is Sx frame.

```
◄ [RF: 00 SN: 112233445577 0][08 04 05 00 0A 09] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01 00] 2C A0 AFFE 0004 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 02] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 03] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 04] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 05] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 06] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF03 0][08 04 05 00 0A 05] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Check that DUT correctly understand the command for switching OFF using RF Fx

And correctly sends the InfoOnOff datapoint value

F1 (multi) frame is then sent

Last frame is Sx frame.

```
◄ [RF: 00 SN: 112233445577 3][08 04 02 00 0A 09] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 02] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 03] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 04] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 05] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 06] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF03 0][08 04 02 00 0A 05] [0A 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00] 2C A0 05FF 0003 01 00  80  : DPT 1.xxx Boolean (switch off)
```

## 7.3 Push-button

### 7.3.1 Linking actuator

The aim is to link a sensor (EITT4) to an actuator (DUT) with various combinations of features.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator.  Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | **EITT4 = 1 sensor<br>(output datapoint)** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **Native<br>RF Ready** | **Native RF Multi** | | | | | | | |
| | | **TX<br>Features** | | no FA<br>no TX Sx<br>no TX Fx | no FA<br>no TX Sx<br>**TX Fx** | no FA<br>**TX Sx**<br>no TX Fx | no FA<br>**TX Sx**<br>**TX Fx** | **FA**<br>no TX Sx<br>no TX Fx | **FA**<br>no TX Sx<br>**TX Fx** | **FA**<br>**TX Sx**<br>no TX Fx | **FA**<br>**TX Sx**<br>**TX Fx** |
| | **RX Features** | | | | | | | | | | |
| **DUT =<br>1 native<br>RF Multi<br>Actuator** | no FA, no RX Sx, no RX Fx | | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x |
| | no FA, no RX Sx, **RX Fx** | | 🟩 case 2_1 | 🟥 case 2_2 | 🟦 case 2_3 | 🟥 case 2_4 | 🟦 case 2_5 | 🟥 case 2_6 | 🟦 case 2_7 | 🟥 case 2_8 | 🟦 case 2_9 |
| | no FA, **RX Sx**, no RX Fx | | 🟥 case 3_1 | 🟥 case 3_2 | 🟥 case 3_3 | 🟪 case 3_4 | 🟪 case 3_5 | 🟥 case 3_6 | 🟥 case 3_7 | 🟪 case 3_8 | 🟪 case 3_9 |
| | no FA, **RX Sx**, **RX Fx** | | 🟩 case 4_1 | 🟥 case 4_2 | 🟦 case 4_3 | 🟪 case 4_4 | 🟦 case 4_5 | 🟥 case 4_6 | 🟦 case 4_7 | 🟪 case 4_8 | 🟦 case 4_9 |
| | **FA**, no RX Sx, no RX Fx | | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x |
| | **FA**, no RX Sx, **RX Fx** | | 🟩 case 6_1 | 🟥 case 6_2 | 🟦 case 6_3 | 🟥 case 6_4 | 🟦 case 6_5 | 🟥 case 6_6 | 🟦 case 6_7<br>(FA) | 🟥 case 6_8 | 🟦 case 6_9<br>(FA) |
| | **FA**, **RX Sx**, no RX Fx | | 🟥 case 7_1 | 🟥 case 7_2 | 🟥 case 7_3 | 🟪 case 7_4 | 🟪 case 7_5 | 🟥 case 7_6 | 🟥 case 7_7 | 🟪 case 7_8<br>(FA) | 🟪 case 7_9<br>(FA) |
| | **FA**, **RX Sx**, **RX Fx** | | 🟩 case 8_1 | 🟥 case 8_2 | 🟦 case 8_3 | 🟪 case 8_4 | 🟦 case 8_5 | 🟥 case 8_2 | 🟦 case 8_7<br>(FA) | 🟪 case 8_8<br>(FA) | 🟦 case 8_9<br>(FA) |

| | |
|---|---|
| 🟪 | Sx link |
| 🟦 | Fx link |
| 🟩 | Ready link |
| 🟥 | no link |

#### 7.3.1.1 Linking actuator: case 4_9

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:no, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x0F

DUT Serial Number = 00090040ff02

EITT4: Sensor Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number = 112233445566

=======================================

Linking actuator channel

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 07 00 00 00] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 0F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0F 00 00 )
```

S=>A Features from the Sensor

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
► [RF: 02 SN: 00090040FF02 3][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
► [RF: 02 SN: 00090040FF02 4][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 1] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 03 )
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 04 )
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 6][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 7][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 0] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
► [RF: 02 SN: 00090040FF02 3][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 0F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0F 00 00 )
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
► [RF: 02 SN: 00090040FF02 4][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
► [RF: 02 SN: 00090040FF02 5][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
◄ [RF: 02 SN: 00090040FF02 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.1.2 Linking actuator: case 6_3

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

DUT Serial Number = 00090040ff02

EITT4: Sensor Capacities: FA:no, TX Sx:no, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x04

EITT4 Serial Number = 112233445566

=========================================

Linking actuator channel

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

S=>A Features from the Sensor

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 04 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 04 00 00 )
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
► [RF: 02 SN: 00090040FF02 3] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 6][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 7][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 04 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 04 00 00 )
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
► [RF: 02 SN: 00090040FF02 4] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
◄ [RF: 02 SN: 00090040FF02 3][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.1.3 Linking actuator: case 6_5

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

DUT Serial Number = 00090040ff02

EITT4: Sensor Capacities: FA:no, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x0C

EITT4 Serial Number = 112233445566

=======================================

Linking actuator channel

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

S=>A Features from the Sensor

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00 05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01 00 )

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Ready

◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  : FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )

► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00 08 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 01 00 )

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  : FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )

► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)

Send a Multi Fx telegram to switch the DUT to ON

◄ [RF: 00 SN: 112233445566 6][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)

► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)

Send a Multi Fx telegram to switch the DUT to OFF

◄ [RF: 00 SN: 112233445566 7][08 04 07 FF 00 09] 2C A0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)

► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)

◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )

► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )

◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0C 00 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0C 00 00 )

► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )

◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )

► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )

◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )

► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )

◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )

► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )

◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  : NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )

◄ [RF: 02 SN: 00090040FF02 3][08 04 07 FF 00 01] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )

► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)

### 7.3.1.4 Linking actuator: case 6_9

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

DUT Serial Number = 00090040ff02

EITT4: Sensor Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

S=>A Features from the Sensor

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA Slot=FF, Switch OnOff GA=0000)

A=>S Link resp (link added FA Slot=FF GA=0000)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 4] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

A=>S Link resp (link added FA Slot=00 GA=0000)

```
◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Fx with FA

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 01
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx with Fast Ack requested telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 0][08 04 07 FF 01 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 00 SN: 00090040FF02 0] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx with Fast Ack requested telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 1][08 04 07 FF 01 09] 2C A0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 2][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )

► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )

◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )

◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )

► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )

◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )

► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )

◄ [RF: 02 SN: 00090040FF02 3][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )

► [RF: 02 SN: 00090040FF02 3][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.1.5 Linking actuator: case 7_5

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x1E

DUT Serial Number = 00090040ff02

EITT4: Sensor Capacities: FA:no, TX Sx:yes, TX Fx:no, RX Sx:no, RX Fx:no --> 0x0C

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )
```

S=>A Features from the Sensor

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Sx

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 02 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 02
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 08 00 00 05] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Sx telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 6][08 04 08 FF 00 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 08 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Sx telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 7][08 04 08 FF 00 09] 2C A0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 08 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )
```

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )
```

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )
```

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

```
◄ [RF: 02 SN: 00090040FF02 3][08 04 08 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 08 00 00 05] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.1.6 Linking actuator: case 7_9

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT: Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x1E

DUT Serial Number = 00090040ff02

EITT4: Sensor   Capacities: FA:yes, TX Sx:yes, TX Fx:yes,  RX Sx:no, RX Fx:no  --> 0x1C

EITT4 Serial Number = 112233445566

=======================================

Linking actuator channel

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA Slot=FF, Switch OnOff GA=0000)

A=>S Link resp (link added FA Slot=FF GA=0000)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 00 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

A=>S Link resp (link added FA Slot=00 GA=0000)

```
◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Sx with FA

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 02 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 02
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 08 00 00 05] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Sx telegram to switch the DUT to ON with 1 expected FA

```
◄ [RF: 00 SN: 112233445566 0][08 04 08 FF 01 09] 2C A0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
►   [RF: 00 SN: 00090040FF02 0] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Sx telegram to switch the DUT to OFF with 1 expected FA

```
◄ [RF: 00 SN: 112233445566 1][08 04 08 FF 01 09] 2C E0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 2][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )

◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )

► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )

◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )

► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )

◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )

◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )

► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )

◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )

► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )

◄ [RF: 02 SN: 00090040FF02 3][08 04 08 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )

► [RF: 02 SN: 00090040FF02 3][08 04 08 00 00 05] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

#### 7.3.1.7 Linking actuator: case 8_1

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x1F

DUT Serial Number = 00090040ff02

EITT4: Sensor   native Ready sensor

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 01 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 04
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 01 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 04
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 01 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Ready telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 5][08 04 01 FF 00 09] 2C E0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 6][08 04 01 FF 00 09] 2C E0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
◄ [RF: 02 SN: 112233445566 7][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )

► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )

► [RF: 02 SN: 00090040FF02 2] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )

◄ [RF: 02 SN: 112233445566 0] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )

► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )

◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )

►RF: 02 SN: 00090040FF02 4] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )

◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )

► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )

◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )

◄ [RF: 02 SN: 00090040FF02 3][08 04 01 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )

► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.1.8 Linking actuator: case 8_4

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Receiver Capacities: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x1F

DUT Serial Number = 00090040ff02

EITT4: Sensor   Capacities: FA:no, TX Sx:yes,  TX Fx:no, RX Sx:no, RX Fx:no  --> 0x08

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )

► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 08 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 08 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=1007)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0008)

A=>S Link resp (link added CC=01 GA=0008)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 08 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Sx

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 03 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
08 00 01 02 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 08 00 01 02
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Sx telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 6][08 04 08 FF 00 09] 2C E0 AFFE 0008 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 08 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Sx telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 7][08 04 08 FF 00 09] 2C E0 AFFE 0008 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF02 2][08 04 08 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 08 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 08 00 00 )
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 10 07  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 10 07 )
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 02 00 05 )
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 08 )
► [RF: 02 SN: 00090040FF02 5] 0C E0 05FF 0000 08 03  E4 00 00 3B 92 01 00 08  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=92 01 00 08 )
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
◄ [RF: 02 SN: 00090040FF02 3][08 04 07 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

### 7.3.2 Linking sensor

The aim is to link a sensor (DUT) to an actuator (EITT4) with various combinations of features.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator.  Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | **EITT4 = 1 actuator<br>(input datapoint)** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **Native<br>RF Ready** | **Native<br>RF Multi (RX features)** | | | | | | | |
| | | **RX Features** | | no FA<br>no RX Sx<br>no RX Fx | no FA<br>no RX Sx<br>**RX Fx** | no FA<br>**RX Sx**<br>no RX Fx | no FA<br>**RX Sx**<br>**RX Fx** | **FA**<br>no RX Sx<br>no RX Fx | **FA**<br>no RX Sx<br>**RX Fx** | **FA**<br>**RX Sx**<br>no RX Fx | **FA**<br>**RX Sx**<br>**RX Fx** |
| | **TX Features** | | | | | | | | | | |
| **DUT = 1 native RF Multi Sensor** | no FA, no TX Sx, no TX Fx | | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x | 🟥 case 1_x |
| | no FA, no TX Sx, **TX Fx** | | 🟩 case 2_1 | 🟥 case 2_2 | 🟦 case 2_3 | 🟥 case 2_4 | 🟦 case 2_5 | 🟥 case 2_6 | 🟦 case 2_7 | 🟥 case 2_8 | 🟦 case 2_9 |
| | no FA, **TX Sx**, no TX Fx | | 🟥 case 3_1 | 🟥 case 3_2 | 🟥 case 3_3 | 🟪 case 3_4 | 🟪 case 3_5 | 🟥 case 3_6 | 🟥 case 3_7 | 🟪 case 3_8 | 🟪 case 3_9 |
| | no FA, **TX Sx**, **TX Fx** | | 🟩 case 4_1 | 🟥 case 4_2 | 🟦 case 4_3 | 🟪 case 4_4 | 🟦 case 4_5 | 🟥 case 4_6 | 🟦 case 4_7 | 🟪 case 4_8 | 🟦 case 4_9 |
| | **FA**, no TX Sx, no TX Fx | | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x | 🟥 case 5_x |
| | **FA**, no TX Sx, **TX Fx** | | 🟩 case 6_1 | 🟥 case 6_2 | 🟦 case 6_3 | 🟥 case 6_4 | 🟦 case 6_5 | 🟥 case 6_6 | 🟦 case 6_7<br>(FA) | 🟥 case 6_8 | 🟦 case 6_9<br>(FA) |
| | **FA**, **TX Sx**, no TX Fx | | 🟥 case 7_1 | 🟥 case 7_2 | 🟥 case 7_3 | 🟪 case 7_4 | 🟪 case 7_5 | 🟥 case 7_6 | 🟥 case 7_7 | 🟪 case 7_8<br>(FA) | 🟪 case 7_9<br>(FA) |
| | **FA**, **TX Sx**, **TX Fx** | | 🟩 **case 8_1** | 🟥 case 8_2 | 🟦 **case 8_3** | 🟪 **case 8_4** | 🟦 **case 8_5** | 🟥 case 8_6 | 🟦 **case 8_7<br>(FA)** | 🟪 **case 8_8<br>(FA)** | 🟦 case 8_9<br>(FA) |

🟪 Sx link
🟦 Fx link
🟩 Ready link
🟥 no link

(FA) : means that the Fast Acknowledge is managed in the link

### 7.3.2.1 Linking sensor: case 8.1

=========== TEST SETUP ================

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features: native Ready

EITT4 Serial Number = 112233445566

========================================

Linking Sensor

S=>A Start link Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 112233445566 1][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
04 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
04 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Ready mode.

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

### 7.3.2.2 Linking sensor: case 8.3

--------------------------------------

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features:   FA:no, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x0D

EITT4 Serial Number = 112233445566

======================================

Linking Sensor

S=>A Start link  Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0D 00 00 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 1][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Multi Fx mode.

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 02] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

### 7.3.2.3 Linking sensor: case 8.4

=========== TEST SETUP ================

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features: FA:no, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x0E

EITT4 Serial Number = 112233445566

=======================================

Linking Sensor

S=>A Start link Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0E 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Sx

```
◄ [RF: 02 SN: 112233445566 1] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 02 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
02 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Multi Sx mode.

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 05] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

### 7.3.2.4 Linking sensor: case 8.5

=========== TEST SETUP ================

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features: FA:no, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x0F

EITT4 Serial Number = 112233445566

========================================

Linking Sensor

S=>A Start link  Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 0F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0F 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 1] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Multi Fx mode.

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 02] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 02] 0C A0 05FF 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

### 7.3.2.5 Linking sensor: case 8.7

========== TEST SETUP ===============

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

EITT4 Serial Number = 112233445566

========================================

Linking Sensor

S=>A Start link Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(no FA Slot=FF, Switch OnOff status with GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

A=>S Link resp (link added FA Slot=FF GA=0000)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

A=>S Link resp (link added FA Slot=000 GA=0000)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄     06:31:10.8 [RF: 02 SN: 00090040FF02 7] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 1][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Fx with Ack Slot=00

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
01 40 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Multi Fx mode with one Fast Ack requested.

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 01 02][0A 02 01 00] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

#### 7.3.2.6 Linking sensor: case 8.8

=========== TEST SETUP ================

DUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 00090040ff02

EITT4: Actuator Reception Features:   FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x1E

EITT4 Serial Number = 112233445566

========================================

Linking Sensor

S=>A Start link  Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(no FA Slot=FF, Switch OnOff status with GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

A=>S Link resp (link added FA Slot=FF GA=0000)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

A=>S Link resp (link added FA Slot=000 GA=0000)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 7] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 1][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi x with Ack Slot=00

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 02 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
02 40 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Multi Sx mode with one Fast Ack requested.

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 01 05][0A 02 01 00] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

### 7.3.3 Linking 2 actuators

The aim is to link 2 actuators (DUT1 and DUT2) to a sensor (EITT4) with various combinations of features.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator. Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | **EITT4 = 1 sensor (output datapoint)** | | | | |
|---|---|---|---|---|---|---|---|
| | | | **native RF Ready** | **Native RF Multi (TX features)** | | | |
| | | **TX Features** | | **FA**<br>no TX Sx<br>no TX Fx | **FA**<br>no TX Sx<br>**TX Fx** | **FA**<br>**TX Sx**<br>no TX Fx | **FA**<br>**TX Sx**<br>**TX Fx** |
| | **RX Features** | | | | | | |
| **2 DUTs = 1 x native RF Ready + 1 x Multi Actuator (input datapoint)** | **DUT1** : Native Ready<br>**DUT2**: Multi: no FA, **RX Sx**, no RX Fx | | 🟩<br>🟥 case 1_1 | 🟥<br>🟥 case 1_2 | 🟩<br>🟥 case 1_3 | 🟥<br>🟪 case 1_4 | 🟩<br>🟪 case 1_5 |
| | **DUT1** : Native Ready<br>**DUT2**: Multi: no FA, no RX Sx, **RX Fx** | | 🟩<br>🟩 case 2_1 | 🟥<br>🟥 case 2_2 | 🟩<br>🟦 case 2_3 | 🟥<br>🟥 case 2_4 | 🟩<br>🟦 **case 2_5** |
| | **DUT1** : Native Ready<br>**DUT2** : Multi: **FA**, **RX Sx**, no RX Fx | | 🟩<br>🟥 case 3_1 | 🟥<br>🟥 case 3_2 | 🟩<br>🟥 case 3_3 | 🟥<br>🟪 case 3_4 | 🟩<br>🟪 **case 3_5 (FA)** |
| | **DUT1** : Native Ready<br>**DUT2** : Multi: **FA**, no RX Sx, **RX Fx** | | 🟩<br>🟩 case 4_1 | 🟥<br>🟥 case 4_2 | 🟩<br>🟦 case 4_3 (FA) | 🟥<br>🟥 case 4_4 | 🟩<br>🟦 case 4_5 (FA) |

| | |
|---|---|
| 🟪 | Sx link |
| 🟦 | Fx link |
| 🟩 | Ready link |
| 🟥 | no link |

(FA) : means that the Fast Acknowledge is managed in the link

### 7.3.3.1 Linking 2 actuators: case 2.5

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT1 : Native Ready Actuator

DUT1 Serial Number = 00090040ff00

DUT2 : Native Multi Actuator: Receiver Features: FA:no, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x0D

DUT2 Serial Number = 00090040ff02

EITT4: native Multi sensor: Transmitter Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel (DUT1)

========================================

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF00 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF00 0] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 00  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 5][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 00 11 22 33 44 55 66 00
05 00 05  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=02 11 22 33 44 55 66 00 05 00 05 )
```

Property Function command Read( Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF00 1] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 01  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 6][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 01 11 22 33 44 55 66 00
02 00 01  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=03 11 22 33 44 55 66 00 02 00 01 )
```

Property Function command Read( Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF00 2] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 02  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 7][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F FF  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=FF, Data=)
```

Send a Ready telegram to switch the DUT1 to ON

```
◄ [RF: 00 SN: 112233445566 5][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT1 to OFF

```
◄ [RF: 00 SN: 112233445566 6][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

========================================

Linking actuator channel  (DUT2)

========================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 7][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 0D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 0D 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 03  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT2 to ON

```
◄ [RF: 00 SN: 112233445566 0][08 04 07 FF 00 00] 2C A0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT1 to ON

```
◄ [RF: 00 SN: 112233445566 1][08 04 01 FF 00 09] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF00 3][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT2 to OFF

```
◄ [RF: 00 SN: 112233445566 2][08 04 07 FF 01 09] 2C A0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

Send a Ready telegram to switch the DUT1 to OFF

```
◄ [RF: 00 SN: 112233445566 3][08 04 01 FF 00 09] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
► [RF: 02 SN: 00090040FF00 6][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

#### 7.3.3.2 Linking 2 actuators: case 3.5

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT1 : Native Ready Actuator

DUT1 Serial Number = 00090040ff00

DUT2 : Native Multi Actuator: Receiver Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x1E

DUT2 Serial Number = 00090040ff02

EITT4: native Multi sensor: Transmitter Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel (DUT1)

========================================

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
x► [RF: 02 SN: 00090040FF00 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF00 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF00 0] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 00  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 7][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 00 11 22 33 44 55 66 00
05 00 05  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=02 11 22 33 44 55 66 00 05 00 05 )
```

Property Function command Read( Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF00 1] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 01  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 0][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 01 11 22 33 44 55 66 00
02 00 01  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=03 11 22 33 44 55 66 00 02 00 01 )
```

Property Function command Read( Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF00 2] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 02  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 1][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F FF  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=FF, Data=)
```

Send a Ready telegram to switch the DUT1 to ON

```
◄ [RF: 00 SN: 112233445566 5][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF00 2][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT1 to OFF

```
◄ [RF: 00 SN: 112233445566 6][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF00 5][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

========================================

Linking actuator channel   (DUT2)

========================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA=FF Switch OnOff status with GA=0000)

A=>S Link resp (use existing FA=FF=02 GA=0000)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A SetDelLink(FF=00 Switch OnOff GA=0000)

A=>S Link resp (link added FA=00 GA=0000)

```
◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Sx with FA=00

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 01  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 02 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 02
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Sx telegram to switch the DUT2 to ON with nb of expected FA=01

```
◄ [RF: 00 SN: 112233445566 0][08 04 08 FF 01 00][0A 02 01 00] 2C A0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 01] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Sx telegram to switch the DUT2 to OFF wit nb of expected FA=01

```
◄ [RF: 00 SN: 112233445566 1][08 04 08 FF 01 09][0A 02 01 00] 2C A0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 01] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

### 7.3.4 Linking 3 actuators

The aim is to link 3 actuators (DUT1, DUT2 and DUT3) to a sensor (EITT4) with various combinations of features.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator.  Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | **EITT4 = 1 sensor<br>(output datapoint)** | | | | |
|---|---|---|---|---|---|---|---|
| | | | **native<br>RF Ready** | **Native<br>RF Multi (TX features)** | | | |
| | | **TX Features** | | **FA** - no TX Sx - no TX Fx | **FA - no TX Sx - TX Fx** | **FA - TX Sx -** no TX Fx | **FA - TX Sx - TX Fx** |
| **DUTs =<br>1 x native RF Ready + 2 RF Multi Actuators (input datapoint)** | **RX Features** | | | | | | |
| | **DUT1** : Native Ready | | 🟩 | 🟥 | 🟩 | 🟥 | 🟩 |
| | **DUT2**: Multi: no FA, no RX Sx, **RX Fx** | | 🟩 | 🟥 | 🟦 | 🟥 | 🟦 |
| | **DUT3**: Multi: no FA, **RX Sx**, no RX Fx | | 🟥 case 5_1 | 🟥 case 5_2 | 🟥 case 5_3 | 🟪 case 5_4 | 🟪 case 5_5 |
| | **DUT1** : Native Ready | | 🟩 | 🟥 | 🟩 | 🟥 | 🟩 |
| | **DUT2**: Multi: no FA, no RX Sx, **RX Fx** | | 🟩 | 🟥 | 🟦 | 🟥 | 🟦 |
| | **DUT3** : Multi: **FA**, **RX Sx**, no RX Fx | | 🟥 case 6_1 | 🟥 case 6_2 | 🟥 case 6_3 | 🟪 case 6_4 (FA) | 🟪 case 6_5 (FA) |
| | **DUT1** : Native Ready | | 🟩 | 🟥 | 🟩 | 🟥 | 🟩 |
| | **DUT2** : Multi: **FA**, no RX Sx, **RX Fx** | | 🟩 | 🟥 | 🟦 (FA) | 🟥 | 🟦 (FA) |
| | **DUT3**: Multi: no FA, **RX Sx**, no RX Fx | | 🟥 case 7_1 | 🟥 case 7_2 | 🟥 case 7_3 | 🟪 case 7_4 | 🟪 case 7_5 |
| | **DUT1** : Native Ready | | 🟩 | 🟥 | 🟩 | 🟥 | 🟩 |
| | **DUT2** : Multi: **FA**, no RX Sx, **RX Fx** | | 🟩 | 🟥 | 🟦 (FA) | 🟥 | 🟦 **(FA)** |
| | **DUT3** : Multi: **FA**, **RX Sx**, no RX Fx | | 🟥 case 8_1 | 🟥 case 8_2 | 🟥 case 8_3 | 🟪 case 8_4 (FA) | 🟪 **case 8_5 (FA)** |

| | |
|---|---|
| 🟪 | Sx link |
| 🟦 | Fx link |
| 🟩 | Ready link |
| 🟥 | no link |

(FA) : means that the Fast Acknowledge is managed in the link

### 7.3.4.1 Linking 3 actuators: case 8.5

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT1 : Native Ready Actuator

DUT1 Serial Number = 00090040ff00

DUT2 : Native Multi Actuator: Receiver Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

DUT2 Serial Number = 00090040ff01

DUT3 : Native Multi Actuator: Receiver Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x1E

DUT3 Serial Number = 00090040ff02

EITT4: native Multi sensor: Transmitter Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number = 112233445566

========================================

Linking actuator channel (DUT1)

========================================

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF00 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF00 0] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 00  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 00 )
```

```
► [RF: 02 SN: 00090040FF00 5][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 00 11 22 33 44 55 66 00
05 00 05  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=02 11 22 33 44 55 66 00 05 00 05 )
```

Property Function command Read( Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF00 1] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 01  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 01 )
```

```
► [RF: 02 SN: 00090040FF00 6][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 01 11 22 33 44 55 66 00
02 00 01  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=03 11 22 33 44 55 66 00 02 00 01 )
```

Property Function command Read( Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF00 2] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 02  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 02 )
```

```
► [RF: 02 SN: 00090040FF00 7][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F FF  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=FF, Data=)
```

Send a Ready telegram to switch the DUT1 to ON

```
◄ [RF: 00 SN: 112233445566 5][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT1 to OFF

```
◄ [RF: 00 SN: 112233445566 6][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

=====================================

Linking actuator channel  (DUT2)

=====================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 7][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF01 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF01 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF01 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF01 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA=FF Switch OnOff status with GA=0000)

A=>S Link resp (use existing AF=FF GA=0000)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF01 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF01 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A SetDelLink(FA=00 Switch OnOff GA=0000)

A=>S Link resp (link added FA=00 GA=0000)

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF01 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF01 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF01 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF01 0][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Fx with Fast Ack=00

```
◄ [RF: 02 SN: 00090040FF01 1][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 01  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF01 1][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 01
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF01 2][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF01 2][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT2 to ON with 1 expected Fast Ack

```
◄ [RF: 00 SN: 112233445566 7][08 04 07 FF 01 09][0A 02 01 00] 2C A0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

```
► [RF: 00 SN: 00090040FF01 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT1 to ON

```
◄ [RF: 00 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 00 SN: 00090040FF00 0][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT2 to OFF with 1 expected Fast Ack

```
◄ [RF: 00 SN: 112233445566 1][08 04 07 FF 01 09][0A 02 01 00] 2C A0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean
(switch off)
```

```
► [RF: 00 SN: 00090040FF01 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

Send a Ready telegram to switch the DUT1 to OFF

```
◄ [RF: 00 SN: 112233445566 2][08 04 01 FF 00 09] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF00 1][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

========================================

Linking actuator channel  (DUT3)

========================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA=FF Switch OnOff status with GA=0000)

A=>S Link resp (use existing AF=FF GA=0000)

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A SetDelLink(FA=00 Switch OnOff GA=0000)

A=>S Link resp (link added FA=00 GA=0000)

```
◄ [RF: 02 SN: 112233445566 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 66 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 66 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Multi Sx with FA=00

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 01  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 08 00 00 05] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 66 00
02 00 01 02 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 66 00 02 00 01 02
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 08 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 08 00 00 05] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Sx telegram to switch the DUT3 to ON with 1 expected Fast Ack

```
◄ [RF: 00 SN: 112233445566 0][08 04 08 FF 01 09][0A 02 01 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 08 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT2 to ON with 1 expected Fast Ack

◄ [RF: 00 SN: 112233445566 1][08 04 07 FF 01 09][0A 02 01 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)

► [RF: 00 SN: 00090040FF01 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)

Send a Ready telegram to switch the DUT1 to ON

◄ [RF: 00 SN: 112233445566 2][08 04 01 FF 00 09] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)

► [RF: 02 SN: 00090040FF00 4][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)

### 7.3.5 Linking sensor with multiple actuators

The aim is to link 1 sensor (DUT) to 2 or three actuators simulated by EITT4 with various combinations of features.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator. Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | | **EITT4 = 1 x Native Ready + (1or2) RF Multi actuators**<br>**(input datapoint)** | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **RX Features** | **S1** | Native Ready | Native Ready | Native Ready | Native Ready | Native Ready | Native Ready | Native Ready | Native Ready |
| | | | **S2** | Multi Fx | Multi Sx | Multi Fx (FA) | Multi Sx (FA) | Multi Sx | Multi Sx (FA) | Multi Sx | Multi Sx (FA) |
| | | | **S3** | | | | | Multi Fx | Multi Fx | Multi Fx (FA) | Multi Fx (FA) |
| | **TX Features** | | | | | | | | | | |
| **DUT =**<br>**1 RF Multi Sensor**<br>**(output datapoint)** | **FA**, no TX Sx, no TX Fx | | | 🟥<br>case 1_1 | 🟥<br>case 1_2 | 🟥<br>case 1_3 | 🟥<br>case 1_4 | 🟥<br>🟥 case 1_5 | 🟥<br>🟥 case 1_6 | 🟥<br>🟥 case 1_7 | 🟥<br>🟥 case 1_8 |
| | **FA**, no TX Sx, **TX Fx** | | | 🟩<br>🟦<br>case 2_1 | 🟩<br>🟥<br>case 2_2 | 🟩<br>🟦 (FA)<br>case 2_3 | 🟩<br>🟥<br>case 2_4 | 🟩<br>🟥<br>🟦 case 2_5 | 🟩<br>🟥<br>🟦 case 2_6 | 🟩<br>🟥<br>🟦 case 2_7 (FA) | 🟩<br>🟥<br>🟦 case 2_8 (FA) |
| | **FA**, **TX Sx**, no TX Fx | | | 🟥<br>🟥<br>case 3_1 | 🟥<br>🟪<br>case 3_2 | 🟥<br>🟥<br>case 3_3 | 🟥<br>🟪<br>case 3_4 | 🟥<br>🟪<br>🟦 case 3_5 | 🟥<br>🟪<br>🟦 case 3_6 | 🟥<br>🟪<br>🟦 case 3_7 (FA) | 🟥<br>🟥<br>🟥 case 3_8 |
| | **FA**, **TX Sx**, **TX Fx** | | | 🟩<br>🟦<br>case 4_1 | 🟩<br>🟪<br>case 4_2 | 🟩<br>🟦 (FA)<br>case 4_3 | 🟩<br>🟪 (FA)<br>case 4_4 | 🟩<br>🟪<br>🟦 case 4_5 | 🟩<br>🟪 (FA)<br>🟦 case 4_6 | 🟩<br>🟪<br>🟦 case 4_7 (FA) | 🟩<br>🟪 (FA)<br>🟦 **case 4_8 (FA)** |

| | |
|---|---|
| 🟪 | Sx link |
| 🟦 | Fx link |
| 🟩 | Ready link |
| 🟥 | no link |

(FA) : means that the Fast Acknowledge is managed in the link

#### 7.3.5.1 Linking sensor: case 4.8

=========== TEST SETUP ================

BDUT = Sensor

reset to factory

DUT : Sensor Transmission Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

DUT Serial Number = 112233445566

EITT4: Actuator1 Reception Features: native Ready

EITT4 Acutator1 Serial Number = 00090040FF00

EITT4 :Actuator2 Reception Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:yes --> 0x1D

EITT4 Actuator2 Serial Number = 00090040FF01

EITT4 :Actuator3 Reception Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:no --> 0x1E

EITT4 Actuator3 Serial Number = 00090040FF02

========================================

Linking DUT with Actuator1

========================================

S=>A Start link Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF00 0] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF00 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF00 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF00 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF00 4] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 112233445566 1][08 04 01 FF 00 09] 3C 60 AFFE 05FF 05 02  C8 02 36 00 00  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 00 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 00 10
01 00 01 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 00 10 01 00 01
04 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 112233445566 2] 3C 60 AFFE 05FF 05 02  C8 02 36 00 01  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 01 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 00 00
02 00 02 04 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 00 00 02 00 02
04 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 3] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Ready mode.

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

=======================================

Linking DUT with Actuator2

=======================================

S=>A Start link  Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF01 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1D 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1D 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF01 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF01 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(no FA Slot=FF, Switch OnOff status with GA=0000)

```
◄ [RF: 02 SN: 00090040FF01 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )
```

```
► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

A=>S Link resp (link added FA Slot=FF GA=0000)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF01 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

```
◄ [RF: 02 SN: 00090040FF01 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

A=>S Link resp (link added FA Slot=000 GA=0000)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF01 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF01 7] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 2) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 3][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 02  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 02 )
```

```
► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 01 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 01 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 3) --> Switch OnOff link on Multi Fx with Ack Slot=00

```
◄ [RF: 02 SN: 112233445566 4] 3C 60 AFFE 05FF 05 02  C8 02 36 00 03  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 03 )
```

```
► [RF: 02 SN: 112233445566 7][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 01 00
02 00 02 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 01 00 02 00 02
01 40 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 5] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 04 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Ready mode.

A telegram is sent in Multi Fx mode with one Fast Ack requested.

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)

► [RF: 02 SN: 112233445566 2][08 04 01 00 01 02][0A 02 01 00] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

========================================

Linking DUT with Actuator3

========================================

S=>A Start link  Manuf code and 2 datapoint to be linked Bidirectionnal device and no additional frames

```
► [RF: 02 SN: 112233445566 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

A=>S Features from the Actuator

S=>A Features from the Sensor

```
◄ [RF: 02 SN: 00090040FF02 0] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1E 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1E 00 00 )

► [RF: 02 SN: 112233445566 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

```
◄ [RF: 02 SN: 00090040FF02 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )

► [RF: 02 SN: 112233445566 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

A=>S Begin Connection

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

```
◄ [RF: 02 SN: 00090040FF02 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )

► [RF: 02 SN: 112233445566 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

A=>S Link resp (use existing CC=02 GA=1001)

S=>A SetDelLink(no FA Slot=FF, Switch OnOff status with GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 02 10 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 10 01 )

► [RF: 02 SN: 112233445566 6] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

A=>S Link resp (link added FA Slot=FF GA=0000)

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

```
◄ [RF: 02 SN: 00090040FF02 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )

► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

A=>S Link resp (link added CC=01 GA=0002)

S=>A SetDelLink(FA Slot=00, Switch OnOff GA=0000)

```
◄ [RF: 02 SN: 00090040FF02 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

```
► [RF: 02 SN: 112233445566 6] 0C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

A=>S Link resp (link added FA Slot=000 GA=0000)

S=>A Stop link with no error

```
◄ [RF: 02 SN: 00090040FF02 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

A=>S Quit config mode

```
◄ [RF: 02 SN: 00090040FF02 7] 2C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 112233445566 5][08 04 01 FF 00 00] 3C 60 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 112233445566 0][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 00 00 09 00 40 FF 02 10
01 00 01 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 00 09 00 40 FF 02 10 01 00 01
01 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 5) --> Switch OnOff link on Multi x with Ack Slot=00

```
◄ [RF: 02 SN: 112233445566 6] 3C 60 AFFE 05FF 05 02  C8 02 36 00 05  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 05 )
```

```
► [RF: 02 SN: 112233445566 1][08 04 01 00 00 01] 1C 60 05FF AFFE 11 02  C9 02 36 00 01 00 09 00 40 FF 02 00
02 00 02 02 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 00 09 00 40 FF 02 00 02 00 02
02 40 )
```

Property Function command Read( Multi_Object_Link Read iterator 6) --> Shall answer: no more link

```
◄ [RF: 02 SN: 112233445566 7] 3C 60 AFFE 05FF 05 02  C8 02 36 00 06  : FctPropState_Read(ObjIdx=02,
PropId=36, 00 06 )
```

```
► [RF: 02 SN: 112233445566 2][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Press the Push-Button of the DUT.

A telegram is sent in Ready mode.

A telegram is sent in Multi Fx mode with one Fast Ack requested.

A telegram is sent in Multi Sx mode with one Fast Ack requested.

```
► [RF: 00 SN: 112233445566 0][08 04 01 00 00 01] 0C E0 05FF 0002 01 00  81  : DPT 1.xxx Boolean (switch on)

► [RF: 00 SN: 112233445566 0][08 04 01 00 01 02][0A 02 01 00] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)

► [RF: 00 SN: 112233445566 0][08 04 01 00 01 05][0A 02 01 00] 0C A0 05FF 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

### 7.3.6 Linking actuator Fast Ack

The aim is to link 1 actuator (DUT) to 3 sensors (EITT4) with various combinations of features and Fast Ack numbers.

The following table shows the possible combination between the transmit features of the sensor and the receive features of the actuator.  Not all the combinations are checked. Only the most representative and realistic cases are verified.

The cases in bold in the table are verified.

FA = Fast Acknowledge management

TX Sx = the device is capable of transmitting on the slow frequencies

TX Fx = the device is capable of transmitting on the fast frequencies

RX Sx = the device is capable of scanning the slow frequencies (reception)

RX Fx = the device is capable of scanning on the fast frequencies (reception)

no FA = no Fast Acknowledge management

no TX Sx = the device is not capable of transmitting on the slow frequencies

no TX Fx = the device is not capable of transmitting on the fast frequencies

no RX Sx = the device is not capable of scanning the slow frequencies (reception)

no RX Fx = the device is not capable of scanning on the fast frequencies (reception)

| | | | | **EITT4 = 1 Native Ready + 2 RF Multi sensors (output datapoint)** | |
|---|---|---|---|---|---|
| | | **TX Features** | **S1**<br>**S2**<br>**S3** | Native Ready<br>Multi Fx n°1 (FA=0)<br>Multi Fx n°2 (FA=63) | Native Ready<br>Multi Sx n°1 (FA=0)<br>Multi Sx n°2 (FA=63) |
| | **RX Features** | | | | |
| **DUT = 1 RF Multi Actuator (output datapoint)** | **FA**, no RX Sx, **RX Fx** | | | 🟩<br>🟦 (FA)<br>case 1_1 (FA) | 🟩<br>🟥 (FA)<br>case 1_2 (FA) |
| | **FA**, **RX Sx**, no RX Fx | | | 🟥<br>case 2_1 | 🟥<br>🟪 (FA)<br>case 2_2 (FA) |
| | **FA**, **RX Sx**, **RX Fx** | | | 🟩<br>🟦 **(FA)**<br>**case 3_1 (FA)** | 🟩<br>🟪 (FA)<br>case 3_2 (FA) |

| | |
|---|---|
| 🟪 | Sx link |
| 🟦 | Fx link |
| 🟩 | Ready link |
| 🟥 | no link |

(FA) : means that the Fast Acknowledge is managed in the link

### 7.3.6.1 Linking actuator: case 3.1

=========== TEST SETUP ================

DUT = Actuator

reset to factory

DUT : Native Multi Actuator: Receiver Features: FA:yes, TX Sx:yes, TX Fx:yes, RX Sx:yes, RX Fx:yes --> 0x1F

DUT Serial Number = 00090040FF02

EITT4: native ready Sensor1

EITT4 Serial Number Sensor1= 112233445566

EITT4 : Native Multi Sensor2: Transmitter Features: ACK:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number Sensor2 = 112233445567

EITT4 : Native Multi Sensor3: Transmitter Features: ACK:yes, TX Sx:yes, TX Fx:yes, RX Sx:no, RX Fx:no --> 0x1C

EITT4 Serial Number Sensor3 = 112233445568

========================================

Linking actuator channel (DUT with EITT4 sensor1)

========================================

S=>A Start link Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S Features from the Actuator

```
◄ [RF: 02 SN: 112233445566 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
```

A=>S Channel function actuator 0x0103 OnOff Actuator Scene

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445566 1] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445566 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445566 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445566 4] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Object_Link Read iterator 0) --> Switch OnOff Status link on Ready

```
◄ [RF: 02 SN: 00090040FF02 0] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 00  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 00 11 22 33 44 55 66 00
05 00 05  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=02 11 22 33 44 55 66 00 05 00 05 )
```

Property Function command Read( Object_Link Read iterator 1) --> Switch OnOff link on Ready

```
◄ [RF: 02 SN: 00090040FF02 1] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 01  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 1C 60 05FF AFFE 0F 02  C9 00 3F 00 01 11 22 33 44 55 66 00
02 00 01  : FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=00, Data=03 11 22 33 44 55 66 00 02 00 01 )
```

Property Function command Read( Object_Link Read iterator 2) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2] 3C 60 AFFE 05FF 05 02  C8 00 3F 00 02  : FctPropState_Read(ObjIdx=00,
PropId=3F, 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 1C 60 05FF AFFE 04 02  C9 00 3F FF  :
FctPropState_Rsp(ObjIdx=00, PropId=3F, RetCode=FF, Data=)
```

Send a Ready telegram to switch the DUT to ON

```
◄ [RF: 00 SN: 112233445566 5][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Ready telegram to switch the DUT to OFF

```
◄ [RF: 00 SN: 112233445566 6][08 04 01 FF 00 00] 2C E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

========================================

Linking actuator channel  (DUT with EITT4 sensor2)

========================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445567 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445567 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445567 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445567 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA=FF Switch OnOff status with GA=0000)

A=>S Link resp (use existing AF=FF GA=0000)

```
◄ [RF: 02 SN: 112233445567 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445567 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A SetDelLink(FA=00 Switch OnOff GA=0000)

A=>S Link resp (link added FA=00 GA=0000)

```
◄ [RF: 02 SN: 112233445567 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 00 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445567 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 4) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 04  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 04 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 67 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 67 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 5) --> Switch OnOff link on Multi Fx with Fast Ack=00

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 05  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 05 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 67 00
02 00 01 01 40  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 67 00 02 00 01 01
40 )
```

Property Function command Read( Multi_Object_Link Read iterator 6) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 00] 3C 20 AFFE 05FF 05 02  C8 02 36 00 06  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 06 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT to ON with 1 expected Fast Ack

```
◄ [RF: 00 SN: 112233445567 0][08 04 07 FF 01 09][0A 02 01 00] 2C A0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean
(switch on)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT to OFF with 1 expected Fast Ack

```
◄ [RF: 00 SN: 112233445567 1][08 04 07 FF 01 09][0A 02 01 00]  2C A0 AFFE 0002 01 00  80  : DPT 1.xxx
Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```

=======================================

Linking actuator channel  (DUT with EITT4 sensor3)

=======================================

S=>A Start link  Manuf code and 2 datapoints to be linked Bidirectionnal device and no additional frames

A=>S  Features from the Actuator

```
◄ [RF: 02 SN: 112233445568 0][08 04 01 FF 00 09] 2C E0 05FF 0000 08 03  E4 00 00 3B 20 00 09 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=20 00 09 02 )
```

```
► [RF: 02 SN: 00090040FF02 0][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B D0 1F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1F 00 00 )
```

S=>A Features from the Sensor

A=>S  Channel function actuator 0x0103  OnOff Actuator Scene

```
◄ [RF: 02 SN: 112233445568 1] 2C E0 05FF 0000 08 03  E4 00 00 3B D0 1C 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=D0 1C 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 30 01 03 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=30 01 03 00 )
```

S=>A Channel Function Sensor 0x316 generic PB 1/2 info 5

A=>S Begin Connection

```
◄ [RF: 02 SN: 112233445568 2] 2C E0 05FF 0000 08 03  E4 00 00 3B 40 03 16 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=40 03 16 00 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 70 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=70 00 00 00 )
```

S=>A SetDelLink(CC=02 Switch OnOff status with GA=0001)

A=>S Link resp (use existing CC=02 GA=0005)

```
◄ [RF: 02 SN: 112233445568 3] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 02 00 01  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 02 00 01 )
```

```
► [RF: 02 SN: 00090040FF02 3][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 02 00 05  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 02 00 05 )
```

S=>A SetDelLink(FA=FF Switch OnOff status with GA=0000)

A=>S Link resp (use existing AF=FF GA=0000)

```
◄ [RF: 02 SN: 112233445568 4] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 FF 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 4][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 91 FF 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=91 FF 00 00 )
```

S=>A SetDelLink(CC=01 Switch OnOff GA=0002)

A=>S Link resp (link added CC=01 GA=0002)

```
◄ [RF: 02 SN: 112233445568 5] 2C E0 05FF 0000 08 03  E4 00 00 3B 80 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=80 01 00 02 )
```

```
► [RF: 02 SN: 00090040FF02 5][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 01 00 02  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 01 00 02 )
```

S=>A SetDelLink(FA=3F Switch OnOff GA=0000)

A=>S Link resp (link added FA=3F GA=0000)

```
◄ [RF: 02 SN: 112233445568 6] 2C E0 05FF 0000 08 03  E4 00 00 3B 82 3F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=82 3F 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 6][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B 90 3F 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=90 3F 00 00 )
```

S=>A Stop link with no error

A=>S Quit config mode

```
◄ [RF: 02 SN: 112233445568 7] 2C E0 05FF 0000 08 03  E4 00 00 3B A0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=A0 00 00 00 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 01 00 00 01] 0C E0 05FF 0000 08 03  E4 00 00 3B B0 00 00 00  :
NetworkParameterWrite(ObjType=0000, PID=3B, Value=B0 00 00 00 )
```

Property Function command Read( Multi_Object_Link Read iterator 6) --> Switch OnOff Status link on Multi Fx

```
◄ [RF: 02 SN: 00090040FF02 0][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 06  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 06 )
```

```
► [RF: 02 SN: 00090040FF02 7][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 01 11 22 33 44 55 68 00
05 00 05 01 00  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=01 11 22 33 44 55 68 00 05 00 05 01
00 )
```

Property Function command Read( Multi_Object_Link Read iterator 7) --> Switch OnOff link on Multi Fx with FA=00

```
◄ [RF: 02 SN: 00090040FF02 1][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 07  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 07 )
```

```
► [RF: 02 SN: 00090040FF02 1][08 04 07 00 00 02] 1C 20 05FF AFFE 11 02  C9 02 36 00 00 11 22 33 44 55 68 00
02 00 01 01 7F  : FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=00, Data=00 11 22 33 44 55 68 00 02 00 01
01 7F )
```

Property Function command Read( Multi_Object_Link Read iterator 8) --> Shall answer: no more link

```
◄ [RF: 02 SN: 00090040FF02 2][08 04 07 FF 00 09] 3C 20 AFFE 05FF 05 02  C8 02 36 00 08  :
FctPropState_Read(ObjIdx=02, PropId=36, 00 08 )
```

```
► [RF: 02 SN: 00090040FF02 2][08 04 07 00 00 02] 1C 20 05FF AFFE 04 02  C9 02 36 FF  :
FctPropState_Rsp(ObjIdx=02, PropId=36, RetCode=FF, Data=)
```

Send a Multi Fx telegram to switch the DUT to ON with 64 expected Fast Ack

```
◄ [RF: 00 SN: 112233445568 0][08 04 07 FF 40 09][0A 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00] 2C
E0 AFFE 0002 01 00  81  : DPT 1.xxx Boolean (switch on)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  81  : DPT 1.xxx Boolean (switch on)
```

Send a Multi Fx telegram to switch the DUT to OFF with 64 expected Fast Ack

```
◄ [RF: 00 SN: 112233445568 1][08 04 07 FF 40 09][0A 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00] 2C
E0 AFFE 0002 01 00  80  : DPT 1.xxx Boolean (switch off)
```

```
► [RF: 00 SN: 00090040FF02 0][08 04 07 00 00 02] 0C A0 05FF 0005 01 00  80  : DPT 1.xxx Boolean (switch off)
```