



System Conformance Testing

8

Test Suite Supplement B

B

Easy Controller fixed/relocatable DMA

Summary

This document contains test specifications for Testing of Easy Ctrl-Mode products

Version 01.02.01 is a KNX Approved Standard.

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

Document Updates

Version	Date	Modifications
0.3	2002.09	Working Document in KonCert
0.4	2003.07	update prior and after KonCert July meeting
0.5	2003.07	update prior KonCert December meeting
0.6	2004.02	update prior to first product audit
0.7	2004.06	Preparation for release for voting
0.8	2004.09	Last minute comments – sending to release for voting
0.9	2005.06	Integration of comments from Release for voting – preparation for KonCert June meeting – integration of improvements at more product audits – improvement of clause 4.1 by Merten – additional clause 2.1.1.7 – correction in 2.1.1.7 just prior to submission for FV
1.0	2005.11	Resolution of comments of final voting cycle
1.1	2009.06	Readying document for KNX specifications V 2.0
1.2	2009.10	Shift of clause 7 and 8 to Volume 8/3/7 and replace with reference to that Volume
01.02.01	2013.10.24	Editorial updates for the publication of KNX Specifications 2.1.

Filename: 08_TSSB System Conformance Testing - Ctrl-Mode Tests v01.02.01 AS.docx
Version: 01.02.01
Status: Approved Standard
Savedate: 2013.10.24
Number of pages: 13

Contents

1	Introduction	4
2	Test of (reduced) System Interface Objects	6
2.1	Introduction	6
2.1.1	Availability of Interface Object and respective properties	6
3	Test of Localisation Flag	9
4	Test of Localisation Channels – Server Test	10
4.1	Test of sensor channel	10
4.2	Test of actuator channel	11
5	Test of Device Descriptor Type 0 support by client	11
6	Group Address Check Test set-up for Ctrl-Mode	12
7	Test of Address Table	12
8	Test of structure of Association Table	12
9	Test of Object Table (Easy)	13
9.1	Server Tests	13
9.2	Client Tests	13
10	Programming of devices	13
10.1	via programming mode	13
10.2	Programming of devices via serial number	13

1 Introduction

The underneath tests need to be carried out when assessing the conformity of Ctrl-Mode products:

When acting as a normal Ctrl-Mode device (server), it shall be submitted to the following system tests:

- Transport Layer Broadcast (see Volume 8/3/4)
- Transport Layer connection-oriented style 1 (Easy controller relocatable DMA) respectively style 2 (Easy controller fixed DMA) - see appropriate tests in 8/3/4
- Direct Memory Access via Memory Read and Write (see appropriate tests in Volume 8/3/7) – in case of Ctrl Mode Fixed DMA additionally test whether EEPROM can be written from 119h onwards.
- Interface Object Handling (see appropriate tests in Volume 8/3/7 – Tests of System Interface Objects and their properties – see clause 2)
- Reduced Interface Object Handling (see appropriate tests in Volume 8/3/7 – Tests of Interface Object type and Serial number see clause 2)
- Load Controls see Test Suite Supplement G, whereby it shall be taken into account that only a subset of the controls and states shall be supported for Easy mode Relocatable DMA (“load” and “load complete”, see AN 032 - the reaction to other load controls, e.g. unload, is manufacturer specific)
- Restart handling (see appropriate tests in Volume 8/3/7)
- Device Descriptor Type 0 (see appropriate tests in Volume 8/3/7 and Volume 8/3/4) and 2 (see relevant LTS-Test in Test Suite Supplement A), both connection-oriented
- Assignment of Individual address via programming mode or serial number (see appropriate tests in Volume 8/3/7)
- (on open media only) Domain Address Assignment (see appropriate tests in Volume 8/3/7)
- Testing of structure of Address and Association table (see clause 5 and 8)
- Handling of group address check (see appropriate tests of LTS paragraph 3.4 in Test Suite Supplement A – test set-up adjusted to Ctrl-Mode see clause 6)

When acting as a normal Ctrl-Mode-device (server), it shall additionally be submitted to the following programming tests:

- Handling of channel parameters via DMA (see appropriate tests in Volume 8/3/7 – appropriate LTS-Tests paragraph 3.8 in Test Suite Supplement A) or reduced interface objects (see appropriate tests of LTS in Test Suite Supplement A)
- Support of Localisation Flag (see clause3)
- Support of Localisation Channel (see clause4)
- Testing of the length of the Object table (see clause9)

When moreover acting as a Controller (client), the following additional tests shall be carried out:

- Transport Layer tests broadcast (see Volume 8/3/4) and connection oriented (see appropriate tests in Volume 8/3/4)
- Programming of devices via programming mode/serial number – support of IA Check (see relevant LTS-Supervisor tests in Test Suite Supplement A – see clause 10.1 and 10.2 for adaptation of these tests to Ctrl-Mode)
- Distributed address assignment –
 - Case A: planned Individual Address not yet assigned
 - Case B: planned Individual Address already assigned
 - Case C: reaction to T-Disconnect
- Group Address check (see relevant LTS-Supervisor tests in Test Suite Supplement A)
 - positive Case – Group Address does not exist – negative case – Group Address already exists
- Device Descriptor handling type 0 (*see clause 5*) and 2 (see relevant LTS-Supervisor test in Test Suite Supplement A)
 - Case A: returned DD are supported by the controller
 - Case B (negative): returned DD is not supported by the controller
 - Case C (negative): contents of DD is not supported by the controller
- Compliance to connection rules (see Test Suite Supplement E)
- Handling of functional parameters according to the channel code specifications (see TSS E). Use of Load controls – it shall be checked whether the controller only uses the load controls “load” and “load complete”.

Note: The way a controller is updated with knowledge of new easy channels, is manufacturer specific and therefore no part of these test specifications.


As the controller needs as any other connected product its own Individual Address, this has to be assigned prior to the tests.

2 Test of (reduced) System Interface Objects


2.1 Introduction

The following system interface objects are needed for Easy products with relocatable DMA.

#0	#1	#2	#3
Device Obj1.	Address table	Assoc. Table	Application
Prop 1=Type	1=Type	1=Type	1=Type
* Prop 11= Ser. No.	5=load Control	5	5
	7=Table pointer	7	7



Calculation of number of
Group Addresses



Calculation of number of
Associations

* : only in case of support of serial number procedure (see supplement 6)

2.1.1 Availability of Interface Object and respective properties

2.1.1.1 General

When implemented as full interface objects, the availability shall be checked with the PropertyDescriptionRead service (first test step in the underneath paragraphs). In this case, the availability of the required interface objects, their properties and their access rights can also be checked with the ETS App Device Editor, which is available from KNX Association.

When implemented as reduced interface objects only, merely the properties shall be read and/or written with the services PropertyWrite and/or Read (second test step in the underneath paragraphs).

2.1.1.2 Device Object – Property type

Step 1: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT

Test frame (IN): A_PropertyDescription_Read (ObjIndex=0; PropID=1; PropIndex=0)

Acceptance: BDUT sends A_PropertyDescription_Response with correct data

Test frame (OUT): A_PropertyDescription_Response (ObjIndex=0; PropID=1; PropIndex=don't care; property data type=PDT_Unsigned_Int (04h); max. nr. of elements=1; write =disabled)

Note: The object type property is mandatory (read-only) for each implemented object

Step 2: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT

Test frame (IN): A_PropertyValue_Read (ObjIndex=0; PropID=1; Count =1; Start=001)

Acceptance: BDUT sends A_PropertyValue_Response with correct data

Test frame (OUT): A_PropertyValue_Response (ObjIndex=0; PropID=1; Count =1; Start=001; Property Data=0=Object Type ID for “Device Object”)

2.1.1.3 Device Object – Serial number

Step 1: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT

Test frame (IN): A_PropertyDescription_Read (ObjIndex=0; PropID=11d; PropIndex=0)

Acceptance: BDUT sends A_PropertyDescription_Response with correct data

Test frame (OUT): A_PropertyDescription_Response (ObjIndex=0; PropID=11d; PropIndex=don't care; property data type=PDT_Generic_06 (16h or 96h); max. nr. of elements=1; write =en/disabled)

Note: The serial number property may have read and/or write access.

Step 2: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT

Test frame (IN): A_PropertyValue_Read (ObjIndex=0; PropID=11d; Count =1; Start=001)

Acceptance: BDUT sends A_PropertyValue_Response with correct data

Test frame (OUT): A_PropertyValue_Response (ObjIndex=0; PropID=11d; Count =1; Start=001; Property Data=BDUT's Serial Number)

Check received data with the Serial number as indicated in the supplied PIXIT forms.

2.1.1.4 Address Table, Association Table, Application Object – Property Type

The test sequence as described in clause 2.1.1.2 applies, with adapted local Object Index and Object Type ID=1 (Object Type ID for “Address Table Object”), Type ID=2 (Object Type ID for “Association Table Object”) and Type ID=3 (Object Type ID for “Application Object”)

2.1.1.5 Address Table, Association Table, Application Object – Load Control

The tests as laid down in Test Suite Supplement G apply, whereby it shall be taken into account that easy controller devices only support the load controls ‘start loading’ and ‘load complete’ respectively the state ‘loading’ and ‘loaded’.

2.1.1.6 Address Table, Association Table, Application Object – Table Pointer

Step 1: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT

Test frame (IN): A_PropertyDescription_Read (ObjIndex=x; PropID=7d; PropIndex=0)
whereby x = 1 for address table, 2 for association table, 3 for application program

Acceptance: BDUT sends A_PropertyDescription_Response with correct data

Test frame (OUT): A_PropertyDescription_Response (ObjIndex=x; PropID=7d; PropIndex=don't care; property data type=PDT_UNSIGNED_INT (04h); max. nr. of elements=1; write =disabled) whereby x = 1 for address table, 2 for association table, 3 for application program

Step 2: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT

Test frame (IN): A_PropertyValue_Read (ObjIndex=x; PropID=7d; Count =1; Start=001)
whereby x = 1 for address table, 2 for association table, 3 for application program

Acceptance: BDUT sends A_PropertyValue_Response with correct data

Test frame (OUT): A_PropertyValue_Response (ObjIndex=x; PropID=7d; Count =1; Start=001; Property Data=any address) whereby x = 1 for address table, 2 for association table, 3 for application program

Note: in the case of Ctrl Mode fixed DMA, it shall be checked whether the GrAssocTabPtr (Group Association Table Pointer) and the GrObjTabPtr (Group Object Table Pointer) can be read via the addresses 111h respectively 112h. The response from the BDUT may be any address whereby the returned value of the GrAssocTabPtr shall be lower than the one of the GrObjTabPtr and greater than 116h.

2.1.1.7 Parameter Pointer¹

Step 1: Check if BDUT sends property description with correct data

Stimuli: send A_PropertyDescription_Read to BDUT

Test frame (IN): A_PropertyDescription_Read (ObjIndex=3; PropID=51d; PropIndex=0)

Acceptance: BDUT sends A_PropertyDescription_Response with correct data

Test frame (OUT): A_PropertyDescription_Response (ObjIndex=3; PropID=51d; PropIndex=don't care; property data type=PDT_Unsigned_Int (04h); max. nr. of elements=1; write =disabled)

Step 2: Check if BDUT sends property value with correct data.

Stimuli: send A_PropertyValue_Read to BDUT

Test frame (IN): A_PropertyValue_Read (ObjIndex=3; PropID=51d; Count =1; Start=001)

Acceptance: BDUT sends A_PropertyValue_Response with correct data

Test frame (OUT): A_PropertyValue_Response (ObjIndex=3; PropID=51d; Count =1; Start=001; Property Data=any address)

¹ Only Easy implementations based on 0701 according AN069

3 Test of Localisation Flag

It shall be tested that in the case of:

- Sensors

After connection of a new sensor to the network, within a specific time the controller will attribute in accordance to the channel definitions to the relevant objects with the localisation flag set a unique group address. After stimulation of the sensor, a frame containing the assigned group address shall be sent on the bus. The type of stimulus needed to generate a frame is manufacturer specific (e.g. push a button).

- Actuators

After connection of a new actuator to the network, within a specific time the controller will attribute in accordance to the channel definitions to the relevant objects with the localisation flag set a unique group address. The controller will subsequently be sent containing the assigned group address and an appropriate value, causing the actuator to visualize the addressed channel. The type of visualization is manufacturer specific (e.g. actuator changes its switching status).

4 Test of Localisation Channels – Server Test

4.1 Test of sensor channel

- Identify the sensor channel in the BDUT (check via supplied documentation)
- stimulate the controller to attribute group addresses for the two localization Datapoints (Localization_State and Channel_Activation)

Case a) Easy Ctrl-Mode-Device “Fixed-DMA” AND Mask 0012h

Set Runtime Error Flags

DMP-MemWrite_Rco(010Dh, 010Dh, 00h)

Case b) Easy Ctrl-Mode-Device “Relocatable-DMA” and Mask 0020h, 0021h

Set Group Address Table “Loading”

DMP_LoadStateMachineWrite_Rco_IO(object_index = 01h,

PID==PID_LOAD_STATE_CONTROL, start_index = 1, nr_of_elem = 1, Load)

IN BC FFFF 0202 60 80 :T-Connect(Addr=0202)

IN BC FFFF 0202 67 42 82 01 19 C1 00 C1 01 :MemoryWrite(Count=02, Addr=0119, Data=C1 00 C1 01)

OUT B0 0202 FFFF 60 C2 :T-Ack(Seq=0)

IN BC FFFF 0202 64 46 81 01 16 03 :MemoryWrite(Count=01, Addr=0116, Data=03)

OUT B0 0202 FFFF 60 C6 :T-Ack(Seq=1)

IN BC FFFF 0202 67 4A 82 01 52 01 07 02 08 :MemoryWrite(Count=02, Addr=0152, Data=01 07 02 08)

OUT B0 0202 FFFF 60 CA :T-Ack(Seq=2)

IN BC FFFF 0202 64 4E 81 01 51 02 :MemoryWrite(Count=01, Addr=0151, Data=02)

OUT B0 0202 FFFF 60 CE :T-Ack(Seq=3)

IN BC FFFF 0202 60 81 :T-Disconnect

Case a) Easy Ctrl-Mode-Device “Fixed-DMA” AND Mask 0012h

Clear Runtime Error Flags

DMP-MemWrite_Rco(010Dh, 010Dh, FFh)

Case b) Easy Ctrl-Mode-Device “Relocatable-DMA” and Mask 0020h, 0021h

Set Group Address Table “Loaded”

DMP_LoadStateMachineWrite_Rco_IO(object_index = 01h,

PID==PID_LOAD_STATE_CONTROL, start_index = 1, nr_of_elem = 1, LoadCompleted)

- Stimulate the controller to send a group telegram on the Datapoint Localization_State with the value “1” (Localization State ON).

IN BC 1041 C100 E1 00 81

- stimulate the respective sensor channel

- Check whether the Datapoint Channel_Activation sends a telegram, of which the data width corresponds to the number of device channels (1 byte for up to 8 channels, 2 byte for up to 16 channels and 3 byte for up to 24 channels). Check whether the correct bit position is set depending on the activated channel and whether the value is correct.

OUT BC 1101 C101 E2 00 80 02

- Stimulate the controller to end the Localization state by sending a group telegram on the Datapoint Localization_State with the value “0” (Localization State off) and subsequently link the device.

IN BC 1041 C100 E1 00 80

- activate the respective sensor channel
- Check whether a telegram is sent on a non-localization group address.

OUT BC 1101 C800 E1 00 81

4.2 Test of actuator channel

- Identify the actuator channel in the BDUT (check via supplied documentation)
- stimulate the controller to attribute group addresses for the two localization Datapoints (Localization_State and Channel_Activation)

See step 2 in clause 4.1.

- stimulate the controller to send a group telegram on the Datapoint Localization_State with the value “1” (Localization State ON).

IN BC 1041 C200 E1 00 81

- generate according to PIXIT a frame with the appropriate data width (depending on the number of channels, see above), correct bit position (according to the channel to be localized) and value “1” (Localization ON).

IN BC 1041 C201 E2 00 80 04

- Check reaction of the actuator channel
- stimulate the controller to end the Localization state by sending a group telegram on the Datapoint Localization_State with the value “0” (Localization State off) and subsequently link the device.

IN BC 1041 C200 E1 00 80

- send group frame with the now attributed group address to the respective actuator channel and check reaction.

IN BC 1041 C900 E1 00 81

5 Test of Device Descriptor Type 0 support by client

Check whether the BDUT differentiates the type of server (fixed or relocatable DMA) depending on the received DD0 (e.g. 0012h respectively 0021h being System 1 respectively System 2 for TP1).

6 Group Address Check Test set-up for Ctrl-Mode

According to the test as given in TSS A, for Ctrl-Mode servers, when reading the `PID_TABLE_REFERENCE` of the address, association and application interface objects (see test in clause 2) the boundaries shall be noted down. The following entry shall then be made (e.g. by the Device Editor or via EITT) in

- the address table (from its starting point) : 05 xxxx C2 34 C2 35 C2 36 C2 37 where xx is the Individual Address and shall not be changed
- the association table (from its starting point) : 04 01 00 03 01 03 02 02 00.

Examples of Test frames

CASE A - out of range - no answer

(1) IN start BC 00.00.002 00/0000 E7 03 DA 00 01 17 01 C2 38
:NetworkParameterRead(ObjType=0001, PID=17, TestInfo=01 C2 38)

CASE B: one group address in range

(2) IN 00:00:02.0 BC 00.00.002 00/0000 E7 03 DA 00 01 17 05 C2 30
:NetworkParameterRead(ObjType=0001, PID=17, TestInfo=05 C2 30)

(3) OUT 00:00:00.0 BC 00.02.255 00.00.002 67 03 DB 00 01 17 05 C2 30
:NetworkParameterResponse(ObjType=0001, PID=17, TestData=05 C2 30)

CASE C: 1 group address to more than one object

(4) IN 00:00:02.0 BC 00.00.002 00/0000 E7 03 DA 00 01 17 01 C2 36
:NetworkParameterRead(ObjType=0001, PID=17, TestInfo=01 C2 36)

(5) OUT 00:00:00.0 BC 00.02.255 00.00.002 67 03 DB 00 01 17 01 C2 36
:NetworkParameterResponse(ObjType=0001, PID=17, TestData=01 C2 36)

CASE D: two group addresses to one Group Object

(6) IN 00:00:02.0 BC 00.00.002 00/0000 E7 03 DA 00 01 17 02 C2 34
:NetworkParameterRead(ObjType=0001, PID=17, TestInfo=02 C2 34)

(7) OUT 00:00:00.0 BC 00.02.255 00.00.002 67 03 DB 00 01 17 02 C2 34
:NetworkParameterResponse(ObjType=0001, PID=17, TestData=02 C2 34)

CASE E: in range but not linked

(8) IN 00:00:02.0 BC 00.00.002 00/0000 E7 03 DA 00 01 17 01 C2 37
:NetworkParameterRead(ObjType=0001, PID=17, TestInfo=01 C2 37)

(9) OUT 00:00:00.0 BC 00.02.255 00.00.002 67 03 DB 00 01 17 01 C2 37
:NetworkParameterResponse(ObjType=0001, PID=17, TestData=01 C2 37)

CASE F: multicast with supported GA - no answer

(10) IN 00:00:02.0 BC 00.00.002 24/0566 E7 03 DA 00 01 17 01 C2 36 :Faulty multicast

7 Test of Address Table

See Tests of clause 4 in Volume 8/3/7.

8 Test of structure of Association Table

See Tests of clause 4 in Volume 8/3/7.

9 Test of Object Table (Easy)

9.1 Server Tests

Depending on the value returned by reading the Group Object Table pointer in case of Ctrl Mode Fixed DMA respectively the PID_TABLE_REFERENCE of the Application Program Object in case of Ctrl Mode Relocatable DMA), it shall be checked that at the address given by these pointers the length of the object table corresponds to what is given in the supplied PIXIT (i.e. corresponds to the number of Group Objects supported by the BDUT)².

In order to ensure that after the indicated object length, the BDUT's parameter block starts (if applicable), the following shall be done (for instance with the Device Editor):

- reading the value of the first parameter of the first channel according to the channel code definition;
 - setting the value of the first parameter of the first channel according to the channel code definition to another value;
- checking the reaction of the BDUT according to the channel code definition.

9.2 Client Tests

- Stimulate client to read the table length and trace the result with EITT (if necessary as part of the parameter handling)

10 Programming of devices

10.1 via programming mode

Case A: IA assignment of a device installed in the network supporting connection oriented communication

Case B: IA assignment of a device installed in the network not supporting connection oriented communication (i.e. sends a disconnect)

Case C: device is detected on the network with same address as device to be programmed

Case D: Reprogramming with same IA (only in the case where controller allows to do so via IA Assignment via programming mode)

Case E: two devices in programming mode

10.2 Programming of devices via serial number

Case A: IA assignment of a device – normal procedure

Case B: IA assignment of a device – device with same IA

² This number can be greater than the number of objects given by the channel definitions as additional objects are allowed.