IO-Link

IO-Link
Test Specification
V1.1.4

Related to
IO-Link Interface and System Specification V1.1.4

June 2024

Order No: 10.032

IO-Link

File name: **IO-Link_Test-Spec_10032_V1.1.4_Jun24.docx**

This specification has been prepared by the IO-Link test team and released by the IO-Link community as final version. It covers all features of the "IO-Link Interface and System" specification V1.1.4.

Any comments, proposals, requests on this document are appreciated. Please use
www.io-link-projects.com for your entries and provide name and email address.
Login: **IOL-Test-V113**
Password: **Report**

Important notes:

NOTE 1 The IO-Link Community Rules shall be observed prior to the development and marketing of IO-Link products. The document can be downloaded from the www.io-link.com portal.

NOTE 2 Any IO-Link device shall provide an associated IODD file. Easy access to the file and potential updates shall be possible. It is the responsibility of the IO-Link device manufacturer to test the IODD file with the help of the IODD-Checker tool available per download from www.io-link.com.

NOTE 3 Any IO-Link devices shall provide an associated manufacturer declaration on the conformity of the device. A corresponding form with references to relevant documents is available per download from www.io-link.com.

Disclaimer:

The attention of adopters is directed to the possibility that compliance with or adoption of IO-Link Community specifications may require use of an invention covered by patent rights. The IO-Link Community shall not be responsible for identifying patents for which a license may be required by any IO-Link Community specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. IO-Link Community specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

The information contained in this document is subject to change without notice. The material in this document details an IO-Link Community specification in accordance with the license and notices set forth on this page. This document does not represent a commitment to implement any portion of this specification in any company's products.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE IO-LINK COMMUNITY MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL INCLUDING, BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR PARTICULAR PURPOSE OR USE.

In no event shall the IO-Link Community be liable for errors contained herein or for indirect, incidental, special, consequential, reliance or cover damages, including loss of profits, revenue, data or use, incurred by any user or any third party. Compliance with this specification does not absolve manufacturers of IO-Link equipment, from the requirements of safety and regulatory agencies (TÜV, BIA, UL, CSA, etc.).

IO-Link ® is registered trade mark. The use is restricted for members of the IO-Link Community. More detailed terms for the use can be found in the IO-Link Community Rules on www.io-link.com.

Conventions:

In this specification the following key words (in **bold** text) will be used:

shall:	indicates a mandatory requirement. Designers **shall** implement such mandatory requirements to ensure interoperability and to claim conformity with this specification.
should:	indicates flexibility of choice with a strongly preferred implementation.
can:	indicates flexibility of choice with no implied preference (possibility and capability).
may:	indicates a permission.
highly recommended:	indicates that a feature shall be implemented except for well-founded cases. Vendor shall document the deviation within the user manual and within the manufacturer declaration.

Publisher:
IO-Link Community
c/o PROFIBUS Nutzerorganisation
Ohio Straße 8
76149 Karlsruhe
Germany

Phone: +49 721 / 98 61 97 0
Fax: +49 721 / 98 61 97 11
E-mail: info@io-link.com
Web site: www.io-link.com

Revision Log		
Version	**Date**	**Change Note / History / Reason**
D1.1.4	March 24	Draft Version D1.1.4 based on Intermediate Version I1.1.3A, covering all Change Requests 17 to 123 from AK-Test WG.
1.1.4	June 24	Version 1.1.4 based on Version D1.1.4 with Change requests 1 to 33 from Test WG

CONTENTS

0 Introduction

0.1 General

This document together with its parent IEC 61131-9 is part of a series of standards on programmable controllers and the associated peripherals and should be read in conjunction with the other parts of the series.

IEC 61131-9 specifies the Single-Drop digital Communication Interface (IO-Link™[1]) technology as a generic interface for connecting digital/analog sensors and actuators to a Master unit, which may be combined with gateway capabilities to become a fieldbus remote I/O node.

The SDCI physical interface is backward compatible with the usual 24 V I/O signalling specified in IEC 61131-2 and allows in addition digital point-to-point communication at transmission rates of 4,8 kbit/s, 38,4 kbit/s and 230,4 kbit/s.

The SDCI technology specifies parameterization, cyclic exchange of process data, and diagnosis as well as parameter Data Storage capabilities.

This subpart specifies the test cases and associated test environments for SDCI Master and Devices designed and developped according to IEC 61131-9. It provides the necessary preconditions for conformity testing to ensure interoperability and enables manufacturers of Master and Devices to sign a corresponding conformity declaration.

The structure of this document is described in clause 4.2.

Where a conflict exists between this and other IEC standards (except basic safety standards), the provisions of this standard should be considered to govern in the area of programmable controllers and their associated peripherals.

Conformity with IEC 61131-9 cannot be claimed unless the requirements of this document are fulfilled.

Terms of general use are defined in IEC 61131-1 or in the IEC 60050 series. More specific terms are defined in each part.

0.2 Patent declaration

There are no known patents related to the content of this document.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The IO-Link Community shall not be held responsible for identifying any or all such patent rights.

[1] IO-Link^{TM} is a trade name of the "IO-Link Community". Compliance to this standard does not require use of the registered logos for IO-Link^{TM}. Use of the registered logos for IO-Link^{TM} requires permission of the "IO-Link Community".

PROGRAMMABLE CONTROLLERS —

Test specification for Master and Devices according to IEC 61131-9 (Single-drop Digital Communication Interface - SDCI)

1 Scope and objectives

The single-drop digital communication interface (SDCI) technology described in part 9 of the IEC 61131 series focuses on simple sensors and actuators in factory automation, which are nowadays using small and cost-effective microcontrollers. With the help of the SDCI technology, the existing limitations of traditional signal connection technologies such as switching 0/24 V, analog 0 to 10 V, etc. can be turned into a smooth migration. Classic sensors and actuators are usually connected to a fieldbus system via input/output modules in so-called remote I/O peripherals. The (SDCI) Master function enables these peripherals to map SDCI Devices onto a fieldbus system or build up direct gateways. Thus, parameter data can be transferred from the PLC level down to the sensor/actuator level and diagnosis data transferred back in turn by means of the SDCI communication. This is a contribution to consistent parameter storage and maintenance support within a distributed automation system. SDCI is compatible to classic signal switching technology according to part 2 of the IEC 61131 series.

This subpart specifies the test cases and associated test environments for Master and Devices designed and developped according to [6]. The objectives for the specification have been to

- provide a test coverage guaranteeing interoperability of Master and Device,
- protect Master and Devices from destruction or disturbance through operation with partner devices,
- enable manufacturers of Master and Devices to sign a corresponding "manufacturer declaration".

It was not an objective, to care for

- Interoperability of Master-Tools and Master,
- Test of the specific technology of a Device (for example profile tests),
- Stress test of multiple Master Port operations,
- Compliance with standards except for EMC.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60947-5-2, *Low-voltage switchgear and controlgear – Part 5-2: Control circuit Devices and switching elements – Proximity switches*

IEC 61131-2, *Programmable controllers – Part 2: Equipment requirements and tests*

IEC 61131-9, *Programmable controllers – Part 9: Single-drop digital communication interface for small sensors and actuators (SDCI)*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions in addition to those given in IEC 61131-1, IEC 61131-2, and IEC 61131-9 apply.

3.1.1
address

part of the M-sequence control to reference data within data categories of a communication channel

3.1.2
application layer (AL)

<SDCI> part of the protocol responsible for the transmission of Process Data objects and On-Request Data objects

3.1.3
block parameter

consistent parameter access via multiple Indices or Subindices

3.1.4
checksum

<SDCI> complementary part of the overall data integrity measures in the data link layer in addition to the UART parity bit

3.1.5
CHKPDU

integrity protection data within an ISDU communication channel generated through XOR processing the octets of a request or response

3.1.6
coded switching

SDCI communication, based on the standard binary signal levels of IEC 61131-2

3.1.7
COM1

SDCI communication mode with transmission rate of 4,800 kbit/s

3.1.8
COM2

SDCI communication mode with transmission rate of 38,400 kbit/s

3.1.9
COM3

SDCI communication mode with transmission rate of 230,400 kbit/s

3.1.10
COMx

one out of three possible SDCI communication modes COM1, COM2, or COM3

3.1.11
communication error

unexpected disturbance of the SDCI transmission protocol

3.1.12
cycle time

time to transmit a frame between a Master and its Device including the following idle time

3.1.13
communication channel

logical connection between Master and Device

NOTE Four communication channels are defined: process channel, page and ISDU channel (for parameters) and diagnostic channel.

3.1.14
Device

single passive peer to a Master such as a sensor or actuator

NOTE Uppercase "Device" is used for SDCI equipment, while lowercase "Device" is used in a generic manner.

3.1.15
direct parameters
directly (page) addressed parameters transferred acyclically via the page communication channel without acknowledgement

3.1.16
dynamic parameter
part of a Device's parameter set defined by on-board user interfaces such as teach-in buttons or control panels in addition to the static parameters

3.1.17
event
an instance of a change of conditions

NOTE An event is indicated via the event flag within the Device's status cyclic information, then acyclic transfer of event data (typically diagnostics information) is conveyed through the diagnostic communication channel.

[IEC 61158-5-x, modified]

3.1.18
fallback
transition of a port from coded switching to switching signal mode

3.1.19
framing error
perturbed UART frames (physical layer)

3.1.20
interleave
segmented cyclic data exchange for process data with more than 2 octets through subsequent cycles

3.1.21
ISDU
indexed service data unit used for acyclic acknowledged transmission of parameters that can be segmented in a number of M-sequences

3.1.22
Legacy-Device
Device developed according to version V1.0 [5], the predecessor of [1]

3.1.23
Legacy-Master
Master developed according to version V1.0 [5], the predecessor of [1]

3.1.24
Master
active peer connected through ports to one up to n Devices and which provides an interface to the gateway to the upper-level communication systems or PLCs

NOTE Uppercase "Master" is used for SDCI equipment, while lowercase "Master" is used in a generic manner.

3.1.25
message
<SDCI> coherent set of data octets transferred either from a Master to its Device or vice versa following the rules of the SDCI protocol

3.1.26
M-sequence
sequence of two messages comprising a Master message and its subsequent Device message

3.1.27
M-sequence control
first octet in a Master message indicating the read/write operation, the type of the communication channel, and the address, for example offset or flow control

3.1.28
M-sequence error
unexpected or wrong message content, or no response

3.1.29
M-sequence type
one particular M-sequence format out of a set of specified M-sequence formats

3.1.30
on-request data
acyclically transmitted data upon request of the Master application consisting of parameters or event data

3.1.31
PHY-3W (IEC 61131-9 → 3-wire system)
three wire connection to Devices for power, ground, communication and/or switching signals defined in IEC 60947-5-2

3.1.32
physical layer
part of the communication protocol concerned with transmitting raw bits over a communication channel

NOTE Physical layer provides means for wake-up and fallback procedures.

3.1.33
port
communication medium interface of the Master to one Device

3.1.34
port operating mode
state of a Master's port that can be either INACTIVE, DO, DI, SDCI, or ScanMode

3.1.35
process data
input or output values from or to a discrete or continuous automation process cyclically transferred with high priority and in a configured schedule automatically after start-up of a Master

3.1.36
process data cycle
complete transfer of all process data from or to an individual Device that may comprise several cycles in case of segmentation (interleave)

3.1.37
single parameter
independent parameter access via one single Index or Subindex

3.1.38
SIO
port operation mode in accordance with digital input and output defined in IEC 61131-2 that is established after power-up or fallback or unsuccessful communication attempts

3.1.39
static parameter
part of a Device's parameter set to be saved in a Master for the case of replacement without engineering tools

3.1.40
switching signal
binary signal from or to a Device when in SIO mode (as opposed to the "coded switching" SDCI communication)

3.1.41
system management (SM)
<SDCI> means to control and coordinate the internal communication layers and the exceptions within the Master and its ports, and within each Device

3.1.42
UART frame
<SDCI> bit sequence starting with a start bit, followed by eight bits to carry a data octet, followed by an even parity bit and ending with one stop bit

3.1.43
wake-up
procedure for causing a Device to change its mode from SIO to SDCI

3.1.44
wake-up request (WURQ)
physical layer service used by the Master to initiate wake-up of a Device, and put it in a receive ready state

3.2 Symbols and abbreviated terms

Δf_{DTR}	Permissible deviation from data transfer rate, measured in %
ΔPS	Power supply ripple, measured in V
AL	Application Layer
BEP	Bit error probability
C/Q	Connection for communication (C) or switching (Q) signal (SIO)
CL_{eff}	Effective total cable capacity, measured in nF
CQ	Input capacity at C/Q connection, measured in nF
DI	Digital input
DL	Data Link Layer
DO	Digital output
DPP1	Direct Parameter Page 1
DPP2	Direct Parameter Page 2
DTU	Device tester unit
f_{DTR}	Data transfer rate, measured in bit/s
H/L	High/low signal at receiver output
ICS	Current sink for testing
I/O	Input / output
ILL	Input load current at input C/Q to V0, measured in A
IQ	Driver current in saturated operating status ON, measured in A
IQH	Driver current on high-side driver in saturated operating status ON, measured in A
IQL	Driver current on low-side driver in saturated operating status ON, measured in A
IQPK	Maximum driver current in unsaturated operating status ON, measured in A
IQPKH	Maximum driver current on high-side driver in unsaturated operating status ON, measured in A
IQPKL	Maximum driver current on low-side driver in unsaturated operating status ON, measured in A
IQQ	Quiescent current at input C/Q to V0 with inactive output drivers,

	measured in A	
IQ_{WU}	Amplitude of Master’s wake-up request current, measured in A	
IS	Supply current at V+, measured in A	
ISIR	Current pulse supply capability at V+, measured in A	
LED	Light emitting diode	
L-	Ground connection	
L+	Power supply connection	
MTU	Master tester unit	
NRZ	Non return to zero	
n_{WU}	Wake-up retry count	
On/Off	Driver’s ON/OFF switching signal	
ON-REQ	On-request data	
OVD	Signal Overload Detect	
PDCT	Port and Device configuration tool	
PL	Physical layer	
PLC	Programmable logic controller	
PLT	Physical layer test equipment	
PS	Power supply, measured in V	
PSM	Power supply of the Master ("24 V" mains or other)	
r	Time to reach a stable level with reference to the beginning of the start bit, measured in TBIT	
RL_{eff}	Loop resistance of cable, measured in Ω	
s	Time to exit a stable level with reference to the beginning of the start bit, measured in TBIT	
SDCI	Single-drop digital communication interface	
SIO	Standard Input Output (digital switching mode)	[IEC 61131-2]
SM	System Management	
t_1	Character transfer delay on Master, measured in TBIT	
t_2	Character transfer delay on Device, measured in TBIT	
t_A	Response delay on Device, measured in TBIT	
T_{BIT}	Bit time, measured in s	
t_{CYC}	Cycle time on M-sequence level, measured in s	
t_{DF}	Fall time, measured in s	
T_{DMT}	Delay time while establishing Master port communication, measured in TBIT	
t_{DR}	Rise time, measured in s	
T_{DSIO}	Delay time on Device for transition to SIO mode following wake-up request, measured in s	
T_{DWU}	Wake-up retry delay, measured in s	
$t_{M\text{-}sequence}$	M-sequence duration, measured in TBIT	
t_{idle}	Idle time between two M-sequences, measured in s	
t_H	Detection time for high level, measured in s	
t_L	Detection time for low level, measured in s	
t_{ND}	Noise suppression time, measured in s	
T_{OFS}	Temporal offset for process data processing on the Device with reference to start of cycle, measured in s	
T_{PON}	Ramp-up time following power ON, measured in s	

T_{RDL}	Wake-up readiness following power ON, measured in s
T_{REN}	Receive enable, measured in s
T_{SD}	Device detect time, measured in s
T_{WU}	Pulse duration of wake-up request, measured in s
UART	Universal asynchronous receiver transmitter
UML	Unified modelling language
V+	Voltage at L+
V0	Voltage at L-
VD-	Voltage drop on the line between the L- connections on Master and Device, measured in V
VD+	Voltage drop on the line between the L+ connections on Master and Device, measured in V
VDQ	Voltage drop on the line between the C/Q connections on Master and Device, measured in V
VHYS	Hysteresis of receiver threshold voltage, measured in V
VI	Input voltage at connection C/Q with reference to V0, measured in V
VIH	Input voltage range at connection C/Q for high signal, measured in V
VIL	Input voltage range at connection C/Q for low signal, measured in V
VRQ	Residual voltage on driver in saturated operating status ON, measured in V
VRQH	Residual voltage on high-side driver in operating status ON, measured in V
VRQL	Residual voltage on low-side driver in saturated operating status ON, measured in V
VTH	Threshold voltage of receiver with reference to V0, measured in V
VTHH	Threshold voltage of receiver for safe detection of a high signal, measured in V
VTHL	Threshold voltage of receiver for safe detection of a low signal, measured in V
WURQ	Wake-up request pulse

3.3 Conventions

3.3.1 Test case template

This document uses a dedicated template as shown in Table 1 for the particular test cases. It contains explanations on how to use items in left column.

Table 1 – Test case template

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_nnnn (nnnn = 4-digit consecutive number starting with 0001)
Name	Characteristic name of the test case (see 3.3.1.1)
Purpose (short)	Short description of the purpose of the test case (one line maximum)
Equipment under test (EUT)	Master/Master + Port or Device
Test case version	Starts with 1.0. Incremented first number indicates significant changes due to new functionality, the second one indicates changes within the test case
Category / type	See 3.3.1.2
Specification (clause)	[Bibliography, nn], clause or subclause, figure, table, chart, etc.
Configuration / setup	For example, Master-Tester-System (see Figure A.20) shall detect all transmission rates and measure the corresponding delays. It shall not react to the requests.

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Comprehensive description of the purpose of the test case (can be several lines).

TEST CASE	CONDITIONS / PERFORMANCE
	Shall not contain preconditions or instructions.
Precondition	Initial mode of the test set (both EUT and test environment) to be set prior to testing or ID of previous test. Examples: *Tester precondition/Measurement instrument pre-set* … *EUT precondition* …
Procedure	- Step by step description of the test, each step marked by characters a), b), c), etc. - Loops are possible. - "Test step macros" are possible, shall be named "TS_<domain>_xxxx", and defined within the general clause. Examples: a) Test step macro α b) Evaluation 1) c) Single instruction d) Evaluation 2) …
Test parameter	- Shall be specified using definitions within [6] - Can be identified using A), B), C), etc. - Shall be linked to procedure steps, for example a), b), c), etc. - Test loops can be used as specified in 3.3.1.3
Post condition	Final mode of the EUT and its test environment. It is possible to keep evaluation results as input for subsequent test cases if a certain test case gets too complex.

TEST CASE RESULTS	CHECK / REACTION
Evaluation	- A sequence of steps, where the status of the EUT is checked at each step - Each evaluation step is linked to a procedure step - Each evaluation step to be marked by a numeric character 1), 2), 3), etc. Example: 1) Parameter β, Parameter γ, … 2) Value λ …
Test passed	- Approve reaction at each evaluation step whether it is correct ("AND"). - In case of alternate paths are defined, they shall be approved as defined ("OR"). - Approve if deviations can be tolerated as exceptions (see [8]).
Test not passed (examples)	Describe incorrect reaction and describe the reasons for failing
Report	Create brief data of test results such as measurement values, states, Events, implementation exceptions, test exceptions (see [8]), etc., and if test passed or not passed. Data shall be sufficient for a test certificate (option).

3.3.1.1 Name of a test case

Figure 1 shows the structure of the name of a test case.

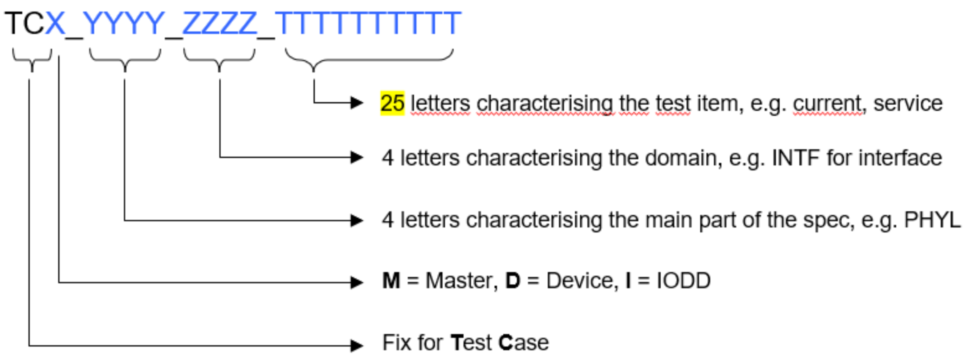

TCX_YYYY_ZZZZ_TTTTTTTTTT
25 letters characterising the test item, e.g. current, service
4 letters characterising the domain, e.g. INTF for interface
4 letters characterising the main part of the spec, e.g. PHYL
M = Master, D = Device, I = IODD
Fix for Test Case

Figure 1 – Structure of the test case name

3.3.1.2 Categories and types of test cases

-CR037-

Table 2 shows the used test case categories within this document.

Table 2 – Test case categories

Category	Definition
Master PL test	Measure port voltages, currents, and timings
Device PL test	Measure Device voltages, currents, and timings
Master/Device signal test	Transmission signal testing (eye diagram, UART frame, etc.)
Device protocol test	Device protocol test (STARTUP, PREOPERATE, OPERATE)
Device ISDU test	Device ISDU protocol test
Device Event test	Device Event handling test
Device DS test	Device Data Storage test
Device/legacy Master test	Device operation on legacy Master
Device DPP test	Test of Device's Direct Parameter page handling
Device application test	Test of Device's behavior with single parameters
Device Block parameter test	Test of Device's behavior with Block parameters
IODD parameter verification test	Test of IODD/Device matching in parameter
IODD functional system test	Test of IODD/Device matching in function
Master protocol test	Master protocol test (timings, STARTUP, PREOPERATE, OPERATE)
Master M-sequence test	Test of used Master M-sequence TYPEs (PD and OD)
Master robustness test	Test of Master's fallback and retry behavior
Master ISDU test	Test of Master's ISDU behavior (error types, limits)
Master Event test	Test of Master's Event behavior (error, notification)
Master DS test	Test of Master Data Storage mechanisms
Master legacy Device test	Test of Master behavior with legacy Device

Table 3 shows the used test case types within this document

Table 3 – Test case types

Types	Definition
Test to pass	Positive test. A function shall perform as specified. Usually, the tests of a domain are beginning with these tests, where no stress is applied.
Test to fail	Negative or stress test. A function shall react with a defined behavior, for example an error indication when boundary conditions are exceeded.

3.3.1.3 Use of test loops (TL)

One means to reduce the complexity of test case descriptions or the number of test cases is using test loops (procedure iterations).

Test loops shall be limited to variations of procedure variables listed in field "Test parameter". Branches in test case procedures shall not depend on these variables, e.g. "if variable = xyz then…". Evaluations should be independent from procedure variables.

Variations of procedure variables are specified in field "Test parameter" as follows:

Test parameter	config = {c1, c2, c3}

That means, the values c1, c2, c3 are sequentially assigned to the procedure variable "Config". Within the procedure steps, the assignment of an actual list value shall be specified as shown in the example below:

Procedure	e) … f) Initialize "config" with first value in list

	g) …

If more than one procedure variable is assigned in the same step, these variables shall be separated by a comma. A test loop can be specified as shown in the example below:

Procedure	i) … j) Repeat from g) with next value in list k) …

A complete example with assignments, evaluations, and test passed is shown below:

Procedure	a) Assign first value to "mode" b) Assign first values to "config", "voltage" c) Procedure step using "config", "voltage", "mode" if needed d) Acquire results e) Evaluation 1) f) Repeat from c) with next "config", "voltage" g) Repeat from b) with next "mode"

Test parameter	config = {c1, c2, c3},
	voltage = {18V, 24V, 30V}
	mode = {mode1, mode2}

Evaluation	1) Check results (should be independent from procedure variables)
Test passed	Result1 (mode1, config=c1, voltage=18 V) = conditions for test to pass Result2 (mode1, config=c2, voltage=24 V) = conditions for test to pass Result3 (mode1, config=c3, voltage=30 V) = conditions for test to pass Result4 (mode2, config=c1, voltage=18 V) = conditions for test to pass Result5 (mode2, config=c2, voltage=24 V) = conditions for test to pass Result6 (mode2, config=c3, voltage=30 V) = conditions for test to pass Hint: As shown above, different "test passed" conditions can be necessary if the conditions to pass a test are depending on certain procedure variables.

3.3.1.4 Handling of reports

Some test case reports require the presentation of measurement values others a flat summary. Generally, for the designer of tester equipment it is possible to always indicate a flat summary if all evaluations are positive (passed) and to show details if evaluations are negative.

3.3.2 Names of variables

Due to the possible implementation of the test cases in software, all used symbols and abbreviated terms in this document (see 3.2) are written in upper case letters without superscript or subscript.

3.3.3 Memory and transmission octet order

Figure 2 demonstrates the order that shall be used when transferring WORD based data types from memory to transmission and vice versa.

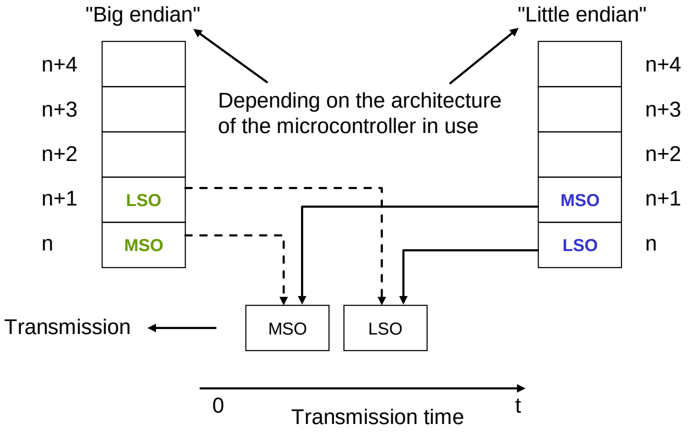


Figure 2 – Memory and transmission octet order

3.3.4 Behavioral descriptions

For the behavioral descriptions the notations of UML 2 [2] are used, mainly timing diagrams.

4 Test strategy for SDCI (IO-Link™[2]) Master and Devices

4.1 Purpose of this specification

This specification describes the test cases and specifies the necessary test equipment in conjunction with its parent document IEC 61131-9 and updates from its support organization. The functionality of Masters supporting legacy Devices according to [5] is covered also. Necessary supplements or clarifications regarding [5] are covered in Annex B.

This document provides the necessary information for the development of testers for a test suite in test laboratories.

4.2 Structure of this document

Clause 5 describes the test cases for the physical layer test of Master and Devices. They mainly require individual manual tests with variable power supplies, individual capacitive and resistive loads, voltage, and current meters as well as oscilloscopes and logic analyzers.

Clause 6 describes the test cases for the Device protocol tests, which can be performed nearly automatically via a Device-Tester-System (see Annex A.2).

Clause 7 describes the concepts of the XML schema and business rules tests for IODDs that have been shifted to [3]. Dedicated test cases verify the consistency of IODD parameters and the real parameters within the associated Device.

Clause 8 describes the test cases for the Master protocol tests, which can be performed automatically via a Master-Tester-System (see Annex A.3).

Clause 9 defines the standards for the environmental tests of Master and Devices. Annex A describes the test tools, their requirements, and the test configurations. Annex B contains a few supplementary specifications filling the gaps of [5]. Annex C provides cross reference listings for test case IDs and test case names. Annex D provides information about an SDCI support organization.

4.3 Conformity classes

4.3.1 Legacy Devices (V1.0)

Test cases for Devices designed and implemented according to [5] are no more supported by this document.

4.3.2 Devices without ISDU

Devices designed and implemented according to [6] that are not supporting the ISDU feature shall either require no parameters or provide a system conform mechanism for the Device replacement without tools. These Devices shall pass all test cases marked correspondingly in this document.

4.3.3 Devices with ISDU

Devices designed and implemented according to [6] shall pass all test cases marked correspondingly in this document. If they omit to implement the Data Storage mechanism according to [6] they still shall provide access to Index 3 and shall provide system conform means for the Device replacement without tools. These Devices shall pass all test cases marked correspondingly in this document.

[2] IO-Link™ is a trade name of the "IO-Link Community". Compliance to this standard does not require use of the registered logos for IO-Link™. Use of the registered logos for IO-Link™ requires permission of the "IO-Link Community".

4.3.4 Legacy Master

Master solely designed and implemented according to [5] are no more supported by this document. However, since Master designed and implemented according to [6] shall support "Legacy Devices", all the corresponding test cases apply.

4.3.5 Master

Master designed and implemented according to [6] shall provide all the specified features. All test cases within this document for Master apply.

4.4 Test of Devices

The test of Devices consists of four steps: Physical layer test, IODD test, protocol test, and environmental test. The protocol tests of SDCI Devices can be performed by a Device-Tester-System that shall be approved by the organization noted in Annex D. The requirements for Device-Tester-Systems are specified in Annex A.2.2.

Figure 3 shows step 1 of the Device test. It contains references to the relevant clauses in this specification and consists of a visual check and manually performed measurements.

If the Device shows specific connectors, cables, or color codings, these deviations shall be documented within the user manual with respect to the original definitions in the standard [6].

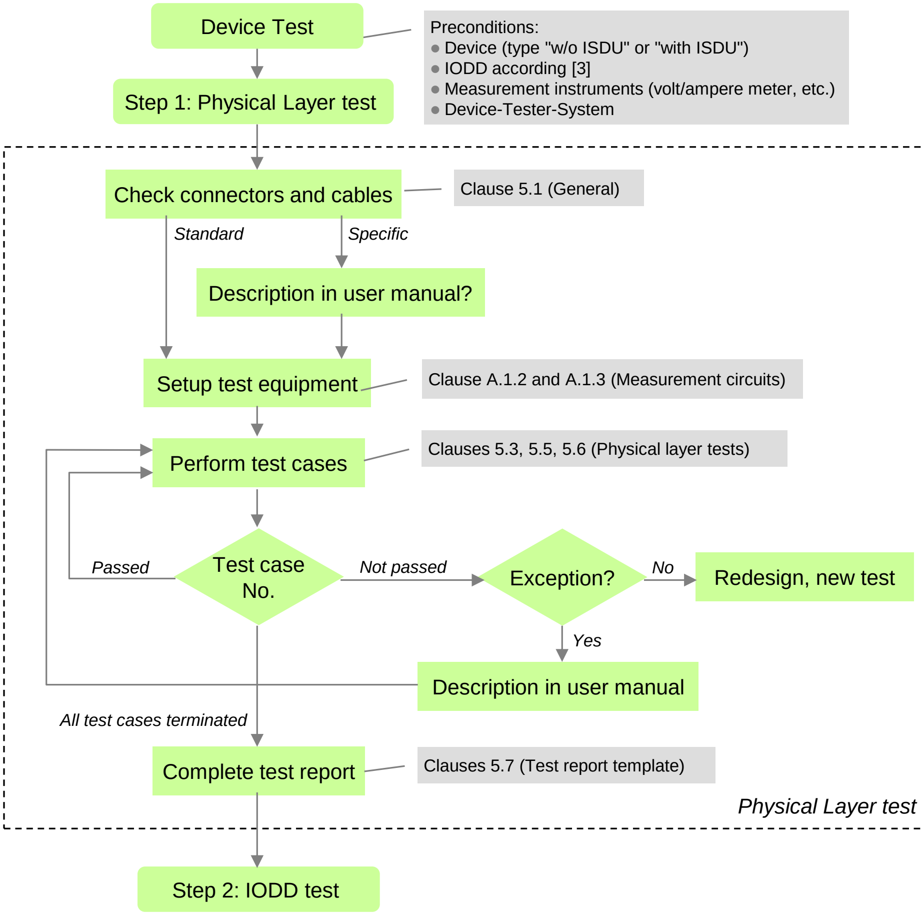

Device Test
Preconditions:
● Device (type "w/o ISDU" or "with ISDU")
● IODD according [3]
● Measurement instruments (volt/ampere meter, etc.)
● Device-Tester-System
Step 1: Physical Layer test
Check connectors and cables
Clause 5.1 (General)
Standard
Specific
Description in user manual?
Setup test equipment
Clause A.1.2 and A.1.3 (Measurement circuits)
Perform test cases
Clauses 5.3, 5.5, 5.6 (Physical layer tests)
Passed
Test case No.
Not passed
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Complete test report
Clauses 5.7 (Test report template)
Physical Layer test
Step 2: IODD test

Figure 3 – Step 1 of the Device test sequence (PL)

If a Device did not pass a certain test case due to measurement values close to the tolerance limits or similar situations it is possible to send an informal request to the organization listed in Annex D. This request shall be comprehensive enough for the experts to allow for an exception under certain conditions or clarification of the specifications. If an exception applies, it shall be documented in the user manual mentioning the possible implications (see [8] for details).

Instead of pursuing such a time-consuming and uncertain way, it is highly recommended for the applicant to rather strive for a robust implementation and conformity of the Device.

Figure 4 shows step 2 of the Device test. It contains references to the relevant clauses in this specification and consists of an IODD-Test with a so-called IODD-Checker-Tool (7.2) and a parameter verification test with the help of the protocol test (Device-Tester-System).

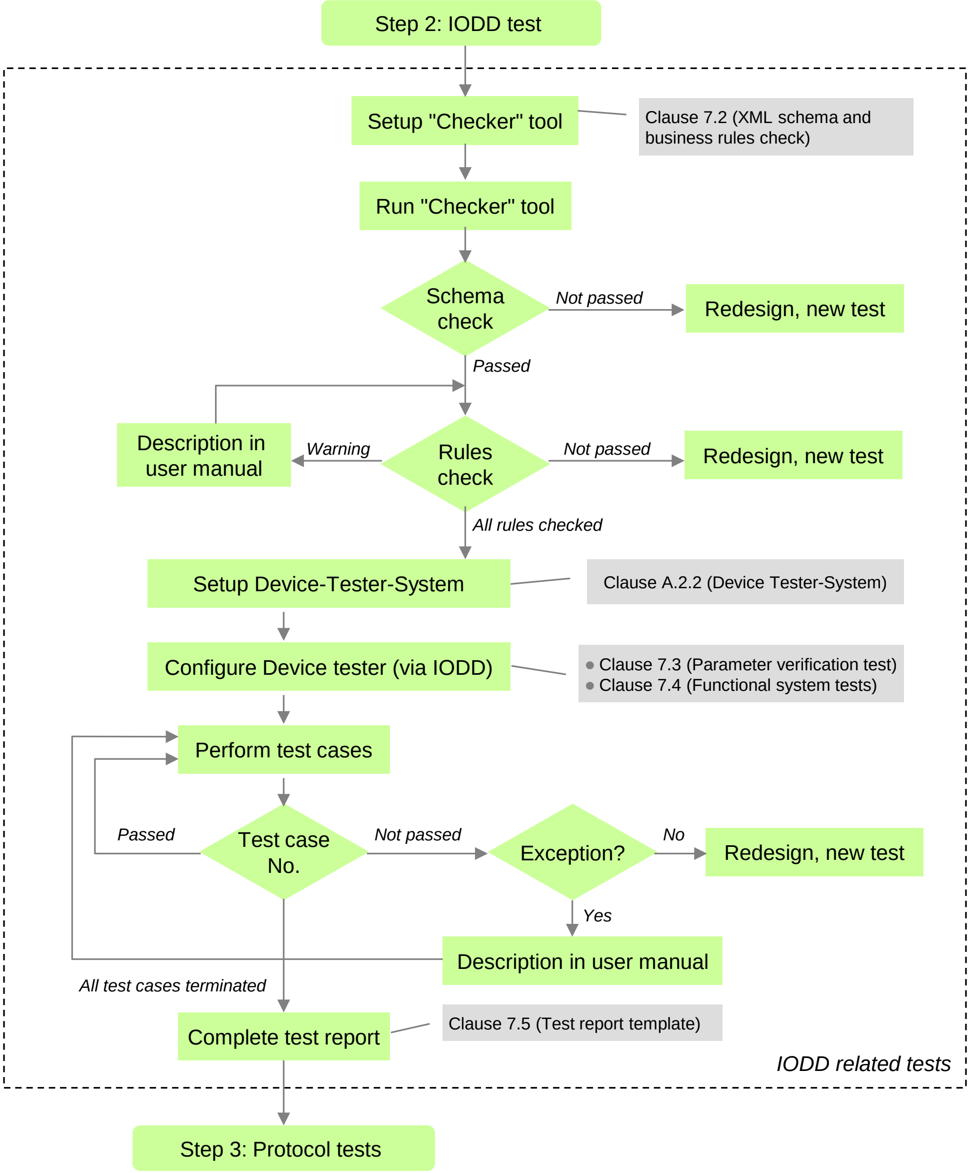

Step 2: IODD test
Setup "Checker" tool
Clause 7.2 (XML schema and business rules check)
Run "Checker" tool
Schema check
Not passed
Redesign, new test
Passed
Description in user manual
Warning
Rules check
Not passed
Redesign, new test
All rules checked
Setup Device-Tester-System
Clause A.2.2 (Device Tester-System)
Configure Device tester (via IODD)
● Clause 7.3 (Parameter verification test)
● Clause 7.4 (Functional system tests)
Perform test cases
Passed
Test case No.
Not passed
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Complete test report
Clause 7.5 (Test report template)
IODD related tests
Step 3: Protocol tests

Figure 4 – Step 2 of the Device test sequence (IODD)

Figure 5 shows step 3 of the Device test. It contains references to the relevant clauses in this specification and consists of an automated protocol test with the help of the Device-Tester-System defined in A.2.2.

Two different sets of test cases are necessary to adjust the tests for the two Device conformity classes: Devices without the ISDU feature developed according to [6], and Devices with ISDU support developed according to [6].

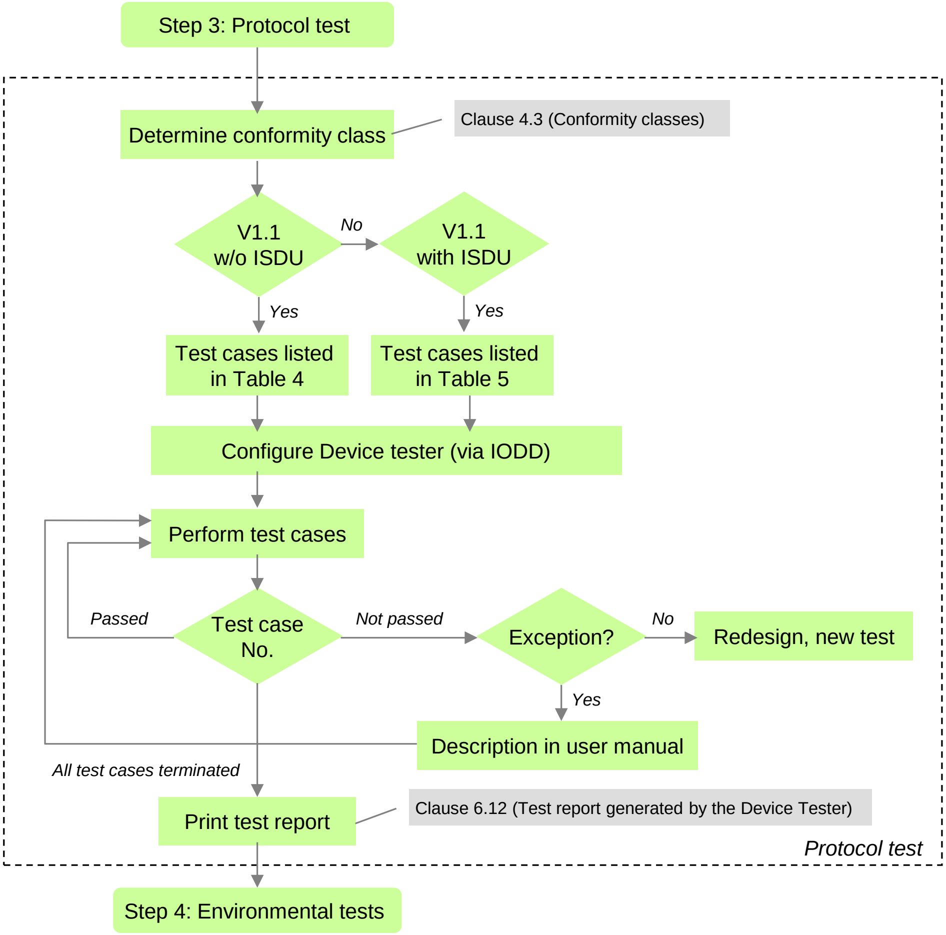

Step 3: Protocol test
Determine conformity class
Clause 4.3 (Conformity classes)
V1.1 w/o ISDU
No
V1.1 with ISDU
Yes
Yes
Test cases listed in Table 4
Test cases listed in Table 5
Configure Device tester (via IODD)
Perform test cases
Passed
Test case No.
Not passed
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Print test report
Clause 6.12 (Test report generated by the Device Tester)
Protocol test
Step 4: Environmental tests

Figure 5 – Step 3 of the Device test sequence (protocol)

-CR043- -CR070- -CR022- -CR066- -CR094- -CR036-

The set of test cases for Devices without ISDU support is defined in Table 4.

Table 4 – Set of test cases for Devices without ISDU support

Major feature	Test cases	Remarks
STARTUP	TC_0034 to TC_0038, TC_0306	New TC_0306 for test of revision management
PREOPERATE	TC_0039 to TC_0041 TC_0043 to TC_0044 TC_0374, TC_0378	TC_0042 removed from list
OPERATE	TC_0045 to TC_0047 TC_0049 TC_0312 and TC_0313 TC_0373, TC_0375, TC_0376	TC_0048 and TC_0051 removed from list
Events	TC_0069 to TC_0076	Exceptions exist (see 6.6.1)
Legacy Master (V1.0)	TC_0085 to TC_0087	If restricted to 134 ms; exceptions for PDInvalid exist (see Table 112)
Direct Parameter page 1	TC_0089 to TC_0097 TC_0101	TC_100 removed

The set of test cases for Devices with ISDU support is defined in Table 5.

Table 5 – Set of test cases for Devices with ISDU support

-CR005- -CR102- -CR043- -CR070- -CR022- -CR066- -CR094- -CR036-

Major feature	Test cases	Remarks
STARTUP	TC_0034 to TC_0038 TC_0306	New TC_0306 for test of revision management
PREOPERATE	TC_0039 to TC_0041 TC_0043 to TC_0044 TC_0377 TC_0374, TC_0378	TC_0042 removed from list
OPERATE	TC_0045 to TC_0047 TC_0049 TC_0312 and TC_0313 TC_0373 , TC_0375 , TC_0376	TC_0048 and TC_0051 removed from list.
ISDU	TC_0052 to TC_0068 TC_0309	New TC_0309 for repeated ISDU segments
Events	TC_0069 to TC_0076	Exceptions exist (see 6.6.1)
Data Storage	TC_0077 to TC_0084 TC_0321 to TC_0322	New TC_0321/322 for Parameter manager, upload flag
Legacy Master (V1.0)	TC_0085 to TC_0087	If compatible; exceptions for PDInvalid exist (see Table 112)
Direct Parameter page 1	TC_0089 to TC_0097 TC_0101	TC_0100 removed
Predefined parameters	TC_0104 TC_0323 TC_0107 to TC_0124 TC_0128 to TC_0129 TC_0131 to TC_0134 TC_0136 to TC_0137 TC_0140 to TC_0142	TC_0105 removed TC_0323 for DS termination marker TC_0130 removed TC_0135 removed TC_0136/0137 for profile parameter
Block parameter	TC_0143 to TC_0148 TC_0324 to TC_0330	TC_0146 reactivated New TC_0324 to TC_0330 for Parameter manager
IODD based tests	TC_0149 to TC_0152 TC_0157 TC_0155 to TC_0156 TC_0314 to TC_0318	IODD parameter verification New for functional system tests (reset, locking, etc.)

Figure 6 shows step 4 of the Device test. It contains references to the relevant clauses in this specification and consists of an EMC test defined in [6] and possible tests according to relevant product standards such as for example the IEC 60947 series. A successfully terminated Device test can be completed by a manufacturer declaration (see Annex D).

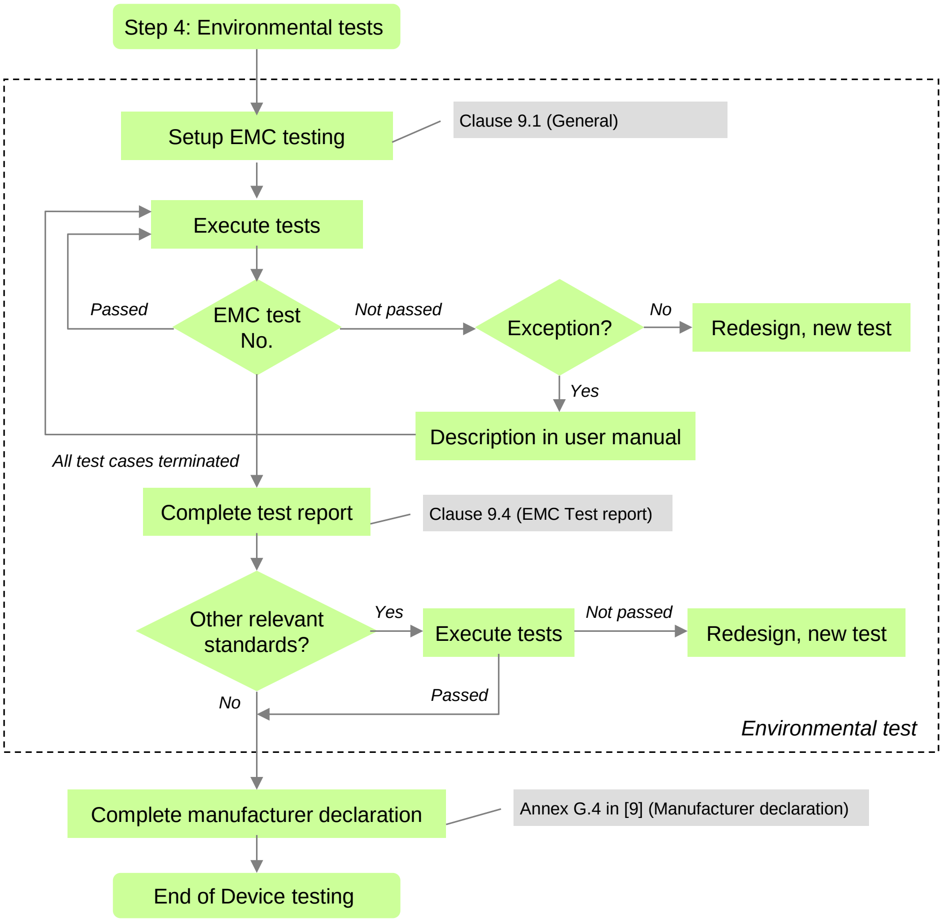


Figure 6 – Step 4 of the Device test sequence (environment)

4.5 Test of Masters

The test of Masters consists of four steps: Physical layer test, protocol test, IODD interpreter test, and environmental test. The protocol tests of Master can be performed by a Master-Tester-System approved by the organization noted in Annex D. The requirements for Master-Tester-Systems are specified in Annex A.3.

Figure 7 shows step 1 of the Master test sequence. It contains references to the relevant clauses in this specification and consists of a visual check and manually performed measurements.

If the Master shows specific connectors, cables, or color codings, these deviations shall be documented within the user manual in respect to the original definitions in the standard [6].

If a Master did not pass a certain test case due to measurement values close to the tolerance limits or similar situations it is possible to send an informal request to the organization listed in Annex D. This request shall be comprehensive enough for the experts to allow for an exception under certain conditions or clarification of the specifications. If an exception applies, it shall be documented in the user manual mentioning the possible implications (see [8] for details).

Instead of pursuing such a time-consuming and uncertain way, it is highly recommended for an applicant to rather strive for a robust implementation and conformity of the Master.

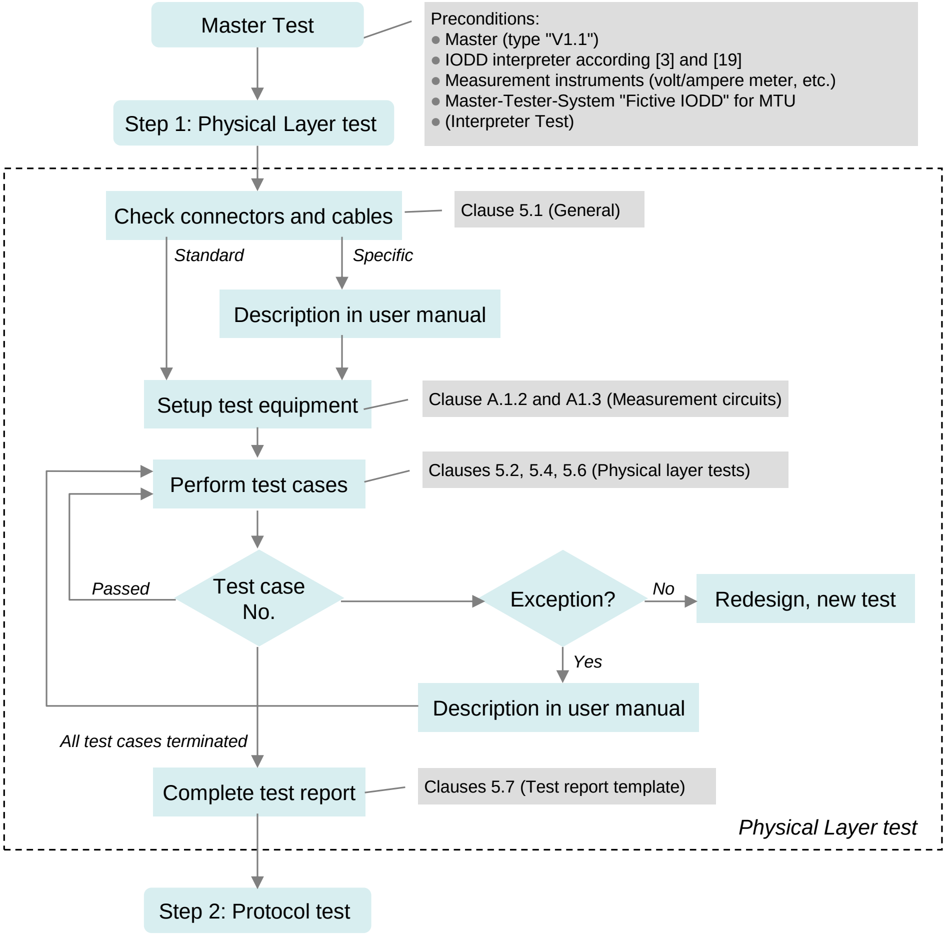
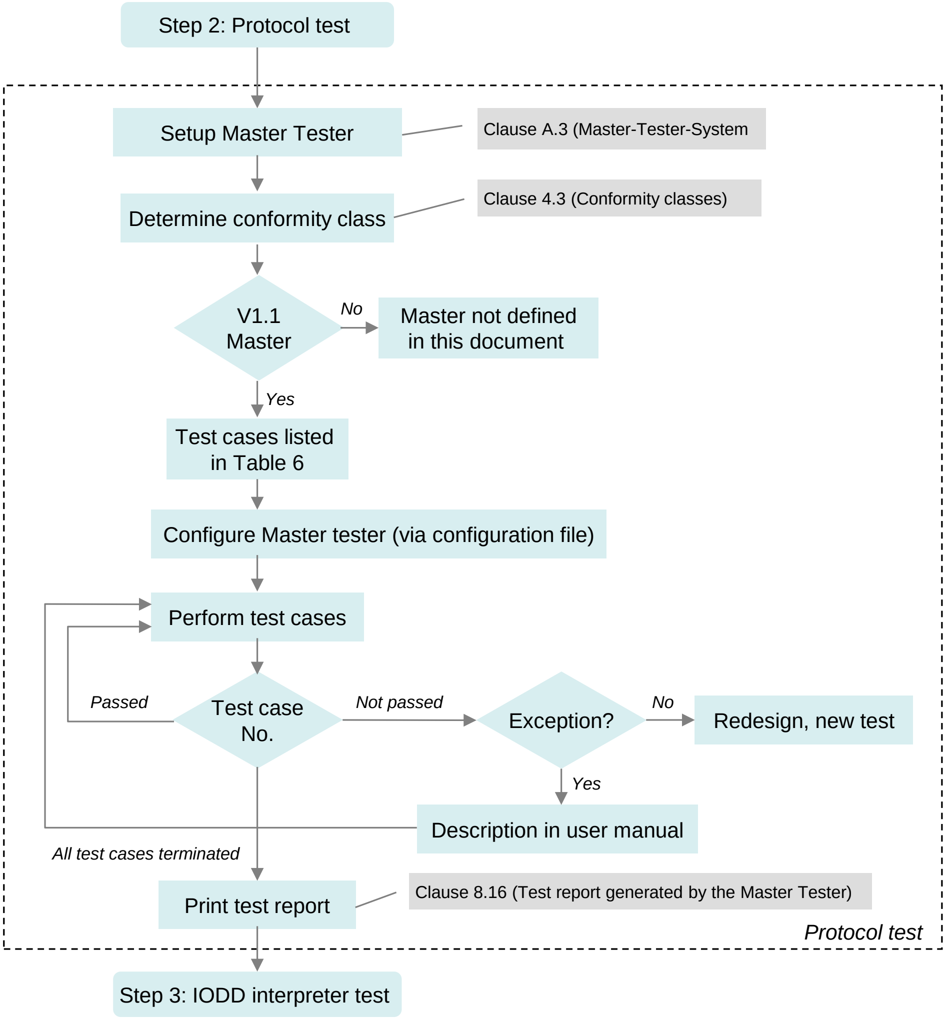

Master Test
Preconditions:
● Master (type "V1.1")
● IODD interpreter according [3] and [19]
● Measurement instruments (volt/ampere meter, etc.)
● Master-Tester-System "Fictive IODD" for MTU
● (Interpreter Test)
Step 1: Physical Layer test
Check connectors and cables
Clause 5.1 (General)
Standard
Specific
Description in user manual
Setup test equipment
Clause A.1.2 and A1.3 (Measurement circuits)
Perform test cases
Clauses 5.2, 5.4, 5.6 (Physical layer tests)
Passed
Test case No.
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Complete test report
Clauses 5.7 (Test report template)
Physical Layer test
Step 2: Protocol test

Figure 7 – Step 1 of the Master test sequence (PL)

Figure 8 shows step 2 of the Master test sequence. It contains references to the relevant clauses in this specification and consists of an automated protocol test with the help of a Master-Tester-System defined in Annex A.3.


Step 2: Protocol test
Setup Master Tester
Clause A.3 (Master-Tester-System
Determine conformity class
Clause 4.3 (Conformity classes)
V1.1 Master
No
Master not defined in this document
Yes
Test cases listed in Table 6
Configure Master tester (via configuration file)
Perform test cases
Passed
Test case No.
Not passed
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Print test report
Clause 8.16 (Test report generated by the Master Tester)
Protocol test
Step 3: IODD interpreter test

375

376 **Figure 8 – Step 2 of the Master test sequence (PL)**

377 -CR050-

378 The set of test cases for Masters (V1.1) is defined in Table 6.

379 **Table 6 – Set of test cases for Masters (V1.1)**

Major feature	Test cases	Remarks
Timings	TC_0158 to TC_0167 TC_0331 to TC_0333	New TC_0331ff: delay time of Master messages
Process Data (PD)	TC_0168 to TC_0173 TC_0298 TC_0334 to TC_0335	TC_0174/0175 removed New TC_0334ff: TYPE_2_V
On-request Data (OD)	TC_0179 to TC_0182 TC_0288	New TC_0288 for Idle after ISDU service
STARTUP	TC_0183 to TC_0194 TC_0307	TC_0195 to TC_0201 removed New TC_0307 for test of revision management
PREOPERATE	TC_0202 to TC_0209	
OPERATE	TC_0336 to TC_0337 TC_0210 to TC_0212	New TC_0336ff: TYPE_1_V
Fallback	TC_0213 to TC_0216	TC_0213 and TC_0214 are optional
Retry	TC_0338 to TC_0349 TC_0222	New approach: TC_0217 to TC_0220 removed
ISDU (application errors)	TC_0223 to TC_0235 TC_0237 to TC_0238	TC_0236 removed

Major feature	Test cases	Remarks
ISDU (derived errors)	TC_0239 to TC_0242 TC_0244	
ISDU (Limit checks)	TC_0243 TC_0245 to TC_0246 TC_0248 to TC_0255	
Events	TC_0256 TC_0261 to TC_0262 TC_0308 TC_0263 to TC_0267 TC_0350 to TC_0353	New TC_0261ff for Events with details New TC_0308 for Notifications TC_0268/0269 removed New TC_0350ff for Events outside OPERATE
Data Storage	TC_0354 to TC_0372	New approach: TC_0270 to TC_0284 removed
Legacy Device (V1.0)	TC_0285 to TC_0290	TC_0291 removed

Figure 9 shows step 3 of the Master test sequence. It contains references to the relevant clauses in this specification. The Master-Tester-System defined in Annex A.3 provides a "Fictive IODD" (see 7.2.2) in accordance with [3]. The user manual of the tool demonstrates how this IODD shall be presented to the user with respect to completeness and correctness.

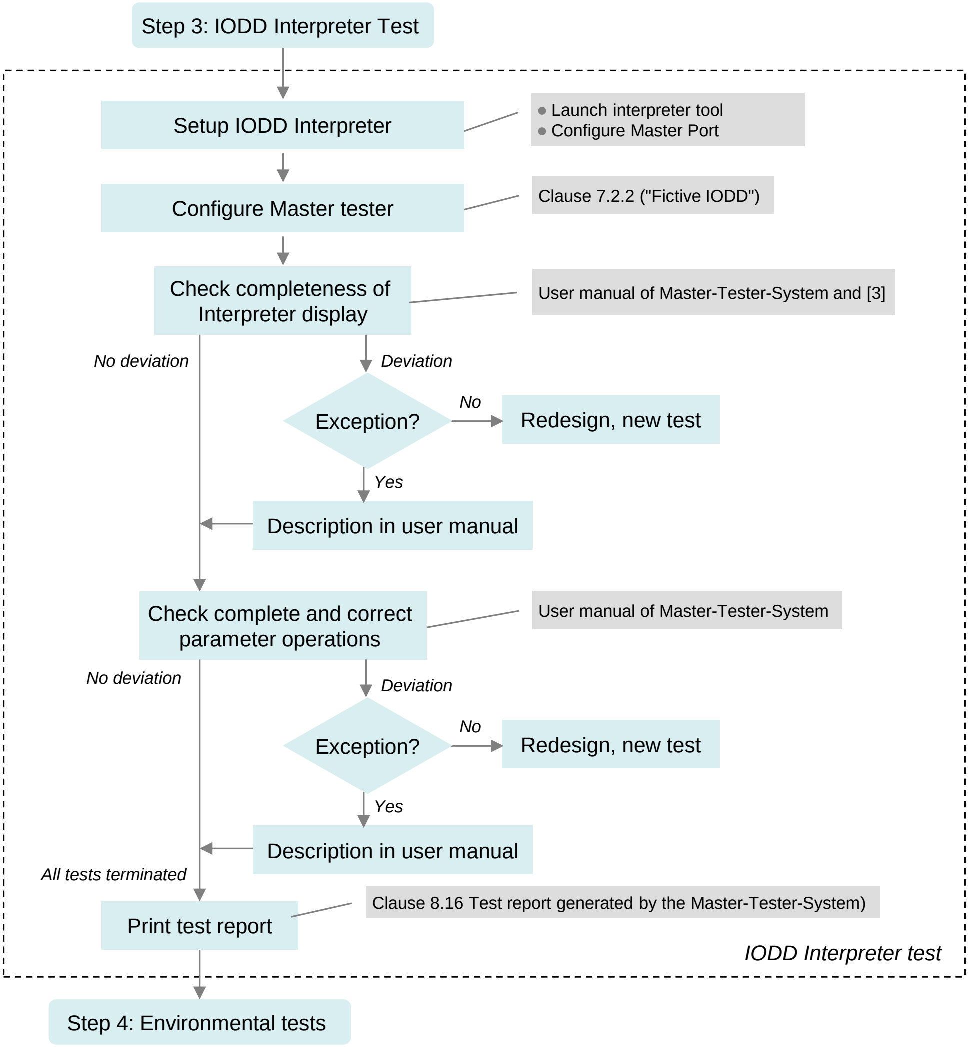

Step 3: IODD Interpreter Test
Setup IODD Interpreter
● Launch interpreter tool
● Configure Master Port
Configure Master tester
Clause 7.2.2 ("Fictive IODD")
Check completeness of Interpreter display
User manual of Master-Tester-System and [3]
No deviation
Deviation
Exception?
No
Redesign, new test
Yes
Description in user manual
Check complete and correct parameter operations
User manual of Master-Tester-System
No deviation
Deviation
Exception?
No
Redesign, new test
Yes
Description in user manual
All tests terminated
Print test report
Clause 8.16 Test report generated by the Master-Tester-System)
IODD Interpreter test
Step 4: Environmental tests

Figure 9 – Step 3 of the Master test sequence (Interpreter)

Figure 10 shows step 4 of the Master test. It contains references to the relevant clauses in this specification and consists of an EMC test defined in [6] and possible tests according to relevant product standards such as for example the IEC 61131-2. A successfully terminated Master test can be completed by a manufacturer declaration (see Annex D).

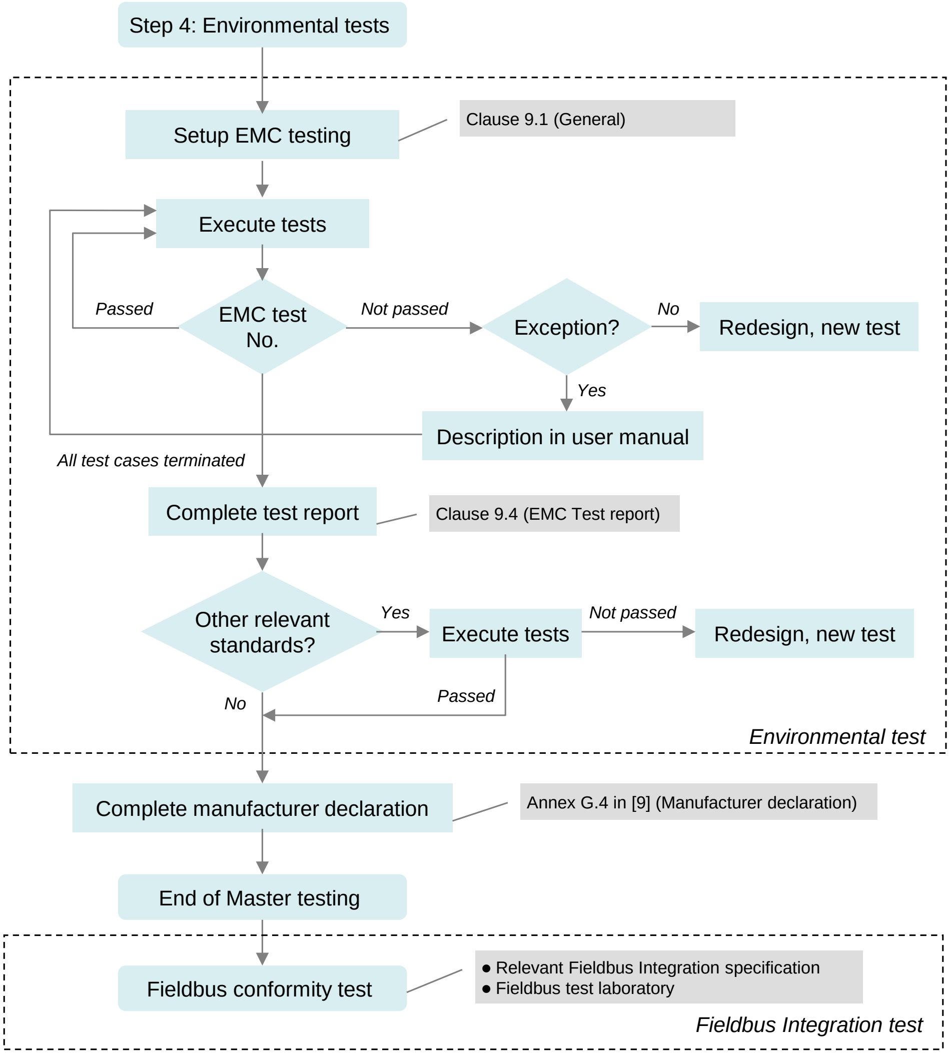

Step 4: Environmental tests
Setup EMC testing
Clause 9.1 (General)
Execute tests
Passed
EMC test No.
Not passed
Exception?
No
Redesign, new test
Yes
Description in user manual
All test cases terminated
Complete test report
Clause 9.4 (EMC Test report)
Other relevant standards?
Yes
Execute tests
Not passed
Redesign, new test
Passed
No
Environmental test
Complete manufacturer declaration
Annex G.4 in [9] (Manufacturer declaration)
End of Master testing
Fieldbus conformity test
● Relevant Fieldbus Integration specification
● Fieldbus test laboratory
Fieldbus Integration test

Figure 10 – Step 4 of the Master test sequence (environment)

5 Physical Layer (PL) tests

5.1 General

The physical layer tests comprise a visual inspection of the type of connector, cable, maximum cable length and color coding of the wires. If customer specific or region specific connectors, cable, and the color coding of the wires deviate from the specifications in [6], the user manual of the Device shall document clearly the differences in comparison with the definitions in the standard.

The physical layer tests comprise also measurements of voltage levels and currents as well as timing limits, slopes and line and message signals. The necessary measurement instruments are defined in Annex A.1.1 and the necessary measurement circuits are defined in Annex A.1.2.

Physical layer tests are carried out at room temperature (15 °C to 35 °C). However, the tests shall pass within the whole operating temperature range for a Device specified by the manufacturer.

5.2 Static parameters of the Master interface

5.2.1 DC supply current capability of Master Port L+

Table 7 defines the test conditions for this test case.

Table 7 – DC supply current capability of Master Port L+

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0001
Name	TCM_PHYL_INTF_VSMISMBEHAV
Purpose (short)	DC supply voltage and current capability of Master Port
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6
Configuration / setup	Variable Master input voltage PSM and variable current sink according to Figure A.3

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	DC supply voltage and current capability of Master Port L+ shall be within specified boundaries. Behavior of both quantities shall be monitored for a certain time.
Precondition	PLT: Current sink between L+ and L- EUT: PORT_DI (see A.4.2)
Procedure	a) Apply first PSM value to Master *;Test parameter* b) Adjust current sink to ISM value *;Test parameter* c) Monitor VSM and ISM at L+ for 30 s d) Evaluation 1) e) Repeat test with next PSM value and ISM value from c)
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual) ISM = {ISMmax, 0 mA} (ISMmax according to user manual)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check VSM and ISM flow
Test passed	20 V ≤ VSM ≤ 30 V, AND ISM is constantly flowing
Test not passed (examples)	VSM < 20 V, OR VSM > 30 V, OR ISM is interrupted
Report	VSM (PSMmin): <value> <ok | nok> VSM (PSMmax): <value> <ok | nok> Steady current flow: <yes/no> <ok | nok>

5.2.2 Power-On supply current capability of Master Port L+

Table 8 defines the test conditions for this test case.

Table 8 – Power-On supply current capability of Master Port L+

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0002
Name	TCM_PHYL_INTF_ISIRM
Purpose (short)	Power-on/Port turn-on supply current capability at Master Port
Equipment under test (EUT)	Master (see Annex B.2).
Test case version	1.3
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6
Configuration / setup	Supply current at Master Port is monitored according to Figure A.3

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Driver capability of the Master Port L+ supply at power-on or switch-on of Port L+. Monitor supply current drawn from the Master Port.
Precondition	PLT: Current sink between L+ and L-; current sink shall be configured to consume ≥ 1000 mA (ISIRM) EUT: PORT_INACTIVE (see A.4.2)
Procedure	a) Set supply voltage of Master to first value of PSM *;Test parameter* b) Exemplary: SMI_PortPowerOffOn(ABPS_PORTPOWERON) (see A.4.6) c) Identify time tmin, where ISM reaches ISIRMmin = 400mA d) Monitor current ISIRM at L+ after tmin for 50 ms e) Evaluation 1) f) Repeat test from step b) with next PSM value *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Evaluate charge QMI using the formula $QMI = \int_{tmin}^{tmin+50ms} ISM(t)dt$
Test passed	Charge QMI ≥ 20 mAs
Test not passed (examples)	Charge QMI < 20 mAs (see Annex B.2)
Report	QMI (PSMmin): <value> <ok | nok> QMI (PSMmax): <value> <ok | nok>

5.2.3 Load current at Master Port C/Q

Table 9 defines the test conditions for this test case.

Table 9 – Load current at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0003
Name	TCM_PHYL_INTF_ILLM
Purpose (short)	Load current at C/Q of Master Port
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6
Configuration / setup	The input current at C/Q of the Master Port is monitored.

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Quiescent current at Master Port C/Q in input mode. Monitor current flowing into C/Q.
Precondition	PLT: Voltage source with value VIM between L- and C/Q EUT: PORT_DI
Procedure	a) Set supply voltage of Master to first PSM value *;Test parameter* b) Measure VSM c) Set voltage VIM = 5V d) Measure current ILLM into C/Q e) Evaluation 1) f) Set voltage VIM = 5,1 V g) Measure current ILLM into C/Q h) Evaluation 2) i) Set voltage VIM = 15V j) Measure current ILLM into C/Q k) Evaluation 3) l) Set voltage VIM = measured value of VSM in b) m) Measure current ILLM into C/Q n) Evaluation 4) o) Repeat from b) with next PSM value *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ILLM < 15 mA 2) Check 2 mA < ILLM < 15 mA 3) Check 2 mA < ILLM < 15 mA 4) Check 2 mA < ILLM < 15 mA
Test passed	All checks OK
Test not passed (examples)	One of the checks above failed
Report	ILLM (VIM = 5 V, PSMmin): <value> <ok | nok> ILLM (VIM = 5,1 V, PSMmin): <value> <ok | nok> ILLM (VIM = 15 V, PSMmin): <value> <ok | nok> ILLM (VIM = measured value of VSM, PSMmin): <value> <ok | nok> ILLM (VIM = 5 V, PSMmax): <value> <ok | nok> ILLM (VIM = 5,1 V, PSMmax): <value> <ok | nok> ILLM (VIM = 15 V, PSMmax): <value> <ok | nok> ILLM (VIM = measured value of VSM, PSMmax): <value> <ok | nok>

5.2.4 High-side residual voltage at Master Port C/Q

Table 10 defines the test conditions for this test case.

Table 10 – High-side residual voltage at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0004
Name	TCM_PHYL_INTF_VRESHIGH
Purpose (short)	Static high-side driver capability
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6
Configuration / setup	The output level at Master C/Q output is measured

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Driver capability of the Master Port high-side driver. Measurement of the voltage drop between supply L+ and C/Q output.
Precondition	PLT: Current sink IQHMmin between C/Q and L-; Voltage measurement for VRQHM between L+ and C/Q EUT: PORT_DO (see A.4.2)
Procedure	a) Set supply voltage of Master to first PSM value *;Test parameter* b) SMI_PDOut(ABPS_DO_HIGH) (see A.4.6) c) Measure VRQHM d) Evaluation 1) e) Repeat from b) with next PSM value *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check VRQHM ≤ 3V
Test passed	All checks OK
Test not passed (examples)	Any of the checks above failed
Report	VRQHM (PSMmin): <value> <ok | nok> VRQHM (PSMmax): <value> <ok | nok>

5.2.5 Low-side residual voltage at Master Port C/Q

Table 11 defines the test conditions for this test case.

Table 11 – Low-side residual voltage at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE	
Identification (ID)	SDCI_TC_0005	
Name	TCM_PHYL_INTF_VRESLOW	
Purpose (short)	Static low-side driver capability	
Equipment under test (EUT)	Master	
Test case version	1.1	
Category / type	Master PL test: test to pass	
Specification (clause)	[6], see 5.3.2.3, Table 6	
Configuration / setup	The output level at Master Port C/Q output is measured	
TEST CASE	**CONDITIONS / PERFORMANCE**	
Purpose (detailed)	Driver capability of the Master Port low-side driver. Measurement of the voltage drop between C/Q output and supply L-.	
Precondition	PLT: Current source IQHMmin between L+ and C/Q; Voltage measurement for VRQLM between C/Q and L- EUT: PORT_DO (see A.4.2)	
Procedure	a) Set supply voltage of Master to first PSM value b) SMI_PDOut(ABPS_DO_LOW) (see A.4.6) c) Measure VRQLM d) Evaluation 1) e) Repeat from b) with next PSM value	*;Test parameter* *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)	
Post condition	–	
TEST CASE RESULTS	**CHECK / REACTION**	
Evaluation	1) Check VRQLM ≤ 3V	
Test passed	All checks OK	
Test not passed (examples)	Any of the checks above failed	
Report	VRQLM (PSMmin): <value> VRQLM (PSMmax): <value>	<ok | nok> <ok | nok>

5.2.6 High-level input threshold voltage at Master Port C/Q

Table 12 defines the test conditions for this test case.

Table 12 – High-level input threshold voltage at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0006
Name	TCM_PHYL_INTF_VTHHM
Purpose (short)	Test of static input high-level threshold at C/Q
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	The digital input signal for C/Q input is monitored (see Figure A.6)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measurement of the threshold voltage for high-level at the C/Q
Precondition	PLT: Voltage source with value VIM between C/Q and L- EUT: PORT_DI (see A.4.2)
Procedure	a) Set supply voltage of Master to first PSM value *;Test parameter* b) Sweep voltage VIM at C/Q from 5 V to 15 V in steps of maximum 0,1 V c) Exemplary: Repeat SMI_PDIn until DI_C/Q = 1; (see Reference for SMI: [6], 11.2.17, Figure 113) d) Measure VIM Transition e) Evaluation 1) f) Repeat from b) with next PSM value *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	Memorize VIM at DI transition 0→1 (all PSM)

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check voltage VIM at DI transition "low" to "high"
Test passed	All checks: 10,5 V < VIM < 13 V (range of VTHHM)
Test not passed (examples)	One of the checks failed
Report	VIM @ Transition 0→1 (PSMmin): <value> <ok | nok> VIM @ Transition 0→1 (PSMmax): <value> <ok | nok>

5.2.7 Low-level input threshold voltage at Master Port C/Q

Table 13 defines the test conditions for this test case.

Table 13 – Low-level input threshold voltage at Master Port C/Q

-CR118-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0007
Name	TCM_PHYL_INTF_VTHLM
Purpose (short)	Static input low-level threshold at C/Q
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	The digital input signal for C/Q input is monitored (see Figure A.6)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measurement of the threshold voltage for low-level at C/Q.
Precondition	PLT: Voltage source with value VIM between and C/Q and L- EUT: PORT_DI (see A.4.2)
Procedure	a) Set supply voltage of Master to first PSM value *;Test parameter* b) Sweep voltage VIM at C/Q from 15 V to 5 V in steps of maximum 0,1 V c) Examplery: Repeat SMI_PDIn until DI_C/Q = 0; (see Reference for SMI: [6], 11.2.17, Figure 113) d) Measure VIM Transition e) Evaluation 1) f) Repeat from b) with next PSM value *;Test parameter*
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	Memorize VIM at DI transition 1→0 (all PSM)

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check voltage VIM at DI transition "high" to "low"
Test passed	For all checks: 8.0 V < VIM < 11,5 V (range of VTHLM)
Test not passed (examples)	One of the checks failed
Report	VIM @ Transition 1→0 (PSMmin): <value> <ok | nok> VIM @ Transition 1→0 (PSMmax): <value> <ok | nok>

5.2.8 Input hysteresis voltage at Master Port C/Q

Table 14 defines the test conditions for this test case.

Table 14 – Input hysteresis voltage at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0008
Name	TCM_PHYL_INTF_VHYSM
Purpose (short)	Calculation of input hysteresis at C/Q
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	See TC_0006 and TC_0007

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Calculation of the hysteresis voltage at C/Q.
Precondition	Value VIM(TC_0006) from TC_0006 is available for all PSM values Value VIM(TC_0007) from TC_0007 is available for all PSM values
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all PSM values: VHYSM = Value VIM(TC_0006) – Value VIM(TC_0007)
Test passed	For all PSM values: Voltage VHYSM is ≥ 0 V
Test not passed (examples)	For any PSM value: Voltage VHYSM is < 0 V
Report	VHYSM (PSMmin): <value> <ok | nok> VHYSM (PSMmax): <value> <ok | nok>

5.2.9 Permissible voltage range at Master Port C/Q

Table 15 defines the test conditions for this test case.

Table 15 – Permissible voltage range at Master Port C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0299
Name	TCM_PHYL_INTF_VOLTRANGECQ
Purpose (short)	Permissible voltage range at Master C/Q
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5, VIL and VIH
Configuration / setup	See Precondition

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The receiver behavior at signal voltages equal to both the negative and positive supply voltage is tested.
Precondition	PLT: Resistor between L+ and C/Q or C/Q and L- respectively EUT: PORT_DI (see A.4.2)
Procedure	a) Set supply voltage of Master to first PSM value *;Test parameter* b) Apply resistor value between L+ and C/Q temporarily for at least 1 min c) Apply resistor value between C/Q and L- temporarily for at least 1 min d) Attach a Device e) Set Master to communication, for example: SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) (see A.4.6) f) Get Master state, for example: SMI_PortStatus *;returns ArgBlock "PortStatusList"* g) Evaluation 1) h) Repeat from b) with next PSM value
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual) Resistor = {1 Ohm}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Argblock
Test passed	For all PSM values: PortStatusList.PortstatusInfo = OPERATE
Test not passed (examples)	For any PSM value: State OPERATE not reached
Report	Communication established: <yes/no> <ok | nok>

5.2.10 Electrical isolation of Master Port class B

Table 16 defines the test conditions for this test case.

Table 16 – Electrical isolation of Master Port class B

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0310
Name	TCM_PHYL_INTF_MRGI
Purpose (short)	Electrical isolation of Master Port class B power lines
Equipment under test (EUT)	Master with Port class B
Test case version	1.0
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.4.2
Configuration / setup	See Figure A.1
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test of electrical isolation between Power 1 and Power 2 for a Class B Master Port measured with isolation test voltages.
Precondition	PLT: See Figure A.1 EUT: PORT_DI (see A.4.2)
Procedure	a) Set PSM to first value *;Test parameter* b) Set Vaux to first value *;Test parameter* c) Measure IgiM0 and voltage VL+ at L+ d) Evaluation 1) e) Set PSM to second value *;Test parameter* f) Set Vaux to second value *;Test parameter* g) Measure IgiM1 h) Evaluation 2)
Test parameter	PSM = {PSMmax, 0 V} according to user manual Vaux = {0 V, 30 V}
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check RIsoM0 = VL+/IgiM0 2) Check RIsoM1 = 30 V/IgiM1
Test passed	RIsoM0 > 2 MOhm, and RIsoM1 > 2 MOhm
Test not passed (examples)	Any evaluation failed
Report	RIsoM0: <value> <ok | nok> RIsoM1: <value> <ok | nok>

5.3 Static parameters of the Device interface

5.3.1 Power supply current consumption of the Device I

Table 17 defines the test conditions for this test case.

Table 17 – Power supply current consumption of the Device I

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0011
Name	TCD_PHYL_INTF_ISD
Purpose (short)	Static power supply current consumption
Equipment under test (EUT)	Device
Test case version	1.3
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6
Configuration / setup	The supply current at Device L+ is measured
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Measurement of the static power supply current at the L+ of the Device
Precondition	PLT: Voltage source with value VSD between L+ and L- EUT: Device set to operation mode with maximum power consumption. Maximum current consumption ISDman according to Device manual is provided.
Procedure	a) Apply first value of supply voltage VSD to the Device *;Test parameter* b) Measure current at L+ c) Evaluation 1) d) Set Device to OPERATE with MinCycleTime e) Measure current at L+ in OPERATE f) Evaluation 2) g) Repeat from b) with next VSD value
Test parameter	VSD = {18V, 30V}
Post condition	Memorize measured currents
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	For all VSD values: 1) Check value of ISDSIOmax 2) Check value of ISDIOLmax
Test passed	ISDSIOmax ≤ 200 mA (Warning if > 200 mA, but ≤ ISDman), and ISDSIOmax ≤ ISDman, and ISDIOLmax ≤ 200 mA (Warning if > 200 mA, but ≤ ISDman), and ISDIOLmax ≤ ISDman
Test not passed (examples)	ISDSIOmax > ISDman, or ISDIOLmax > ISDman
Report	ISDSIOmax (VSD = 18 V): <value/ n/a> <ok | nok | warning> ISDSIOmax (VSD = 30 V): <value/ n/a> <ok | nok | warning> ISDIOLmax (VSD = 18 V): <value> <ok | nok | warning> ISDIOLmax (VSD = 30 V): <value> <ok | nok | warning>

5.3.2 Power supply current consumption of the Device II

Table 18 defines the test conditions for this test case.

Table 18 – Power supply current consumption of the Device II

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0320
Name	TCD_PHYL_INTF_DOCISD
Purpose (short)	Power supply current consumption above recommendation
Equipment under test (EUT)	Device, supply current ISD > 200mA (see TC_0011 report: Table 17)
Test case version	1.0
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, 5.4.1, Table 6
Configuration / setup	–
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Verification in Device datasheet or user manual for increased power supply current requirement (> 200 mA) of the Device.
Precondition	PLT: not required EUT: Device datasheet or user manual (maximum current consumption ISDman)
Procedure	a) Check Device documentation for hint on constraints or user responsibility due to increased power supply current requirement b) Input: Documentation available / not available c) Evaluation 1)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Retrieve information on maximum current consumption
Test passed	Information available
Test not passed (examples)	Information insufficient or not available in documentation for user
Report	Type of information: <value> <ok | nok | warning>

5.3.3 Power-on behavior of the Device

Table 19 defines the test conditions for this test case.

Table 19 – Power-on behavior of the Device

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0012
Name	TCD_PHYL_INTF_ISIRD
Purpose (short)	Power-on current consumption (charge) and Device behavior
Equipment under test (EUT)	Device with maximum supply current ISD ≤ 200 mA
Test case version	1.2
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.3, Table 6 and 5.4.1, Table 10
Configuration / setup	The Device is powered by a Master with current supply capabilities according to ISIRM and ISM. See Figure A.4.

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The current and communication of the Device is monitored upon power-on: The Device charge requirements and behavior at power-on are verified at minimum and maximum power supply conditions.
Precondition	PLT: Voltage source with value VSD and current monitor for ISD between L+ and L- EUT: Device is powered off
Procedure	a) Switch on Device with first value of supply voltage VSD *;Test parameter* b) Trigger ISD measurement, if VSDmin (18V) is reached (tSTART) c) Stop ISD measurement at time tSTART + TRDL d) Evaluation 1) e) Set Device to OPERATE f) Count attempts in STARTUP g) Evaluation 2) h) Switch off Device (de-energize) i) Switch on Device with next value of supply voltage VSD *;Test parameter* j) Repeat from b) with next VSD value
Test parameter	VSD = {18 V, 30 V}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all VSD values: 1) Evaluate charge QISD using the formula $QISD = \int_{tSTART}^{tSTART+TRDL} ISD(t)dt$ 2) Check if communication is established and check STARTUP count
Test passed	QISD ≤ 70mAs, and OPERATE established and STARTUP count = 1
Test not passed (examples)	Charge QISD > 70 mAs, or OPERATE not established, or STARTUP count > 1
Report	QISD (VSD = 18 V): <value> <ok | nok> STARTUP count (VSD = 18 V): <value> <ok | nok> QISD (VSD = 30 V): <value> <ok | nok> STARTUP count (VSD = 30 V): <value> <ok | nok>

5.3.4 High-side residual voltage at Device C/Q

Table 20 defines the test conditions for this test case. This test case is only applicable for Devices with SIO mode or Devices where the output driver can be stimulated as if in SIO mode (e.g. via ASICs).

Table 20 – High-side residual voltage at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0013
Name	TCD_PHYL_INTF_VRESHIGH
Purpose (short)	Static high-side driver capability
Equipment under test (EUT)	Device where C/Q can be configured as output with static high level
Test case version	1.3
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.4, Table 7
Configuration / setup	The high-side output level of the Device C/Q output is measured according to Figure A.3

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Driver capability of the Device high-side driver. Measurement of the voltage drop between supply L+ and C/Q output under load condition of 50 mA.
Precondition	PLT: Voltage source with value VSD between L+ and L- EUT: Device is powered off
Procedure	a) Apply first supply voltage VSD to the Device *;Test parameter* b) Apply current sink/source with 50 mA to C/Q c) Measure voltage VCQ between L+ and C/Q d) Evaluation 1) e) Repeat from b) with next VSD value
Test parameter	VSD = {18 V, 30 V}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check VCQ
Test passed	For all VSD values: VCQ ≤ 3,0 V
Test not passed (examples)	For any VSD value: VCQ > 3,0 V
Report	VCQ (VSD = 18 V): <value> <ok | nok> VCQ (VSD = 30 V): <value> <ok | nok>

5.3.5 Low-side residual voltage at Device C/Q

Table 21 defines the test conditions for this test case. This test case is only applicable for Devices with SIO mode or Devices where the output driver can be stimulated as if in SIO mode (e.g. via ASICs).

Table 21 – Low-side residual voltage at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0014
Name	TCD_PHYL_INTF_VRESLOW
Purpose (short)	Static low-side driver capability
Equipment under test (EUT)	Device where C/Q can be configured as output with static low level
Test case version	1.3
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.4, Table 7
Configuration / setup	The output level of the Device C/Q output is measured according to Figure A.3
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Driver capability of the Device low-side driver. Measurement of the voltage drop between negative supply L- and C/Q output at sink current of 50 mA
Precondition	PLT: Voltage source with value VSD between L+ and L- EUT: Device is powered off
Procedure	a) Apply first supply voltage VSD to the Device *;Test parameter* b) Apply current sink/source with 50 mA to C/Q c) Measure voltage VCQ between C/Q and L- d) Evaluation 1) e) Repeat from b) with next VSD value
Test parameter	VSD = {18 V, 30 V}
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check VCQ
Test passed	For all VSD values: VCQ ≤ 3,0 V
Test not passed (examples)	For any VSD value: VCQ > 3,0 V
Report	VCQ (VSD = 18 V): <value> <ok | nok> VCQ (VSD = 30 V): <value> <ok | nok>

5.3.6 Pull-down or quiescent current at Device C/Q

Table 22 defines the test conditions for this test case.

Table 22 – Pull-down or quiescent current at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0015
Name	TCD_PHYL_INTF_IQQD
Purpose (short)	Static quiescent current
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.4, Table 7
Configuration / setup	Measurement of current at C/Q in receive mode

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measurement of the quiescent current into C/Q in receive mode
Precondition	PLT: Voltage source with value VSD between L+ and L- and stimulus voltage source VID between C/Q and L-. EUT: Device C/Q is set to a receive mode
Procedure	a) Apply first value of supply voltage VSD to the Device ;Test parameter b) Apply voltage VID with value of input threshold high voltage (VTHHD = 13 V) c) Measure current ICQ at C/Q d) Evaluation 1) e) Apply voltage VID with value of positive supply voltage VSD f) Measure current ICQ at C/Q g) Evaluation 2) h) Repeat from b) with next VSD value
Test parameter	VSD = {18 V, 30 V}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all VSD values: 1) Check value 1 of ICQ 2) Check value 2 of ICQ
Test passed	For all VSD values: Value 1 and value 2 ≤ 15 mA
Test not passed (examples)	For all VSD values: Value 1 or value 2 > 15 mA
Report	ICQ (VSD = 18 V, VID = 13V): <value> <ok | nok> ICQ (VSD = 18 V, VID = VSD): <value> <ok | nok> ICQ (VSD = 30 V, VID = 13V): <value> <ok | nok> ICQ (VSD = 30 V, VID = VSD): <value> <ok | nok>

5.3.7 High-level input threshold voltage at Device C/Q

Table 23 defines the test conditions for this test case.

Table 23 – High-level input threshold voltage at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0016
Name	TCD_PHYL_INTF_VTHHD
Purpose (short)	Statically input high-level threshold at C/Q
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	See Figure A.10

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measurement of the threshold voltage for high-level at the C/Q
Precondition	PLT: Voltage source with value VSD between L+ and L- and a stimulus voltage VID between C/Q and L- EUT: Device set to a *PL test set-up*, where a monitor signal of internal UART is available. This test set-up assumes C/Q port is in receive mode and change of the state at the monitor signal is observed. Otherwise, appropriate information.
Procedure	a) If *PL test set-up* is available perform steps b) to f) b) Apply first value of supply voltage VSD to the Device *;Test parameter* c) Sweep stimulus voltage VID from 5 V to 15 V in steps of maximum 0,1 V d) Capture the value of VID on the state change of monitor signal e) Evaluation 1) f) Repeat from c) with next VSD value g) If *PL test set-up* is not available perform steps h) to i) h) Provide VID manually as input from ASIC data sheet or earlier measurements i) Evaluation 1)
Test parameter	VSD = {18 V, 30 V}
Post condition	Memorize value VID (for all VSD)

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all measured values: 1) Check VID
Test passed	VID ≥ 10.5V, and VID ≤ 13.0 V
Test not passed (examples)	VID < 10.5V, or VID > 13.0 V
Report	VID @ Transition 0→1 (VSD = 18 V): <value> <ok | nok> VID @ Transition 0→1 (VSD = 30 V): <value> <ok | nok>

5.3.8 Low-level input threshold voltage at Device C/Q

Table 24 defines the test conditions for this test case.

Table 24 – Low-level input threshold voltage at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0017
Name	TCD_PHYL_INTF_VTHLD
Purpose (short)	Static input low-level threshold at C/Q
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	See Figure A.10

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measurement of the threshold voltage for low-level at the C/Q
Precondition	PLT: Voltage source with value VSD between L+ and L- and a stimulus voltage VID between C/Q and L- EUT: Device set to a *PL test set-up*, where a monitor signal of internal UART is available. This test set-up assumes C/Q port is in receive mode and change of the state at the monitor signal is observed. Otherwise, appropriate information.
Procedure	a) If *PL test set-up* is available perform steps b) to f) b) Apply first value of supply voltage VSD to the Device *;Test parameter* c) Sweep stimulus voltage VID from 15 V to 5 V in steps of maximum 0,1 V d) Capture the value of VID on the state change of monitor signal e) Evaluation 1) f) Repeat from c) with next VSD value g) If *PL test set-up* is not available perform steps h) to i) h) Provide VID manually as input from ASIC data sheet or earlier measurements i) Evaluation 1)
Test parameter	VSD = {18 V, 30 V}
Post condition	Memorize value VID (for all VSD)

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all measured values: 1) Check VID
Test passed	VID ≥ 8,0 V, AND VID ≤ 11,5 V
Test not passed (examples)	VID < 8,0V, OR VID > 11,5 V
Report	VID @ Transition 1→0 (VSD = 18 V): <value> <ok | nok> VID @ Transition 1→0 (VSD = 30 V): <value> <ok | nok>

5.3.9 Input hysteresis voltage at Device C/Q

Table 25 defines the test conditions for this test case.

Table 25 – Input hysteresis voltage at Device C/Q

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0018
Name	TCD_PHYL_INTF_VHYSD
Purpose (short)	Calculation of input hysteresis at C/Q
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5
Configuration / setup	See TC_0016 and TC_0017

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Calculation of the hysteresis voltage at C/Q based on VTHHD and VTHLD
Precondition	Value VID(TC_0016) from TC_0016 is available for all VSD values Value VID(TC_0017) from TC_0017 is available for all VSD values
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all VSD values: VHYSD = Value VID(TC_0016) – Value VID(TC_0017)
Test passed	For all VSD values: VHYSD ≥ 0 V
Test not passed (examples)	For any VSD value: VHYSD < 0 V
Report	VHYSD (VSD = 18 V): <value> <ok | nok> VHYSD (VSD = 30 V): <value> <ok | nok>

5.3.10 Permissible voltage range at Device C/Q

Table 26 defines the test conditions for this test case.

Table 26 – Permissible voltage range at Device C/Q

-CR115-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0300
Name	TCD_PHYL_INTF_VOLTRANGECQ
Purpose (short)	Permissible voltage range at Device C/Q
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.2.2, Table 5, VIL and VIH
Configuration / setup	See Figure A.10 with modifications (see Precondition)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device behavior is tested after exposures to signal voltages exceeding the supply voltage
Precondition	PLT: Voltage source with value VSD between L+ and L- and a stimulus voltage VID between C/Q and L- via a series resistance of 1 Ohm, and a Master compliant to [6] EUT: Device is attached to PLT
Procedure	a) Apply supply voltage (VSD = 29 V) to Device b) Start SDCI communication c) Stop SDCI communication (disconnect C/Q from Master port, EUT C/Q is still in receive mode) d) Apply first value of stimulus voltage VID for at least 1 min ;Test parameter e) Apply next value of stimulus voltage VID for at least 1 min ;Test parameter f) Remove stimulus voltage VID g) Reconnect C/Q to Master port h) Start SDCI communication i) Evaluation 1)
Test parameter	VID = {(VSD + 1 V), -1 V}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check communication
Test passed	Device established communication at least in 5 s and no errors.
Test not passed (examples)	Communication not established in 5 s , or Errors occurred
Report	Communication established: <yes/no> <ok | nok>

5.3.11 Electrical isolation of Device power supplies

Table 27 defines the test conditions for this test case.

Table 27 – Electrical isolation of Device power supplies

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0311
Name	TCD_PHYL_INTF_DRGI
Purpose (short)	Electrical isolation of Device with Master Port class B requirements
Equipment under test (EUT)	Device using Power 1 and Power 2
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.4.2
Configuration / setup	See Figure A.2
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test of electrical isolation between Power 1 and Power 2 of a Device with Master Port Class B requirements
Precondition	PLT: See Figure A.2 EUT: Device is operational
Procedure	a) Apply supply VSD to Device with VSDmax (VSD = 30 V) b) Apply first value of Vaux at auxiliary power supply *;Test parameter* c) Measure current IgiD0 d) Evaluation 1) e) Apply next value of Vaux at auxiliary power supply *;Test parameter* f) Measure current IgiD1 g) Evaluation 2)
Test parameter	Vaux = {0 V, 30 V}
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check RIsoD0 = VSDmax/IgiD0 2) Check RIsoD1 = VSDmax/IgiD1
Test passed	RIsoD0 > 2 MOhm, and RIsoD1 > 2 MOhm
Test not passed (examples)	Any evaluation failed
Report	RIsoD0: <value> <ok | nok> RIsoD1: <value> <ok | nok>

5.4 Wake-Up generation of the Master interface

5.4.1 Wake-Up current pulse high

Table 28 defines the test conditions for this test case.

Table 28 – Wake-Up current pulse high

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0021
Name	TCM_PHYL_INTF_IQWUH
Purpose (short)	Driver capability of the wake-up pulse – high-side driver
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 8
Configuration / setup	See Figure A.5

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The minimum requirement for the Master output peak current IQPKHM is verified with a resistive load at Master C/Q. A voltage VIM above the threshold VTHHmax during a wakeup pulse indicates that the minimum requirement is met.
Precondition	PLT: See Figure A.5 EUT: PORT_AUTOSTART (see A.4.2)
Procedure	a) Apply first supply voltage PSM to Master *;Test parameter* b) Measure voltage VIM during Wake-up pulse c) Evaluation 1) d) Repeat from b) with next PSM value
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual); Rload = {L+(PSMmin) / IQPKHMmin, L+(PSMmax) / IQPKHMmin}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check voltage VIM
Test passed	For all PSM: VIM ≥ VTHHmax
Test not passed (examples)	For any PSM: VIM < VTHHmax
Report	VIM @ WURQ (PSMmin): <value> <ok | nok> VIM @ WURQ (PSMmax): <value> <ok | nok>

5.4.2 Wake-Up pulse duration high

Table 29 defines the test conditions for this test case.

Table 29 – Wake-Up pulse duration high

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0022
Name	TCM_PHYL_INTF_TWUH
Purpose (short)	Wake-Up pulse duration (high pulse)
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.3.3 Table 8
Configuration / setup	See Figure A.7

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Verify the requirements for the wake-up pulse duration (high pulse) at Master C/Q under maximum resistive load conditions. The minimum and maximum pulse duration is measured with evaluations at the extreme values of the thresholds VTHL and VTHH.
Precondition	PLT: Resistive load (Rload) between C/Q and L-; voltage measurement between C/Q and L- EUT: PORT_AUTOSTART (see A.4.2)
Procedure	a) Apply first supply voltage PSM to Master and Rload *;Test parameter* b) Trigger time measurement t1 on voltage level at C/Q ≥ VTHHmin c) Trigger time measurement t2 on voltage level at C/Q ≤ VTHLmin d) Evaluation 1) e) Trigger time measurement t3 on voltage level at C/Q ≥ VTHHmax f) Trigger time measurement t4 on voltage level at C/Q ≤ VTHLmax g) Evaluation 2) h) Repeat from b) with next PSM value
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual); Rload = {L+(PSMmin) / IQPKHMmin, L+(PSMmax) / IQPKHMmin}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all PSM values (see Figure A.17): 1) Determine tWUmax = t2 – t1 2) Determine tWUmin = t4 – t3
Test passed	75 µs ≤ tWUmin, and tWUmax ≤ 85 µs
Test not passed (examples)	tWUmax > 85 µs, or tWUmin < 75 µs
Report	tWUmin @ WURQ: <value> <ok | nok> tWUmax @ WURQ: <value> <ok | nok>

5.4.3 Wake-Up current pulse low

Table 30defines the test conditions for this test case.

Table 30 – Wake-Up current pulse low

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0023
Name	TCM_PHYL_INTF_IQWUL
Purpose (short)	Drive capability of the wake-up pulse – low-side driver
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.3.3 Table 8
Configuration / setup	See Figure A.8

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The minimum requirement for the Master output peak current IQPKLM is verified with a resistive load at Master C/Q. A voltage at C/Q below the threshold low level of a receiver VTHLmin during a wake-up pulse indicates that the minimum requirement is met.
Precondition	PLT: Resistive load (Rload) between L+ and C/Q; voltage measurement between C/Q and L- EUT: PORT_AUTOSTART (see A.4.2)
Procedure	a) Apply first supply voltage PSM to the Master and first Rload *;Test parameter* b) Measure voltage VIM during Wake-up pulse c) Evaluation 1) d) Repeat from b) with next PSM value and next Rload value
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual); Rload = {(VL+min - 8 V)/0,5 A, (VL+max - 8 V)/0,5 A}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all PSM values: 1) Check voltage VIM
Test passed	For all PSM values: VIM ≤ VTHLmin
Test not passed (examples)	For any PSM value: VIM > VTHLmin
Report	VIM @ WURQ (PSMmin): <value> <ok | nok> VIM @ WURQ (PSMmax): <value> <ok | nok>

5.4.4 Wake-Up pulse duration low

Table 31 defines the test conditions for this test case.

Table 31 – Wake-Up pulse duration low

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0024
Name	TCM_PHYL_INTF_TWUL
Purpose (short)	Wake-Up pulse duration (low pulse)
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 8
Configuration / setup	See Figure A.8

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Verify the requirements for the wake-up pulse duration (low pulse) at Master C/Q under maximum resistive load conditions. The minimum and maximum pulse duration is measured with evaluations at the extreme values of thresholds VTHL and VTHH
Precondition	PLT: Resistive load (Rload) between L+ and C/Q; voltage measurement between C/Q and L- EUT: PORT_AUTOSTART (see A.4.2)
Procedure	a) Apply first supply voltage PSM to Master and first Rload *;Test parameter* b) Trigger time measurement t1 on voltage level at C/Q ≤ VTHLmax c) Trigger time measurement t2 on voltage level at C/Q ≥ VTHHmax d) Evaluation 1) e) Trigger time measurement t3 on voltage level at C/Q ≤ VTHLmin f) Trigger time measurement t4 on voltage level at C/Q ≥ VTHHmin g) Evaluation 2) h) Repeat from b) with next PSM value and next Rload value
Test parameter	PSM = {PSMmin, PSMmax} (according to user manual); Rload = {(VL+min - 8 V)/0,5 A, (VL+max - 8 V)/0,5 A}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all PSM values (see Figure A.18): 1) Determine tWUmax = t2 – t1 2) Determine tWUmin = t4 – t3
Test passed	75 µs ≤ tWUmin, and tWUmax ≤ 85 µs
Test not passed (examples)	tWUmax > 85 µs, or tWUmin < 75 µs
Report	tWUmin @ WURQ: <value> <ok | nok> tWUmax @ WURQ: <value> <ok | nok>

5.5 Wake-Up detection of the Device interface

5.5.1 Wake-Up pulse detection high

Table 32 defines the test conditions for this test case.

Table 32 – Wake-Up pulse detection high

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0025
Name	TCD_PHYL_INTF_TWUH
Purpose (short)	Wake-Up pulse detection capability (high pulse)
Equipment under test (EUT)	Device with SIO mode, C/Q with signal low
Test case version	1.2
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 8
Configuration / setup	Signal generator connected to EUT according Figure A.11

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	A wake-up pulse is applied to Device C/Q with timing at specified limits. The wake-up pulse is indicated via a test signal/indicator on the Device side. The Device reaction is evaluated. This test is only applicable for Devices that can drive C/Q in SIO mode to low and where means to monitor the reaction on the WURQ are provided (e.g. Interrupt line of IO-Link PHY).
Precondition	PLT: Signal Generator with high impedance or low output EUT: SIO mode. Device set to a *PL test set-up*, where a monitor signal of the wake-up detection is available, e.g. interrupt line of PHY. Otherwise appropriate information is available.
Procedure	a) If *PL test set-up* is available, perform b) to j) b) Set L+ Voltage VSD of the Device to first value ;Test parameter c) Configure EUT to drive output in SIO mode to signal low d) Apply current pulse with IQWU = IQPKHmin (current source) and of TWUmin (75 µs) to C/Q. e) Monitor test signal/WURQ indication at Device f) Evaluation 1) g) Apply current pulse with IQWU = IQPKHmin (current source) and of TWUmax (85 µs) to C/Q. h) Monitor test signal/WURQ indication at Device i) Evaluation 1) j) Repeat test from c) with next VSD voltage k) If *PL test set-up* is not available, proof of correct pulse detection is provided by manufacturer
Test parameter	VSD = {18V, 30V}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check monitored test signal for WURQ indication
Test passed	WURQ indicated, or Proof of correct pulse detection is provided by manufacturer
Test not passed (examples)	WURQ not indicated, or No proof
Report	WURQ (VSD = 18 V): <yes/no> <ok | nok> WURQ (VSD = 30 V): <yes/no> <ok | nok> Proof of correct pulse detection: <yes/no> <ok | nok>

5.5.2 Wake-Up pulse detection low

Table 33 defines the test conditions for this test case.

Table 33 – Wake-Up pulse detection low

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0026
Name	TCD_PHYL_INTF_TWUL
Purpose (short)	Wake-Up pulse detection capability (low pulse)
Equipment under test (EUT)	Device with SIO mode, C/Q with signal high
Test case version	1.2
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 8
Configuration / setup	Signal generator connected to EUT according Figure A.11
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	A wake-up pulse is applied at the Device C/Q port with timing at specified limits. The wake-up request "WURQ" is indicated via a test signal / indicator. The Device reaction is evaluated. This test is only applicable for Devices that can drive C/Q in SIO mode to high and where means to monitor the reaction on the WURQ are provided (e.g. Interrupt line of IO-Link PHY).
Precondition	PLT: Signal Generator with high impedance or high output EUT: SIO mode. Device set to a *PL test set-up*, where a monitor signal of the wake-up detection is available, e.g. interrupt line of PHY. Otherwise appropriate information is available.
Procedure	a) If *PL test set-up* is available, perform b) to j) b) Set L+ Voltage VSD to Device to first value ;*Test parameter* c) Configure EUT to drive output in SIO mode to signal high d) Apply current pulse with IQWU = IQPKHmin (current source) and of TWUmin (75 µs) to C/Q. e) Monitor test signal / WURQ indication at Device f) Evaluation 1) g) Apply current pulse with IQWU = IQPKHmin (current source) and of TWUmax (85 µs) to C/Q. h) Monitor test signal / WURQ indication at Device i) Evaluation 1) j) Repeat test from c) with next VSD voltage k) If *PL test set-up* is not available, proof of correct pulse detection is provided by manufacturer
Test parameter	VSD = {18V, 30V}
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check monitored test signal for WURQ indication
Test passed	WURQ indicated, or Proof of correct pulse detection is provided by manufacturer
Test not passed (examples)	WURQ not indicated, or No proof
Report	WURQ (VSD = 18 V): <yes/no> <ok | nok> WURQ (VSD = 30 V): <yes/no> <ok | nok> Proof of correct pulse detection: <yes/no> <ok | nok>

5.5.3 Wake-Up receive enable delay (C/Q high)

Table 34 defines the test conditions for this test case.

Table 34 – Wake-Up receive enable delay (C/Q high)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0027
Name	TCD_PHYL_INTF_TRENHIGH
Purpose (short)	Determine Receive Enable Delay after Wake-up Request
Equipment under test (EUT)	Device with SIO mode, C/Q signal high
Test case version	1.3
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 10
Configuration / setup	See Figure A.12
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	The Device shall release the high-side output driver after successful reception of a wake-up request. Measure wake-up receive enable delay of the Device with high signal at C/Q. The delay time is measured with a resistive voltage divider applied between L+ to C/Q and C/Q to L.
Precondition	PLT: Signal generator with high impedance or high output EUT: SIO Mode (output signal high)
Procedure	a) Stimulate Device for SIO mode output signal = high b) Apply wake-up request pulse (negative pulse) and return to high impedance c) Monitor C/Q signal starting from negative edge of wake-up request pulse for a duration of TRENmax d) Evaluation 1)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check tREN = time from start of wake-up request until level of C/Q reaches VTHLmin
Test passed	tREN ≤ 500 µs
Test not passed (examples)	VTHLmin not reached
Report	tREN @ C/Q = high: <value> <ok | nok>

5.5.4 Wake-Up receive enable delay (C/Q low)

Table 35 defines the test conditions for this test case.

Table 35 – Wake-Up receive enable delay (C/Q low)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0028
Name	TCD_PHYL_INTF_TRENLOW
Purpose (short)	Determine Receive Enable Delay after Wake-up Request
Equipment under test (EUT)	Device with SIO mode, C/Q signal low
Test case version	1.3
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3.3.3, Table 10
Configuration / setup	See Figure A.13
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	The Device shall release the low-side output driver after successful reception of a wake-up request. Measure wake-up receive enable delay of the Device with C/Q low. The delay time is measured with a resistive voltage divider applied between L+ to C/Q and C/Q to L.
Precondition	PLT: Signal Generator with high impedance or low output EUT: SIO mode (output signal low)
Procedure	a) Stimulate Device for SIO mode output signal = low b) Apply wake-up request pulse (positive pulse) and return to high impedance c) Monitor C/Q signal starting from positive edge of wake-up request pulse for a duration of TRENmax d) Evaluation 1)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check tREN = time from start of wake-up request until level of C/Q reaches VTHHmax
Test passed	tREN ≤ 500 µs
Test not passed (examples)	VTHHmax not reached
Report	tREN @ C/Q = low: <value> <ok | nok>

5.5.5 SDCI readiness delay

Table 36 defines the test conditions for this test case.

Table 36 – SDCI readiness delay

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0029
Name	TCD_PHYL_INTF_TRDL
Purpose (short)	SDCI Receive Enable Delay after Power-On
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 5.3, 7,3.2, Table 7, Table 9, Table 42
Configuration / setup	Reference-Master and EUT with VSD and C/Q monitoring (see Figure A.14)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device shall be able to enter communication after the SDCI readiness delay. Measure SDCI readiness delay of the Device after a power-on situation.
Precondition	PLT: Reference-Master, Port power = OFF EUT: unpowered
Procedure	a) Port power = ON, e.g. via SMI_PortPowerOffOn(ABPS_PORTPOWERON) b) Monitor VSD voltage at L+ c) Identify time t0 when VSD reaches VSDmin = 18V d) Master starts communication with wake-up sequence at t0 + TRDLmax e) Monitor C/Q signal for more than t0 + TRDLmax + TDWU f) Evaluation 1)
Test parameter	VSM = 24 V
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Device responses to Wake-up request at Device transmission rate
Test passed	Device responds to first Wake-up request and communicates
Test not passed (examples)	Device does not respond to first Wake-up request
Report	Response to first Wake-up request @ TRDL: <yes/no> <ok | nok>

5.5.6 Time to return to SIO after failed wake-up

Table 37 defines the test conditions for this test case.

Table 37 – Time to return to SIO after failed wake-up

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0301
Name	TCD_PHYL_INTF_TDELAYTOSIO
Purpose (short)	Time TDSIO within permitted range
Equipment under test (EUT)	Device with SIO mode, C/Q signal high
Test case version	1.1
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 7.3.2.2, Table 42
Configuration / setup	Master and EUT with CQ monitoring according to Figure A.9
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Delay time TDSIO from an unsuccessful Wake-Up to the return of the Device to SIO-mode is tested.
Precondition	PLT: See Figure A.9 EUT: SIO mode. Device set to a *PL test set-up*, where the output of the Device can be set to high level
Procedure	a) Apply negative wake-up request pulse to Device b) Monitor signal at C/Q c) Measure tDSIO from WURQ to transition to stable ’H’-level at C/Q d) Repeat another 5 times steps a) to c). Memorize all values of tDSIO e) Evaluation 1)
Test parameter	VSD = 24 V
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) tDSIO of all 6 measurements shall be within specified limits. Identify minimum and maximum values from all measured values of TDSIO
Test passed	For all values: 60 ms ≤ tDSIO ≤ 300 ms
Test not passed (examples)	For any value: tDSIO < 60 ms, or tDSIO > 300 ms
Report	tDSIOmin: <value> tDSIOmax: <value> tDSIO within specified boundaries: <yes/no> <ok | nok>

5.5.7 Time to Fallback after Master command

Table 38 defines the test conditions for this test case.

Table 38 – Time to Fallback after Master command

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0302
Name	TCD_PHYL_INTF_TTOFALLBACK
Purpose (short)	Time tFBD within permitted range
Equipment under test (EUT)	Device with C/Q high in SIO mode (indicated in IODD)
Test case version	1.2
Category / type	Device PL test: test to pass
Specification (clause)	[6], see 7.3.2.3, Table 42
Configuration / setup	Reference-Master and EUT (see A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	tFBD from a confirmed MasterCommand "Fallback" to the return of the Device to SIO-mode is tested.
Precondition	Reference-Master: PORT_INACTIVE EUT: C/Q high in SIO mode
Procedure	a) Set Device to OPERATE mode e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) b) Read VendorID e.g. via SMI_PortStatus c) Read MasterCycleTime from Master or EUT d) Set Device to SIO mode (apply "Fallback" MasterCommand), e.g. via SMI_PortConfiguration(ABPS_PORTINACTIVE) e) Monitor signal level at C/Q f) Measure tFBD from end of the first Device reply message to a Master Write message with MasterCommand "Fallback" and the transition to stable 'H'-level at C/Q g) Evaluation 1) h) Set Device to PREOPERATE mode e.g. via SMI_PortConfiguration (ABPS_TPYE_COMP <VendorID ≠ PortStatus.VendorID>) *;stimulate config error* i) Set Device to SIO mode (apply "Fallback" MasterCommand), e.g. via SMI_PortConfiguration(ABPS_PORTINACTIVE) j) Monitor level at C/Q k) Measure tFBD from end of the first Device reply message to a Master Write message with MasterCommand "Fallback" and the transition to stable 'H'-level at C/Q l) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Fallback delay tFBD @ OPERATE 2) Check Fallback delay tFBD @ PREOPERATE
Test passed	Three times MasterCycleTime ≤ tFBD @ OPERATE ≤ 500 ms, and Three times RecoveryTime ≤ tFBD @ PREOPERATE ≤ 500 ms
Test not passed (examples)	tFBD @ OPERATE < three times MasterCycleTime, or tFBD @ OPERATE > 500 ms, or tFBD @ PREOPERATE < three times RecoveryTime, or tFBD @ PREOPERATE > 500 ms
Report	tFBD @ OPERATE: <value> <ok | nok> tFBD @ PREOPERATE: <value> <ok | nok>

5.6 Dynamic parameters of the Master and Device interface

5.6.1 Bit eye-diagram with maximum load (Master)

Table 39 defines the test conditions for this test case.

Table 39 – Bit eye-diagram with maximum load (Master)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0030
Name	TCM_PHYL_INTF_BITEYEMAXLOAD
Purpose (short)	Eye-diagram of ‘low’ and ‘high’ bits
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 22, Table 5, Table 9
Configuration / setup	EUT and Reference-Device (see Figure A.15 and clause A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye diagram under maximum load conditions. Test waveform for bits at the C/Q input on the receiver side with a maximum permissible load applied. UART frames shall provide rising and falling edges at every bit position.
Precondition	Reference-Device: disconnected EUT: PORT_INACTIVE
Procedure	a) Attach line simulation circuit b) Attach Reference-Device with first values from COM and CQD *;Test parameter* c) Apply first value of PSM to Master *;Test parameter* d) Set Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) e) Wait until OPERATE is reached to check for communication, e.g. via SMI services f) Record waveforms on Device side for a minimum of 100 EUT UART frames g) Evaluation 1) h) Repeat from step d) with next value of PSM i) Repeat from step c) with next values from COM and CQD

Test parameter	COM = {COM2, COM3},
	CQD = {10 nF, 1 nF},
	PSM = {PSMmin, PSMmax} (according to user manual)

Post condition	Memorize waveforms

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Determine tDR, tDF, VIHDMAX, V+D, V0D, VILDMIN in waveform
Test passed	For all monitored bits of the UART frames: tDR, tDF < 0,2 TBIT, and VIHDMAX – V+D < 1,0 V, and V0D – VILDMIN < 1,0 V
Test not passed (examples)	Any of the conditions in test passed is not met
Report	Bit eye-diagram @ maximum load (PSMmin): <yes/no> <ok | nok> Bit eye-diagram @ maximum load (PSMmax): <yes/no> <ok | nok>

5.6.2 Bit eye-diagram with maximum load (Device)

Table 39 defines the test conditions for this test case.

Table 40 – Bit eye-diagram with maximum load (Device)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0294
Name	TCD_PHYL_INTF_BITEYEMAXLOAD
Purpose (short)	Eye-diagram of ‘low’ and ‘high’ bits
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 22, Table 5, Table 9,
Configuration / setup	Reference-Master connected to EUT (see Figure A.15 and clause A.1.5)
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	The waveform shall meet the requirements of the eye diagram under maximum load conditions. Test waveform for bits at C/Q input on the receiver side with a maximum permissible load applied. UART frames shall provide rising and falling edges at every bit position.
Precondition	Reference-Master: PORT_INACTIVE EUT: SIO mode
Procedure	a) Attach line simulation b) Attach Reference-Master with CQM = 1 nF c) Set VSM to first value *;Test parameter* d) Set Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) e) Wait until OPERATE is reached to check for communication, e.g. via SMI services f) Record waveforms on Master side for a minimum of 100 EUT UART frames g) Evaluation 1) h) Repeat from step c) with next value of VSM
Test parameter	VSM = {20V, 30V}
Post condition	Memorize waveforms
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Determine tDR, tDF, VIHMMAX, V+M, V0M, VILMMIN in waveform
Test passed	For all monitored bits of the UART frames: tDR, tDF < 0,2 TBIT, and VIHMMAX – V+M < 1,0 V, and V0M – VILMMIN < 1,0 V
Test not passed (examples)	Any of the conditions in test passed is not met
Report	Bit eye-diagram @ maximum load (VSM = 20 V): <yes/no> <ok | nok> Bit eye-diagram @ maximum load (VSM = 30 V): <yes/no> <ok | nok>

5.6.3 Bit eye-diagram with minimum load (Master)

Table 41 defines the test conditions for this test case.

Table 41 – Bit eye-diagram with minimum load (Master)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0031
Name	TCM_PHYL_INTF_BITEYEMINLOAD
Purpose (short)	Eye-diagram of ‘low’ and ‘high’ bits
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 22, Table 5, Table 9
Configuration / setup	EUT connected to Reference-Device without line simulation circuit (see A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under minimum line load conditions. Test waveform for bits at the C/Q input on the receiver side with an applied minimum load. UART frames shall provide rising and falling edges at every bit position.
Precondition	Reference-Device: disconnected EUT: PORT_INACTIVE
Procedure	a) Attach Reference-Device with CQD = 500 pF and set first value of COM *;Test parameter* b) Apply first value of PSM to Master *;Test parameter* c) Set Master to communication, e.g. via SMI_PortConfiguration (ABPS_NO_TYPE_CHECK) d) Wait until OPERATE is reached to check for communication, e.g. via SMI services e) Record bit-waveforms on the Device side for a minimum of 100 EUT UART frames f) Evaluation 1) g) Repeat from step c) with next value of PSM h) Repeat from step b) with next value of COM
Test parameter	COM= {COM2, COM3} PSM = {PSMmin, PSMmax} (according to user manual)
Post condition	Memorize waveforms

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Determine tDR, tDF, VIHDMAX, V+D, V0D, VILDMIN in waveform
Test passed	For all monitored bits of the UART frames: tDR, tDF < 0,2 TBIT, and VIHDMAX – V+D < 1,0 V, and V0D – VILDMIN < 1,0 V
Test not passed (examples)	Any of the conditions in test passed is not met
Report	Bit eye-diagram @ minimum load (PSMmin): <yes/no> <ok | nok> Bit eye-diagram @ minimum load (PSMmax): <yes/no> <ok | nok>

5.6.4 Bit eye-diagram with minimum load (Device)

Table 41 defines the test conditions for this test case.

Table 42 – Bit eye-diagram with minimum load (Device)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0295
Name	TCD_PHYL_INTF_BITEYEMINLOAD
Purpose (short)	Eye-diagram of ‘low’ and ‘high’ bits
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 22, Table 5, Table 9
Configuration / setup	Reference-Master connected to EUT without line simulation (see A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under minimum line load conditions. Test waveform for bits at C/Q input on the receiver side with an applied minimum line load. UART frames shall provide rising and falling edges at every bit position.
Precondition	Reference-Master: PORT_INACTIVE EUT: SIO mode
Procedure	a) Attach Reference-Master with CQM = 500pF b) Set VSM to first value *;Test parameter* c) Set Reference-Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) d) Wait until OPERATE to check for communication, e.g. via SMI services e) Record waveforms on the Master side for a minimum of 100 EUT UART frames f) Evaluation 1) g) Repeat from step c) with next value of VSM
Test parameter	VSM = {20V, 30V}
Post condition	Memorize waveforms

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Determine tDR, tDF, VIHMMAX, V+M, V0M, VILMMIN in waveform
Test passed	For all monitored bits of the UART frames: tDR, tDF < 0,2 TBIT, and VIHMMAX – V+M < 1,0 V, and V0M – VILMMIN < 1,0 V
Test not passed (examples)	Any of the conditions in test passed is not met
Report	Bit eye-diagram @ minimum load (VSM = 20 V): <yes/no> <ok | nok> Bit eye-diagram @ minimum load (VSM = 30 V): <yes/no> <ok | nok>

5.6.5 UART frame eye-diagram with maximum load (Master)

Table 43 defines the test conditions for this test case.

Table 43 – UART frame eye-diagram with maximum load (Master)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0032
Name	TCM_PHYL_INTF_UARTEYEMAXLOAD
Purpose (short)	Eye-diagram of the UART frame
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 23, Table 9
Configuration / setup	EUT and Reference-Device using line simulation (see Figure A.15 and A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under maximum line load conditions. Test waveform for UART frame at the C/Q input on the receiver side with an applied maximum permissible load.
Precondition	Memorized bit waveforms from TC_0030
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Apply mask with keep-out areas according to [6], Figure 23 to recorded UART frame waveforms of TC_0030
Test passed	UART frame waveforms do not enter keep-out areas
Test not passed (examples)	Any UART frame waveform does enter keep-out areas
Report	UART frame eye-diagram @ maximum load (PSMmin): <yes/no> <ok | nok> UART frame eye-diagram @ maximum load (PSMmax): <yes/no> <ok | nok>

5.6.6 UART frame eye-diagram with maximum load (Device)

Table 43 defines the test conditions for this test case.

Table 44 – UART frame eye-diagram with maximum load (Device)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0296
Name	TCD_PHYL_INTF_UARTEYEMAXLOAD
Purpose (short)	Eye-diagram of the UART frame
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 23, Table 9
Configuration / setup	Reference-Master connected to EUT (see Figure A.15 and A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under maximum line load conditions. Test waveform for UART frame at the C/Q input on the receiver side with an applied maximum permissible line load. UART frames shall provide rising and falling edges at every bit position.
Precondition	Memorized waveforms of TC_0294
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Apply mask with keep-out areas according to [6], Figure 23 to recorded UART frame waveforms of TC_0294
Test passed	UART frame waveform does not enter keep-out areas
Test not passed (examples)	Any UART frame waveform enters keep-out areas
Report	UART frame eye-diagram @ maximum load (VSM = 20 V): <yes/no> <ok | nok> UART frame eye-diagram @ maximum load (VSM = 30 V): <yes/no> <ok | nok>

5.6.7 UART frame eye-diagram with minimum load (Master)

Table 45 defines the test conditions for this test case.

Table 45 – UART frame eye-diagram with minimum load (Master)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0033
Name	TCM_PHYL_INTF_UARTEYEMINLOAD
Purpose (short)	Eye-diagram of UART frame
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 23, Table 9
Configuration / setup	EUT and Reference-Device connected without line simulation (see A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under minimum line load conditions. Test waveform for UART frame at the C/Q input on the receiver side with a minimum load. UART frames shall provide rising and falling edges at every bit position.
Precondition	Memorized waveforms of TC_0031
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Apply mask with keep-out areas according to [6], Figure 23 to recorded UART frame waveforms of TC_0031
Test passed	UART frame waveforms do not enter keep-out areas
Test not passed (examples)	Any UART frame waveform enters keep-out areas
Report	UART frame eye-diagram @ minimum load (PSMmin): <yes/no> <ok | nok> UART frame eye-diagram @ minimum load (PSMmax): <yes/no> <ok | nok>

5.6.8 UART frame eye-diagram with minimum load (Device)

Table 45 defines the test conditions for this test case.

Table 46 – UART frame eye-diagram with minimum load (Device)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0297
Name	TCD_PHYL_INTF_UARTEYEMINLOAD
Purpose (short)	Eye-diagram of UART frame
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device signal test: test to pass
Specification (clause)	[6], see 5.3.3.2, Figure 23, Table 9
Configuration / setup	Reference-Master connected to EUT without line simulation (see A.1.5)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The waveform shall meet the requirements of the eye-diagram under minimum line load conditions. Test waveform for UART frame at the C/Q input on the receiver side with a minimum load. UART frames shall provide rising and falling edges at every bit position.
Precondition	Memorized waveforms of TC_0295
Procedure	–
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Apply mask with keep-out areas according to [6], Figure 23 to recorded UART frame waveforms of TC_0295
Test passed	UART frame waveform is outside keep-out areas
Test not passed (examples)	Any UART frame waveform is inside keep-out areas
Report	UART frame eye-diagram @ minimum load (VSM = 20 V): <yes/no> <ok | nok> UART frame eye-diagram @ minimum load (VSM = 30 V): <yes/no> <ok | nok>

5.6.9 UART frame transmission delay of Master (Ports)

Table 47 defines the test conditions for this test case.

Table 47 – UART frame transmission delay of Master (Ports)

-CR103-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0303
Name	TCM_PHYL_INTF_UARTTRANSDELAY
Purpose (short)	UART frame transmission delay is within permitted range
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master signal test: test to pass
Specification (clause)	[6], see A.3.3, equation (A.3)
Configuration / setup	EUT with Reference-Device and C/Q line monitor (Figure A.16)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The delay time between two consecutive UART frames of a Master message is measured.
Precondition	Reference-Device: Connected EUT: PORT_INACTIVE
Procedure	a) Set Master to communication, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) b) Record waveform at C/Q of at least 7 Master messages in STARTUP, PREOPERATE, and OPERATE c) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Determine time tFRAME between the rising edges of start bits of consecutive UART frames for all recorded Master messages. Calculate UART frame transmission delay t1 for all evaluated UART frames with: t1 = (tFRAME – 11 TBIT) Determine minimum and maximum values from all calculated delays t1
Test passed	For all values t1: t1 ≤ 1 TBIT
Test not passed (examples)	Any value t1: t1 > 1 TBIT
Report	t1min: <minimum value of t1> t1max: <maximum value of t1> t1 within specified boundaries: <yes/no> <ok | nok>

5.6.10 UART frame transmission delay of Device

Table 48 defines the test conditions for this test case.

Table 48 – UART frame transmission delay of Device

-CR103-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0304
Name	TCD_PHYL_INTF_UARTTRANSDELAY
Purpose (short)	UART frame transmission delay is within permitted range
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device signal test: test to pass
Specification (clause)	[6], see A.3.4, equation (A.4)
Configuration / setup	Device connected to Reference-Master and C/Q line monitor (Figure A.16)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The delay time between two consecutive UART frames of a Device reply message is measured.
Precondition	Reference-Master: PORT_INACTIVE EUT: SIO mode
Procedure	a) Set Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) b) Monitor waveform of at least 7 Device reply messages with more than 1 UART frame in STARTUP, PREOPERATE, and OPERATE at C/Q c) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Determine time tFRAME between the rising edges of start bits of consecutive UART frames for all recorded Device reply messages. Calculate UART frame transmission delay t2 for all evaluated UART frames with: t2 = (tFRAME – 11 TBIT)
Test passed	For all values t2: t2 ≤ 3 TBIT
Test not passed (examples)	Any value t2: t2 > 3 TBIT
Report	t2min: <minimum value of t2> t2max: <maximum value of t2> t2 within specified boundaries: <yes/no> <ok | nok>

5.6.11 Response time of Device

Table 49 defines the test conditions for this test case.

Table 49 – Response time of Device

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0305
Name	TCD_PHYL_INTF_RESPONSETIME
Purpose (short)	Device response time is within permitted range
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device signal test: test to pass
Specification (clause)	[6], see A.3.5, equation (A.5)
Configuration / setup	Device connected to Reference-Master and C/Q line monitor (Figure A.16)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The delay time between Master messages to Device reply message (end of last UART frame to begin of first UART frame) is measured.
Precondition	Reference-Master: PORT_INACTIVE EUT: SIO mode
Procedure	a) Set Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) b) Monitor all M-sequences with Master Read messages in STARTUP c) Monitor all M-sequences with Master Write messages in STARTUP d) Monitor all M-sequences with Master Read messages in PREOPERATE e) Monitor all M-sequences with Master Write messages in PREOPERATE f) Monitor at least 100 M-sequences with Master Read messages in OPERATE g) Monitor at least 100 M-sequences with Master Write messages in OPERATE h) Monitor all M-sequences with 4 Device Events in OPERATE if applicable i) Evaluation 1)
Test parameter	Exemplary: Device test Events according to IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all values tA: 1) Determine time tDELAY between the rising edges of the start bit of the last UART frame of the Master message to the start bit of the first UART frame of the Device reply message for all recorded M-sequences. Calculate response time of the Device tA for all evaluated M-sequences with: tA = (tDELAY – 11 TBIT)
Test passed	All values tA: 1 TBIT ≤ tA ≤ 10 TBIT
Test not passed (examples)	Any value tA: tA < 1 TBIT OR tA > 10 TBIT
Report	tAmin: < minimum value of tA > tAmax: < maximum value of tA > tA within specified boundaries: <yes/no> <ok | nok>

5.6.12 Device response without transmission errors

Table 50 defines the test conditions for this test case.

Table 50 – Device response without transmission errors

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0319
Name	TCD_PHYL_INTF_TRANSMISSIONERRORS
Purpose (short)	Device response without transmission errors @ standard noise-free test conditions
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device signal test: test to pass
Specification (clause)	[6], Annex I, Figure I.1
Configuration / setup	Reference-Master connected to EUT and C/Q line monitor (Figure A.16)

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device response without transmission errors at standard noise-free test conditions: - Parity - Checksum - Missing Device response
Precondition	Reference-Master: PORT_INACTIVE EUT: SIO mode
Procedure	a) Set Master to SDCI mode, e.g. via SMI_PortConfiguration(ABPS_NO_TYPE_CHECK) b) Monitor all M-sequences with Master read messages in STARTUP c) Monitor all M-sequences with Master write messages in STARTUP d) Monitor all M-sequences with Master read messages in PREOPERATE e) Monitor all M-sequences with Master write messages in PREOPERATE f) Monitor at least 100 M-sequences with Master read messages in OPERATE g) Monitor at least 100 M-sequences with Master write messages in OPERATE h) Monitor all M-sequences with four Master Events in OPERATE if applicable j) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	Count transmission errors in all recorded Device responses
Test passed	No transmission error detected
Test not passed (examples)	Transmission error detected
Report	Transmission errors: <yes/no> <ok | nok>

5.7 Test report template for PL tests

Table 51 shows the template for the test report of the Physical Layer tests.

Table 51 – Template for the test report of the Physical Layer tests

TEST CASE ID	TEST results	ok/ nok	Statement/ Exception
SDCI_TC_0001	VSM (PSMmin): <value> VSM (PSMmax): <value> Steady current flow: <yes/no>		
SDCI_TC_0002	QMI (PSMmin): <value> QMI (PSMmax): <value>		
SDCI_TC_0003	ILLM (VIM = 5 V, PSMmin): <value> ILLM (VIM = 5,1 V, PSMmin): <value> ILLM (VIM = 15 V, PSMmin): <value> ILLM (VIM = measured value of VSM, PSMmin): <value> ILLM (VIM = 5 V, PSMmax): <value> ILLM (VIM = 5,1 V, PSMmax): <value> ILLM (VIM = 15 V, PSMmax): <value> ILLM (VIM = measured value of VSM, PSMmax): <value>		
SDCI_TC_0004	VRQHM (PSMmin): <value> VRQHM (PSMmax): <value>		
SDCI_TC_0005	VRQLM (PSMmin): <value> VRQLM (PSMmax): <value>		
SDCI_TC_0006	VIM @ Transition 0→1 (PSMmin): <value> VIM @ Transition 0→1 (PSMmax): <value>		
SDCI_TC_0007	VIM @ Transition 1→0 (PSMmin): <value> VIM @ Transition 1→0 (PSMmax): <value>		
SDCI_TC_0008	VHYSM (PSMmin): <value> VHYSM (PSMmax): <value>		
SDCI_TC_0299	Communication established: <yes/no>		
SDCI_TC_0310	RIsoM0: <value> RIsoM1: <value>		
SDCI_TC_0011	ISDSIOmax (VSD = 18 V): <value/ n/a> ISDSIOmax (VSD = 30 V): <value/ n/a> ISDIOLmax (VSD = 18 V): <value> ISDIOLmax (VSD = 30 V): <value>		Warning
SDCI_TC_0320	Type of information: <value>		Warning
SDCI_TC_0012	QISD (VSD = 18 V): <value> STARTUP count (VSD = 18 V): <value> QISD (VSD = 30 V): <value> STARTUP count (VSD = 30 V): <value>		
SDCI_TC_0013	VCQ (VSD = 18 V): <value> VCQ (VSD = 30 V): <value>		
SDCI_TC_0014	VCQ (VSD = 18 V): <value> VCQ (VSD = 30 V): <value>		
SDCI_TC_0015	ICQ (VSD = 18 V, VID = 13V): <value> ICQ (VSD = 18 V, VID = VSD): <value> ICQ (VSD = 30 V, VID = 13V): <value> ICQ (VSD = 30 V, VID = VSD): <value>		
SDCI_TC_0016	VID @ Transition 0→1 (VSD = 18 V): <value> VID @ Transition 0→1 (VSD = 30 V): <value>		
SDCI_TC_0017	VID @ Transition 1→0 (VSD = 18 V): <value> VID @ Transition 1→0 (VSD = 30 V): <value>		
SDCI_TC_0018	VHYSD (VSD = 18 V): <value> VHYSD (VSD = 30 V): <value>		
SDCI_TC_0300	Communication established: <yes/no>		
SDCI_TC_0311	RIsoD0: <value> RIsoD1: <value>		

TEST CASE ID	TEST results	ok/ nok	Statement/ Exception
SDCI_TC_0021	VIM @ WURQ (PSMmin): <value> VIM @ WURQ (PSMmax): <value>		
SDCI_TC_0022	tWUmin @ WURQ: <value> tWUmax @ WURQ: <value>		
SDCI_TC_0023	VIM @ WURQ (PSMmin): <value> VIM @ WURQ (PSMmax): <value>		
SDCI_TC_0024	tWUmin @ WURQ: <value> tWUmax @ WURQ: <value>		
SDCI_TC_0025	WURQ (VSD = 18 V): <yes/no> WURQ (VSD = 30 V): <yes/no> Proof of correct pulse detection: <yes/no>		
SDCI_TC_0026	WURQ (VSD = 18 V): <yes/no> WURQ (VSD = 30 V): <yes/no> Proof of correct pulse detection: <yes/no>		
SDCI_TC_0027	tREN @ C/Q = high: <value>		
SDCI_TC_0028	tREN @ C/Q = low: <value>		
SDCI_TC_0029	Response to first Wake-up request @TRDL: <yes/no>		
SDCI_TC_0301	tDSIOmin: <value> tDSIOmax: <value> tDSIO within specified boundaries: <yes/no>		
SDCI_TC_0302	tFBD @ OPERATE: <value> tFBD @ PREOPERATE: <value>		
SDCI_TC_0030	Bit eye-diagram @ maximum load (PSMmin): <yes/no> Bit eye-diagram @ maximum load (PSMmax): <yes/no>		
SDCI_TC_0294	Bit eye-diagram @ maximum load (VSM = 20 V): <yes/no> Bit eye-diagram @ maximum load (VSM = 30 V): <yes/no>		
SDCI_TC_0031	Bit eye-diagram @ minimum load (PSMmin): <yes/no> Bit eye-diagram @ minimum load (PSMmax): <yes/no>		
SDCI_TC_0295	Bit eye-diagram @ minimum load (VSM = 20 V): <yes/no> Bit eye-diagram @ minimum load (VSM = 30 V): <yes/no>		
SDCI_TC_0032	UART frame eye-diagram @ maximum load (PSMmin): <yes/no> UART frame eye-diagram @ maximum load (PSMmax): <yes/no>		
SDCI_TC_0296	UART frame eye-diagr. @ maximum load (VSM = 20 V): <yes/no> UART frame eye-diagr. @ maximum load (VSM = 30 V): <yes/no>		
SDCI_TC_0033	UART frame eye-diagram @ minimum load (PSMmin): <yes/no> UART frame eye-diagram @ minimum load (PSMmax): <yes/no>		
SDCI_TC_0297	UART frame eye-diagr. @ minimum load (VSM = 20 V): <yes/no> UART frame eye-diagr. @ minimum load (VSM = 30 V): <yes/no>		
SDCI_TC_0303	t1min: <minimum value of t1> t1max: <maximum value of t1> t1 within specified boundaries: <yes/no>		
SDCI_TC_0304	t2min: <minimum value of t2> t2max: <maximum value of t2> t2 within specified boundaries: <yes/no>:		
SDCI_TC_0305	tAmin: < minimum value of tA > tAmax: < maximum value of tA > tA within specified boundaries: <yes/no>:		
SDCI_TC_0319	Transmission errors: <yes/no>:		

737

738

6 Device protocol test cases

6.1 General

-CR019-

The protocol tests can be performed almost automatically with the help of a Device-Tester as defined in A.2.2. The test sequences are described in 4.4 together with a list of the relevant test cases for Devices without ISDU support in Table 4, and a list of the relevant test cases for Devices with ISDU support in Table 5. Supplementary requirements for Legacy-Devices beyond the definitions in [5] are listed in Annex B.

Procedure results not mentioned in the evaluation part are assumed to be successful for passing the tests.

6.2 STARTUP

6.2.1 STARTUP cycle time

Table 52 defines the test conditions for this test case.

Table 52 – STARTUP cycle time

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0034
Name	TCD_DLPC_STUP_CYCTIME
Purpose (short)	Test STARTUP cycle time
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.3.2.5, 9.3.3.2, and A.2.6
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device shall perform in state STARTUP at any cycle time ≥ 100 TBIT
Precondition	DTU: Wake-up and ComRequest are achieved until STARTUP is reached EUT: Device is in SDCI mode
Procedure	a) Master takes first value of CycTime in field "Test parameter" b) Master reads communication parameter (Direct Parameter 0x02 to 0x06) c) Master repeats at b) with next value of CycTime
Test parameter	CycTime {100 TBIT, 10 000 TBIT, 10 s}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Device response after b)
Test passed	The Device responds to all read requests with valid (constant) data
Test not passed (examples)	No or incorrect response at any of the read requests
Report	Cycle time variation in STARTUP: <ok | nok>

6.2.2 From STARTUP to OPERATE

Table 53 defines the test conditions for this test case.

Table 53 – From STARTUP to OPERATE

-CR089-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0035
Name	TCD_DLPC_STUP_STUPOPER1
Purpose (short)	Test state transition STARTUP to OPERATE
Equipment under test (EUT)	Device, except those with M-sequence TYPE_0 in OPERATE
Test case version	1.3
Category / type	Device protocol test: test to pass and test to fail
Specification (clause)	[6], see 7.3.2.5, 9.3.3.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	A Device supporting protocol revision V1.0 shall switch from STARTUP to OPERATE after receiving Master command 0x99. This transition is only used by Master supporting V1.0 according [5]. A Device not supporting protocol revision V1.0 shall not switch from STARTUP to OPERATE after receiving Master command 0x99.
Precondition	DTU: Wake-up and ComRequest are achieved until STARTUP is reached EUT: Device is in SDCI mode
Procedure	a) Master reads communication parameters (Direct Parameter 0x02 to 0x06) b) Master takes first value of MasterCycleTime in field "Test parameter" c) Master sends MasterCycleTime d) Master sends Master command 0x99 "DeviceOperate" e) Master sends ISDU "idle" request using the M-sequence TYPE for OPERATE f) Check Device response g) Master and Device switch to STARTUP h) Repeat from c) with next MasterCycleTime in field "Test parameter"
Test parameter	MasterCycleTime {MinCycleTime of Device, 0xBF (= 132,8 ms)}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) If Device supports protocol revision V1.0, check in step f) whether Device responds using M-Sequence TYPE for OPERATE 2) If Device does not support protocol revision V1.0, check in step f) whether Device is not responding
Test passed	The Device responds correctly to any request according to its protocol support
Test not passed (examples)	The Device does not respond correctly to any request according to its protocol support or timeout
Report	Transition from STARTUP directly to OPERATE: <ok | nok>

6.2.3 Master start-up with overwrite of the RID (compatible)

Table 54 defines the test conditions for this test case.

Table 54 – Master start-up with overwrite of the RID (compatible)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0306
Name	TCD_DLPC_CHCK_OVERRIDOK
Purpose (short)	Check Device start-up behavior with overwrite of the Device RID (compatible)
Equipment under test (EUT)	Device, except those not supporting protocol revision backward compatibility
Test case version	1.2
Category / type	Device protocol test: test to pass
Specification (clause)	[6], 10.6.3
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Check Device start-up behavior with overwrite of the RevisionID (compatible). This test supposes the active switching to protocol revision 1.0. The Master overwrites the RID and the Device accepts the requested protocol version.
Precondition	DTU: SIO mode; Port is configured to RevisionID 0x10 EUT: RevisionID is set to factory settings (0x11)
Procedure	a) Master performs WURQ, ComRequest, Startup up to MasterIdent b) Master detects incorrect "RevisionID" c) Master overwrites the RevisionID with the requested legacy RevisionID d) Master writes MasterCommand 0x96 e) Master reads communication parameters again f) Master switches the "modified" Device into OPERATE mode
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) After step e), check RevisionID 2) After step f), check Device response
Test passed	RevisionID (RID) = 0x10, and Device response with M-sequence TYPE for OPERATE
Test not passed (examples)	No response, or Device provides RID = 0x11, or Incorrect Device response
Report	Active switching to legacy protocol revision accepted: <ok | nok>

6.2.4 Illegal STARTUP to OPERATE

Table 55 defines the test conditions for this test case.

Table 55 – Illegal STARTUP to OPERATE

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0036
Name	TCD_DLPC_STUP_STUPOPER2
Purpose (short)	Test illegal state transition STARTUP to OPERATE
Equipment under test (EUT)	Device, except those with M-sequence TYPE_0 in OPERATE
Test case version	1.2
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.2.3.5, 9.3.3.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device shall not switch from STARTUP to OPERATE unless it receives a Master command 0x99.
Precondition	DTU: Wake-up and ComRequest are achieved until STARTUP is reached EUT: Device is in SDCI mode
Procedure	a) Master reads communication parameters (Direct Parameter 0x02 to 0x06) b) Master sends MasterCycleTime c) Master sends ISDU "idle" request using the M-sequence TYPE for OPERATE
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After step c), check Device response
Test passed	No Device response
Test not passed (examples)	Any Device response
Report	Device does not leave STARTUP: <ok | nok>

6.2.5 From OPERATE to STARTUP via Master command

Table 56 defines the test conditions for this test case.

Table 56 – From OPERATE to STARTUP via Master command

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0037
Name	TCD_DLPC_OPER_OPERSTUP1
Purpose (short)	Test correct state transition from OPERATE to STARTUP
Equipment under test (EUT)	Device, except those with M-sequence TYPE_0 in OPERATE
Test case version	1.2
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.2.3.5, 9.3.3.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct state transition from OPERATE to STARTUP via Master command
Precondition	DTU: SDCI communication, OPERATE EUT: Device is in SDCI mode
Procedure	a) Master sends MasterCommand 0x97 "DeviceStartup" b) Master sends ISDU "idle" request using the M-sequence TYPE for OPERATE c) Master reads Direct Parameter page address 0x02 using M-Sequence TYPE_0
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After step b), check Device response 2) After step c), check Device response
Test passed	No response in 1), and Correct response in 2)
Test not passed (examples)	Any response in 1), or Incorrect response in 2)
Report	Transition from OPERATE state to STARTUP state: <ok | nok>

6.2.6 From OPERATE to STARTUP via M-sequence TYPE_0

Table 57 defines the test conditions for this test case.

Table 57 – From OPERATE to STARTUP via M-sequence TYPE_0

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0038
Name	TCD_DLPC_OPER_OPERSTAR2
Purpose (short)	Test state transition OPERATE to STARTUP
Equipment under test (EUT)	Device, except those with M-sequence TYPE_0 in OPERATE
Test case version	1.3
Category / type	Device protocol test: test to pass (positive testing)
Specification (clause)	[6], see 7.2.3.5, 9.3.3.2; see NOTE in Table A.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test state transition OPERATE to STARTUP
Precondition	DTU: SDCI communication, OPERATE EUT: Device is in SDCI mode
Procedure	a) Master sends M-sequence TYPE_0 to read Direct Parameter page address 0x02 b) Master sends M-sequence TYPE_0 to read Direct Parameter page address 0x02 c) Master sends ISDU "idle" request, using the M-sequence TYPE for OPERATE
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After step a), check Device response 2) After step b), check Device response 3) After step c), check Device response
Test passed	No response in 1), and Correct response in 2), and No response in 3)
Test not passed (examples)	Any response in 1), or Incorrect response in 2), or Any response in 3)
Report	Incorrect M-sequence in OPERATE state: <ok | nok>

6.3 PREOPERATE

6.3.1 From STARTUP to PREOPERATE Read

Table 58 defines the test conditions for this test case.

Table 58 – From STARTUP to PREOPERATE Read

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0039
Name	TCD_DLPC_PROP_READDPP1
Purpose (short)	Switch Device from STARTUP to PREOPERATE and read DPP1.
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device protocol test: test to pass, test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.7
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Switch Device from STARTUP to PREOPERATE via Master command 0x9A and read DPP1. Device activates On-request Data, ISDU and Event handler and returns DL_Mode.ind (PREOPERATE). Device reply message to Master read message to be checked.
Precondition	- Initialize communication (WURQ) - Communication initialization successful (both in STARTUP state) - Save M-sequenceCapability, PDIn and PDOut for later comparison
Procedure	a) DTU sends MasterCommand PREOPERATE (0x9A) b) DTU read message: Read DPP1
Test parameter	M-sequenceCapability, PDIn, and PDOut
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether MasterCommand 0x9A results in a correct reaction of the Device 2) Determine with saved settings on PDIn, PDOut, and M-sequenceCapability the expected M-sequenceTYPE 3) Check whether the Device reply message has been received with the expected amount of On-request Data octets. 4) Check whether no process data has been transmitted.
Test passed	Device reply message has been received with the expected amount of On-request Data octets
Test not passed (examples)	In 1) MasterCommand 0x95 results in a state ≠ PREOPERATE, or In 3), 4) No or incorrect response from the Device
Report	Read of DPP1 in PREOPERATE state: <ok | nok>

6.3.2 From STARTUP to PREOPERATE Write

Table 59 defines the test conditions for this test case.

Table 59 – From STARTUP to PREOPERATE Write

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0040
Name	TCD_DLPC_PROP_WRITEDPP1
Purpose (short)	Switch Device from STARTUP to PREOPERATE and write DPP1.
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device protocol test: test to pass, test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.7
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Switch Device from STARTUP to PREOPERATE state via MasterCommand 0x9A and read DPP1. Master writes DPP1. Device reply message to Master read message to be checked. NOTE The number of octets to write depends on the used M-sequenceTYPE. The DPP1 will not be completely written, but a write request with a valid M-sequenceTYPE length will be accepted.
Precondition	- Initialize communication (WURQ) - Communication initialization between Master and Device has been successful (both in STARTUP) - Save M-sequenceCapability, PDIn, and PDOut for later comparison
Procedure	a) DTU sends MasterCommand PREOPERATE (0x9A). b) Device acknowledges command c) DTU and Device changed to PREOPERATE d) Read DPP1 (one M-sequence) e) Save the Device's response On-request Data f) DTU builds a write message with the saved On-request Data ("mirror") g) DTU writes DPP1 (one message) in correct length h) Receive Device response
Test parameter	M-sequenceCapability, PDIn and PDOut
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether MasterCommand 0x9A results in correct response of the Device 2) Determine with saved settings on PDIn, PDOut, and M-sequenceCapability the expected M-sequenceTYPE 3) Check whether the Device reply message has been received and contains no On-request Data octets
Test passed	DPP1 write command has been accepted
Test not passed (examples)	In 1) MasterCommand 0x9A results in an incorrect response of the Device In 2), 3) No or incorrect response from the Device
Report	Write of DPP1 in PREOPERATE state: <ok | nok>

6.3.3 From STARTUP to PREOPERATE short message

Table 60 defines the test conditions for this test case.

Table 60 – From STARTUP to PREOPERATE short message

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0041
Name	TCD_DLPC_PROP_SHORTMESSAGE
Purpose (short)	Test behavior to truncated M-sequence request
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device protocol test: test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.7
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Switch Device into PREOPERATE. To emulate message disturbances, caused for example by electromagnetic interference, DTU sends one octet less than required. The Device shall not respond to this truncated M-sequence request and respond to the following request without error.
Precondition	DTU and Device in PREOPERATE
Procedure	a) DTU writes to parameter "VendorID" in DPP1 (address 0x07/0x08) with one octet less than the normal request length b) DTU writes to parameter "VendorID" in DPP1 (address 0x07/0x08) after the shortest possible time (MinCycleTime, see Table A.9 and B.1.4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response in a) 2) Check response in b)
Test passed	In 1) No response on the first request, and In 2) Response on the second request
Test not passed (examples)	In 1) Response on the first request, or In 2) No response to the second request
Report	First response: <ok | nok> Second response: <ok | nok>

6.3.4 From PREOPERATE to STARTUP via simulated reset

Table 61 defines the test conditions for this test case.

Table 61 – From PREOPERATE to STARTUP via simulated reset

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0043
Name	TCD_DLPC_PROP_SIMRESET
Purpose (short)	Switch Device back to STARTUP from PREOPERATE via simulation of a reset
Equipment under test (EUT)	Device in PREOPERATE
Test case version	1.2
Category / type	Device protocol test: test to pass, test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34 (T12), and Annex A, Table A.7; see NOTE in Table A.9 on TYPE_0
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master and Device are in PREOPERATE. Master sends a TYPE_0 message (simulation of a Master reset – FHInfo_ILLEGAL_FRAMETYPE). The Device shall switch to STARTUP (deactivate On-request Data, ISDU and Event handler) and shall send a TYPE_0 response.
Precondition	- Establish communication (WURQ) - Communication between DTU and Device successful (both in STARTUP) - Save M-sequenceCapability, PDIn, and PDOut for later comparison - DTU sends MasterCommand PREOPERATE (0x9A) - DTU and Device in PREOPERATE
Procedure	a) DTU checks usage of TYPE_0 in PREOPERATE. If yes, stop test and raise exception. Otherwise: b) DTU sends TYPE_0 Read request to get MinCycleTime c) DTU sends TYPE_0 read request to get MinCycleTime again
Test parameter	M-sequenceCapability, PDIn, and PDOut
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether the Master TYPE_0 request in b) results in no response message from the Device 2) Check whether the Master TYPE_0 request in c) results in a response message of TYPE_0 with "MinCycleTime" from the Device
Test passed	Received "MinCycleTime" with TYPE_0 message after second Read attempt
Test not passed (examples)	Any response of the Device in procedure step b), or Device responds in procedure step c): - Device used incorrect M-sequence TYPE - No or incorrect response from Device
Report	Received "MinCycleTime" in TYPE_0 Device message: <ok | nok | exception>

6.3.5 From PREOPERATE to STARTUP with M-sequence fault

Table 62 defines the test conditions for this test case.

Table 62 – From PREOPERATE to STARTUP with M-sequence fault

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0044
Name	TCD_DLPC_PROP_FRAMEFAULT
Purpose (short)	Force Device into STARTUP by sending the OPERATE M-sequence type
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device protocol test: test to pass, test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34 (T12), and Annex A, Table A.7; see NOTE in Table A.9 on TYPE_0
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall not support another different M-sequence type while in PREOPERATE. The Device shall switch to STARTUP when detecting an illegal M-sequence type. Equal M-sequence types for PREOPERATE and OPERATE are legal, and M-sequence types TYPE_1_2 and TYPE_1_V are not treated as different.
Precondition	Device in PREOPERATE
Procedure	a) Determine expected M-sequence type and OD length from PDIn, PDOut, and M-sequenceCapability b) Read parameter "MinCycleTime" with the OPERATE M-sequence types c) Read again parameter "MinCycleTime" with the PREOPERATE M-sequence types d) Read again parameter "MinCycleTime" with the STARTUP M-sequence type
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response in b) with respect to used M-sequence types 2) Check Read response in c) with respect to used M-sequence types 3) Check Read response in d) with respect to used M-sequence types
Test passed	Case a: PREOPERATE = TYPE_0 and ≠ OPERATE M-sequence types: - No Device response at Read in b) - Correct data at Read in c) and d) - Implementation exception due to use of TYPE_0 Case b: PREOPERATE ≠ TYPE_0 and ≠ OPERATE M-sequence types: - No Device response at Read in b) and c) - Correct data at Read in d) Case c: PREOPERATE = TYPE_0 and = OPERATE M-sequence types: - Correct data at Read in b), c), and d) - Implementation exception due to use of TYPE_0 Case d: PREOPERATE ≠ TYPE_0 and = OPERATE M-sequence types: - No Device response at Read in d)
Test not passed (examples)	Any evaluation failed
Report	Tests in case a: <ok | nok | exception> Tests in case b: <ok | nok > Tests in case c: <ok | nok | exception> Tests in case d: <ok | nok >

6.3.6 Minimum Recovery Time in PREOPERATE Mode

-CR102-

Table 63 defines the test conditions for this test case.

Table 63 – From Minimum Recovery Time in PREOPERATE Mode

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0377
Name	TCD_DLPC_PROP_RECOVERY_TIME
Purpose (short)	Check if Device works with minimum recovery time in PREOPERATE mode
Equipment under test (EUT)	Device with ISDU support
Test case version	1.0
Category / type	Device protocol test: test to pass
Specification (clause)	[6], Table A.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The DTU shall use the minimum recovery time in PREOPERATE. The test case checks if the Device answers properly on each Master M-sequence.
Precondition	DTU: in PREOPERATE mode, DTU shall use minimum recovery time as specified in [6], Table A.8.
Procedure	a) DTU reads communication parameter (Direct Parameter 0x02 to 0x06) b) DTU writes to parameter A the value “test” c) DTU reads parameter A
Test parameter	parameter A: application specific tag (index 24)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Device response in a) to c): Device must answer to any M-Sequence. No repetitions are allowed 2) Check after c) if read value equals the value written in b) if the device resonse is positive in step b)
Test passed	All evaluation steps ok.
Test not passed (examples)	Any evaluation step failed
Report	All evaluations <ok | nok>

6.4 OPERATE

6.4.1 From PREOPERATE to OPERATE Read

Table 64 defines the test conditions for this test case.

Table 64 – From PREOPERATE to OPERATE Read

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0045
Name	TCD_DLPC_OPER_READ
Purpose (short)	Turn Master and Device into OPERATE via MasterCommand 0x99 and 0x98
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends MasterCommand 0x99 and 0x98 and both (Device and Master) switch into OPERATE.
Precondition	- Establish a communication (WURQ) - Communication between Master and Device successful (both in STARTUP) - Save M-sequenceCapability, PDIn, and PDOut for later comparison - DTU sends MasterCommand PREOPERATE (0x9A) - DTU and Device in PREOPERATE - Read DPP1 and save it in a variable for further use
Procedure	a) DTU sends MasterCommand 0x99 (OPERATE) followed by 0x98 (PD output valid) b) Read DPP1 for comparison with the appropriate M-sequence types for OPERATE
Test parameter	M-sequenceCapability, PDIn, and PDOut
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check reaction of Device upon MasterCommand 0x99 and 0x98 2) Determine M-sequence type via saved settings of M-sequenceCapability, PDIn, and PDOut 3) Check whether DPP1 values were received completely (see "Test parameters")
Test passed	Device in OPERATE and DPP1 received correctly and no PD
Test not passed (examples)	- State ≠ OPERATE - No or incorrect response from Device - Device did not use the expected M-sequence type
Report	DPP1 received in OPERATE: <ok | nok >

6.4.2 From PREOPERATE to OPERATE Write

Table 65 defines the test conditions for this test case.

Table 65 – From PREOPERATE to OPERATE Write

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0046
Name	TCD_DLPC_OPER_WRITE
Purpose (short)	Turn Device from PREOPERATE to OPERATE write
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends MasterCommand 0x99 and 0x98 and both (Device and Master) switch to OPERATE. Master writes DPP1. Device reply message to Master write message to be checked. NOTE The number of octets to write depends on the used M-sequenceTYPE. The DPP1 will not be completely written, but a write request with a valid M-sequence type length will be accepted.
Precondition	- Save M-sequenceCapability, PDIn, and PDOut for later comparison - Master and Device in PREOPERATE state
Procedure	a) DTU sends MasterCommand 0x99 (OPERATE) followed by 0x98 (PD output valid) b) Master changes to OPERATE c) Read DPP1 (one M-sequence) d) Save On-request Data of the Device's response ("mirror") e) Master prepares a write message with saved On-request Data f) Master writes DPP1 (one M-sequence) with correct length
Test parameter	M-sequenceCapability, PDIn, and PDOut
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check reaction of Device upon MasterCommand 0x99 and 0x98 2) Determine M-sequence type via saved settings of M-sequenceCapability, PDIn, and PDOut 3) Check Device response (write DPP1)
Test passed	Device in OPERATE, and DPP1 accepted
Test not passed (examples)	- State ≠ OPERATE - No or incorrect response from Device - Device did not use the expected M-sequence type
Report	DPP1 write accepted in OPERATE: <ok | nok >

6.4.3 From PREOPERATE to OPERATE negative Write

Table 66 defines the test conditions for this test case.

Table 66 – From PREOPERATE to OPERATE negative Write

-CR029-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0047
Name	TCD_DLPC_OPER_NEGWRITE
Purpose (short)	Switch Device from PREOPERATE to OPERATE and check negative write response
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device protocol test: test to pass, test to fail
Specification (clause)	[6], see 7.3.2.5, Figure 34, and Annex A.2, Table A.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Switch Device from PREOPERATE to OPERATE state via Master command 0x99 and 0x98. The Master writes DPP1 with one M-sequence and "parameter length underrun", i.e. one octet less than specified. Check whether the Device does not respond to this incomplete M-sequence. It is also a test purpose to send a correct M-sequence after the minimum cycle time of OPERATE and check the response.
Precondition	Master and Device in PREOPERATE state
Procedure	a) DTU sends MasterCommand 0x99 (OPERATE) followed by 0x98 (PD output valid) b) Master changes to OPERATE. c) Read DPP1 (one M-sequence) d) Save the On-request Data of the Device response e) Master prepares a write message with the saved On-request Data but one octet less than specified for the Master write message f) Master writes DPP1 (one M-sequence) with reduced length g) Master prepares a write message with the saved On-request Data in correct length h) Master writes DPP1 (one M-sequence) in correct length after the minimum cycle time of OPERATE
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether MasterCommand 0x99 results in a positive response of the Device 2) Check whether Device response (write DPP1 with one octet less than expected) results in no response 3) Check whether Device response (write DPP1 with correct length) results in an expected Device reply message
Test passed	All checks ok after using the incomplete M-sequence type
Test not passed (examples)	- OPERATE command results in a negative response, or - Device responds to message with incomplete M-sequence, or - Device did not use the expected M-sequence type, or - No Device response to the complete M-sequence
Report	DPP1 write only accepted with specified length in OPERATE: <ok | nok >

6.4.4 From OPERATE to STARTUP via simulated reset

Table 67 defines the test conditions for this test case.

Table 67 – From OPERATE to STARTUP via simulated reset

TEST CASE ATTRIBUTES	**IDENTIFICATION / REFERENCE**
Identification (ID)	SDCI_TC_0049
Name	TCD_DLPC_OPER_SIMRESET
Purpose (short)	Switch Device from OPERATE back to STARTUP via a simulated reset
Equipment under test (EUT)	Device without TYPE_0 in OPERATE
Test case version	1.2
Category / type	Device protocol test: test to pass
Specification (clause)	[6], see 7.3.2.5, Figure 34 (T11), and Annex A, Table A.8; see NOTE in Table A.9
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master and Device are in OPERATE. Master sends a TYPE_0 message (simulation of a Master reset). Device shall switch to STARTUP (deactivate On-request Data, ISDU and Event handler) and send a TYPE_0 response.
Precondition	- Master and Device in PREOPERATE - Master sends MasterCommand OPERATE (0x99) - Master and Device in OPERATE
Procedure	a) Master sends TYPE_0 Read request for the MinCycleTime parameter b) Master sends TYPE_0 Read request for the MinCycleTime parameter again
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	Check whether the Master TYPE_0 request results in a Device TYPE_0 response message with the MinCycleTime parameter value
Test passed	Received the MinCycleTime parameter value at second TYPE_0 message
Test not passed (examples)	Any response of the Device in procedure step a) No or incorrect response from Device in procedure step b)
Report	Received MinCycleTime value at second TYPE_0 message: <ok | nok >

6.4.5 Actuator behavior at PDOut invalid

Table 68 defines the test conditions for this test case.

Table 68 – Actuator behavior at PDOut invalid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0312
Name	TCD_DLPC_OPER_OUTINVALID
Purpose (short)	Failsafe reaction on PDOut invalid
Equipment under test (EUT)	Device supporting PDOut
Test case version	1.0
Category / type	Device protocol test: test to pass
Specification (clause)	[6], clause 10.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall detect the change of PDOut validity state and react as described in the user manual
Precondition	Device is in OPERATE and PDOut are marked as valid
Procedure	a) Write MasterCommand 0x99 (OPERATE) to Device *;Process output data invalid* b) Observe Device reaction (failsafe state)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether Device acknowledged MasterCommand in a) 2) Check Device reaction against description in user manual in b)
Test passed	Device changes state as described in user manual
Test not passed (examples)	Device reaction deviates from expected reaction
Report	Actuator state at PDOut invalid: <ok | nok >

6.4.6 Actuator behavior at communication loss

Table 69 defines the test conditions for this test case.

Table 69 – Actuator behavior at communication loss

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0313
Name	TCD_DLPC_OPER_CONNECTIONLOSS
Purpose (short)	Failsafe reaction on communication loss
Equipment under test (EUT)	Device supporting PDOut
Test case version	1.0
Category / type	Device protocol test: test to pass
Specification (clause)	[6], clause 7.3.3.5, 10.2 and 10.8.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device shall detect the loss of communication and react as described in the user manual
Precondition	Device is in communication mode and the PDout are marked as valid
Procedure	a) Communication stopped without MasterCommand "fallback", no further Wake-up b) Observe Device reaction (failsafe state)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	Check Device reaction after communication stop against description in user manual
Test passed	Device changes state as described in user manual
Test not passed (examples)	Device reaction deviates from expected reaction
Report	Actuator state at COMLOSS: <ok | nok >

6.4.7 PD Status for output Process Data only

-CR036-

Table 70 defines the test conditions for this test case.

Table 70 – PD Status for output Process Data only

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0376
Name	TCD_DLPC_OPER_PDVALIDITYINDICATION
Purpose (short)	Check PD Status flag for Device output Process Data only
Equipment under test (EUT)	Device with output Process Data only
Test case version	1.0
Category / type	Device protocol test; test to pass (positive testing)
Specification (clause)	[7], A.1.5
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	A Device with no input Process Data but with output Process Data shall always indicate “Process Data valid” in the CKS octet
Precondition	Device is in OPERATE and PDOut are marked as invalid
Procedure	a) Master reads Checksum / status (CKS) octet from Device b) Write MasterCommand 0x98 (ProcessDataOutputOperate) to Device c) Master reads Checksum / status (CKS) octet from Device
Test parameter	-
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step a) that PD Status flag indicates “Process Data valid” 2) Check after step c) that PD Status flag indicates “Process Data valid”
Test passed	All evaluation steps ok.
Test not passed (example)	Any evaluation step failed
Report	Device indicates “Process Data valid” <ok | nok>

6.5 ISDU (Indexed Service Data Unit)

6.5.1 Prearrangement measures and configuration

-CR029- -CR018- -CR027- -CR113- -CR066-

The possibility of write operations is a precondition for the ISDU test cases. The only possible writeable Index usable by all Device vendors is the "Application Specific Tag" parameter (Index = 0x0018). However, this parameter is optional.

Therefore, the vendor shall provide the necessary Index information for ISDU write/read operations (Config 1, 2, 3, and 7) within the IODD of the Device:

- Config1 (8-bit Index without ExtLength): An 8-bit readable and writeable SDCI Index of the Device that shall not provide data of type StringT and data length shall be less than 12 octets (ISDU read operation shall not use the "ExtLength").
- Config2 (16-bit Index ISDU access): If the Device supports 16-bit Indices, Config2 shall be a readable and writeable 16-bit Index. If the Device does not support any 16-bit Index, Config2 can be any 16-bit Index. The Device shall respond with correct ErrorCodes in case this Index is addressed. Via this Index, the test system will check the 16-bit capabilities (coping with the 16-bit ISDU addressing scheme) of the Device.
- Config3 (8-bit Index with ExtLength): An 8-bit readable SDCI Index of the Device providing more than 13 octets data for ISDU read operations with an "ExtLength" octet in an ISDU read response.
- Config7 (IndexToGenerateEvent): This Index is a parameter for Devices supporting Events. It can be used to stimulate up to two specific Test Events within the Device. Four values shall be supported by the Device:
  - EventA_Appear,
  - EventA_Disappear,
  - EventB_Appear, and
  - EventB_Disappear.

  A Write of this parameter causes the Device to stimulate the corresponding Event with the corresponding Event mode. The data type of the parameter is UIntegerT, bitLength = 8. There are Devices not supporting appearing and disappearing Events but supporting Notifications. For the test of these kind of Events, the values for Event "appeared" and Event "disappeared" shall be the same. The test system shall detect that the values are the same and perform a Notification test.

In addition to the information in the IODD the Test System will automatically generate a list of block parameters based on the Paramter set 1, see 6.7.1.1.

6.5.2 Availability of ISDU via M-sequence Capability

Table 71 defines the test conditions for this test case.

Table 71 – Availability of ISDU via M-sequence Capability

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0052
Name	TCD_DLPC_ISDU_AVAILFSEQCAP
Purpose (short)	Availability of ISDU via M-sequenceCapability in DPP1
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see Annex B.1.4, Table B.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads the Device's M-sequenceCapability via DPP1 and checks whether the ISDU data channel is supported.
Precondition	- Establish communication (WURQ) - Establish communication into PREOPERATE or OPERATE respectively
Procedure	a) Read M-sequenceCapability (DPP1, address 0x03) b) Check Bit "0" of the parameter M-sequenceCapability
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Received M-sequenceCapability 2) Bit "0" = 1 *;ISDU communication channel is supported*
Test passed	Bit "0" = 1
Test not passed (examples)	- No response from the Device - Bit "0" = 0 *;ISDU communication channel is not supported*
Report	Availability of the ISDU service: <ok | nok >

6.5.3 "Idle/Busy" check

Table 72 defines the test conditions for this test case.

Table 72 – "Idle/Busy" check

-CR021-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0053
Name	TCD_DLIC_ISDU_IDLEBUSYCHECK
Purpose (short)	Device response upon invalid FlowCtrl requests of the Master while in state “Idle_1”
Equipment under test (EUT)	Device
Test case version	1.3
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.2, Table 52, and Annex A.5, Table A.12, Table A.14
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device responds with "No Service" after reception of request of the Master with invalid FlowCtrl contents.
Precondition	DTU: Communication without Parameter or Event access EUT: -
Procedure	a) DTU sends Read requests with channel = ISDU and FlowCtrl from 0x11 to 0x1F
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Device response in a)
Test passed	Device response contains "No Service"
Test not passed (examples)	No response from the Device, or Device response contains ≠ "No Service", or Communication error
Report	ISDU FlowCtrl ignores invalid contents: <ok | nok >

6.5.4 Read 8-bit Index

Table 73 defines the test conditions for this test case.

Table 73 – Read 8-bit Index

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0054
Name	TCD_DLIC_ISDU_READINDEX8
Purpose (short)	Device response to an ISDU Read request with 8-bit Index
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4, Figure 52, and Annex A.2, Table A.10
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU read request for VendorName (0x10, mandatory). Device responds with expected VendorName.
Precondition	Master in PREOPERATE or OPERATE respectively
Procedure	a) Master: Sends ISDU Read request to Index 0x10 b) Receive Read response "busy" (0x01) until Device is ready c) Check and save Read response ("temp") d) Save I-Service, Length, Data and Checksum in variables and save ISDU Read response M-sequence COUNT
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether the I-Service code of the positive Read response = 0b1101 2) Compare Length variable with the actual received M-sequence COUNT 3) Check "temp" 4) Calculate checksum and compare with saved checksum
Test passed	"temp" received is complete as expected as vendorName from IODD, and I-Service, Length and Checksum are correct
Test not passed (examples)	No, negative or wrong response from the Device
Report	"VendorName" read correctly from Device: <ok | nok >

6.5.5 Read 8-bit Index with ExtLength

Table 74 defines the test conditions for this test case.

Table 74 – Read 8-bit Index with ExtLength

-CR027-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0055
Name	TCD_DLIC_ISDU_READ8EXTLENGTH
Purpose (short)	Read request with 8-bit Index and Read response with ExtLength
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4, Annex A.5.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU Read request using "Config3". Device responds according "Config3" (ISDU response with ExtLength octet possible).
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively - Variable size to be set to > 13 octets
Procedure	a) DTU: Sends ISDU Read request to Index defined in "Config3" b) Receive Read response "busy" (0x01) until Device is ready c) Save I-Service, Length, Data and Checksum in variables and save ISDU Read response M-sequence COUNT
Test parameter	"Config3" according to 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether the I-Service code of the positive Read response = 0b1101 2) Length = 0b0001 3) Compare ExtLength variable with M-sequence COUNT 4) Calculate checksum and compare with saved Checksum
Test passed	Response correct according to "Config3", and I-Service, Length, ExtLength and Checksum are correct
Test not passed (examples)	No, negative or wrong response from the Device
Report	"Config3" read correctly from Device: <ok | nok >

6.5.6 Write 8-bit Index

Table 75 defines the test conditions for this test case.

Table 75 – Write 8-bit Index

-CR018-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0056
Name	TCD_DLIC_ISDU_WRITE8
Purpose (short)	Write request with 8-bit Index is possible
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU write request according "Config1". Check whether the Write request has been successful.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Write request using "Config1" b) Receive Write response "busy" (0x01) until Device is ready c) Receive Write response d) DTU sends ISDU Read request on Index of "Config1" e) Save received data as "temp"
Test parameter	"Config1" according to 6.5.1. Manufacturer/vendor is responsible for choosing a variable for the test not changing value after Write.
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ISDU response in c) 2) Check received data in e)
Test passed	Positive Write response and "temp" = Config1 from IODD or ErrorType ≠ "Index not available"
Test not passed (examples)	Negative Write response ≠ "Index not available", or "temp" ≠ Config1 from IODD
Report	8-bit Write request: <ok | nok >

6.5.7 Read 8-bit Index reserved

Table 76 defines the test conditions for this test case.

Table 76 – Read 8-bit Index reserved

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0057
Name	TCD_DLIC_ISDU_READ8RESERVED
Purpose (short)	Device response “Index not available” upon Read of 8-bit reserved Index
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU Read request for Index = 0xFF (reserved) and receives "Index not available".
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request on Index 0xFF, Subindex "0" b) Receive Read response "busy" (0x01) until Device is ready c) Receive Read response when ready
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response in c)
Test passed	Negative Read response (I-Service: 0b1100, Length: 0b0100) with ErrorCode = 0x80, and AdditionalCode = 0x11 “Index not available”
Test not passed (examples)	Positive response from Device or incorrect ErrorCode or AdditionalCode
Report	Negative read response upon reserved 8-bit Index: <ok | nok >

6.5.8 Read 8-bit Index with unavailable Subindex

Table 77 defines the test conditions for this test case.

Table 77 – Read 8-bit Index with unavailable Subindex

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0058
Name	TCD_DLIC_ISDU_READ8NOSUBINDEX
Purpose (short)	8-bit Read response when Subindex not available
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU Read request for VendorName (Index = 0x10, mandatory, StringT) and Subindex = 0x02. This results in a negative Read response, “Subindex not available”.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request on Index 0x10, Subindex 2 b) Receive Read response "busy" (0x01) until Device is ready c) Receive Read response when ready
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response in c)
Test passed	Negative Read response (I-Service: 0b1100, Length: 0b0100) with ErrorCode = 0x80, and AdditionalCode = 0x12 “Subindex not available”
Test not passed (examples)	Positive response from Device or incorrect ErrorCode or AdditionalCode
Report	Negative read response upon unavailable Subindex: <ok | nok >

6.5.9 Read 16-bit Index

Table 78 defines the test conditions for this test case.

Table 78 – Read 16-bit Index

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0059
Name	TCD_DLIC_ISDU_READ16
Purpose (short)	Read response with 16-bit Index using "Config2"
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4, Figure 52, and Annex A.2, Table A.10
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU read request using "Config2". Device responds according "Config2". If the Device does not support 16-bit Indices, access shall be rejected with the correct ErrorType.
Precondition	- DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request to Index defined in "Config2" b) Receive Read response "busy" (0x01) until Device is ready c) Save I-Service, Length, Data and Checksum in variables and save ISDU Read response
Test parameter	"Config2" in 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response
Test passed	Positive read response or ErrorType "Index not available", and I-Service, Length and Checksum are correct
Test not passed (examples)	Any other negative response or no response
Report	"Config2" read correctly from Device: <ok | nok >

6.5.10 Write 16-bit Index

Table 79 defines the test conditions for this test case.

Table 79 – Write 16-bit Index

-CR018-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0060
Name	TCD_DLIC_ISDU_WRITE16
Purpose (short)	16-bit Write request is possible
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends 16-bit ISDU Write request using "Config2". Check whether the Write request was successful. If the Device does not support 16 bit addressing, the access shall be rejected with the correct ErrorCode.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends 16-bit ISDU Write request using "Config2" b) Receive response "busy" (0x01) until Device is ready c) Receive Write response d) DTU sends ISDU Read request using "Config2" e) Save received data in "temp"
Test parameter	"Config2" according to 6.5.1. Manufacturer/vendor is responsible for choosing a variable for the test not changing value after Write.
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ISDU response in c) 2) Check received data in e)
Test passed	Positive Write response and "temp" = Config2 from IODD or ErrorType = "Index not available"
Test not passed (examples)	Negative write response ≠ "Index not available", or "temp" ≠ Config2 from IODD
Report	"Config2" changed after Write request: <ok | nok >

6.5.11 Read 16-bit Index reserved

Table 80 defines the test conditions for this test case.

Table 80 – Read 16-bit Index reserved

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0061
Name	TCD_DLIC_ISDU_READ16RESERVED
Purpose (short)	Device response “Index not available” upon Read to 16-bit reserved Index
Equipment under test (EUT)	Device supporting 16-bit Indices
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master sends ISDU Read request to Index = 0x7F32 (reserved) and receives a negative response with “Index not available”
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request to Index 0x7F32, Subindex "0" b) Receive Read response "busy" (0x01) until Device is ready c) Receive Read response when ready
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check Read response in c)
Test passed	Negative Read response (I-Service: 0b1100, Length: 0b0100) with ErrorCode = 0x80, and AdditionalCode = 0x11 “Index not available”
Test not passed (examples)	Positive response from Device or incorrect ErrorCode or AdditionalCode
Report	Negative Read response upon reserved 16-bit Index: <ok | nok >

6.5.12 Read 16-bit Index with unavailable Subindex

Table 81 defines the test conditions for this test case.

Table 81 – Read 16-bit Index with unavailable Subindex

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0062
Name	TCD_DLIC_ISDU_READ16NOSUBINDEX
Purpose (short)	16-bit Read response when Subindex not available
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends ISDU read request for Config2 with Subindex "0" and Subindex "2" (not implemented). This results in a negative read response depending on Subindex.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request with Index in "Config2" and Subindex "0" b) Receive Read response "busy" (0x01) until Device is ready c) Receive and save Read response in "Sub0" d) DTU sends ISDU Read request with Index in Config2 and Subindex = "2" e) Receive Read response "busy" (0x01) until Device is ready f) Receive and save read response in "Sub2"
Test parameter	"Config2" in 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response combinations - Sub0 = Sub2 = ErrorType "Index not available" - Sub0 = positive response AND Sub2 = ErrorType "SubIndex not available"
Test passed	If exactly one of the evaluations is correct
Test not passed (examples)	None of the evaluations are correct
Report	Negative Read response upon unavailable Subindex: <ok | nok >

6.5.13 Write 8-bit Index with data length overrun

Table 82 defines the test conditions for this test case.

Table 82 – Write 8-bit Index with data length overrun

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0063
Name	TCD_DLIC_ISDU_WRITE8LENOVERRUN
Purpose (short)	Response of 8-bit Write request with 70 octets when 64 are permitted
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends Write request with 70 octets for "Config1" (maximum of 64 octets). Write Request with ExtLength. Check whether the Write request was denied with "Parameter length overrun".
Precondition	- Device in PREOPERATE state - DTU in PREOPERATE state
Procedure	a) DTU sends ISDU read request with Config1 b) Receive and save ISDU response in "temp1" c) Master: Send ISDU Write request to Index with Config1 containing 70 octets length and different content from "temp1" d) Receive Write response "busy" (0x01) until Device is ready e) Receive Write response f) DTU sends ISDU Read request with Config1 g) Receive and save ISDU response in "temp2"
Test parameter	"Config1" in 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ISDU response from e) 2) Check received data from g)
Test passed	ErrorType “Parameter length overrun” received "temp1" = "temp2"
Test not passed (examples)	Positive Write response or different ErrorType from Device "temp1" ≠ "temp2"
Report	Negative Read response upon length overrun: <ok | nok >

6.5.14 Write 8-bit Index with data length underrun

Table 82 defines the test conditions for this test case.

Table 83 – Write 8-bit Index with with data length underrun

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0064
Name	TCD_DLIC_ISDU_WRITE8WRONGLEN
Purpose (short)	Response of 8-bit Write request with one octet less than expected
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends Write request with one octet less than the "m" expected octets for Config1. Check whether the Write request was denied with "Parameter length underrun".
Precondition	- Device in PREOPERATE state - DTU in PREOPERATE state
Procedure	a) DTU sends ISDU read request with Config1 b) Receive and save ISDU response in "temp1" c) DTU sends ISDU Write request to Index with Config1 containing one octet less length and different content from "temp1" d) Receive Write response "busy" (0x01) until Device is ready e) Receive Write response f) DTU sends ISDU Read request with Config1 g) Receive and save ISDU response in "temp2"
Test parameter	"Config1" in 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ISDU response in e) 2) Check received data in g)
Test passed	ErrorType “Parameter length underrun” received, and "temp1" = "temp2"
Test not passed (examples)	Positive Write response or different ErrorType from Device, or "temp1" ≠ "temp2"
Report	Negative read response upon length underrun: <ok | nok >

6.5.15 Read 8-bit Index with incorrect Checksum value

Table 82 defines the test conditions for this test case.

Table 84 – Read 8-bit Index with incorrect Checksum value

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0065
Name	TCD_DLIC_ISDU_WRITE8WRONGCHECKSUM
Purpose (short)	Response of 8-bit Read request with incorrect checksum value
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, Annex A.5, Table A.12, and Table A.14
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master sends 8-bit Read request with an incorrect Checksum value to get VendorName (0x10). The correct Checksum value is 0x8B, the value in use is 0xFF. A negative Read response "No Service" is expected.
Precondition	- Device in PREOPERATE state or OPERATE respectively - DTU in PREOPERATE state or OPERATE respectively
Procedure	a) DTU sends ISDU Read request on Index 0x10 with incorrect checksum 0xFF b) Receive response "busy" (0x01) until Device is ready c) Receive Read response when ready
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check Read response in c)
Test passed	Negative Read response with I-Service = 0b0000 ("No Service")
Test not passed (examples)	Positive response from Device, or ErrorType with not permitted AdditionalCode
Report	Read response "No Service" upon incorrect Checksum value: <ok | nok >

6.5.16 Write 8-bit Index on read only Index

Table 85 defines the test conditions for this test case.

Table 85 – Write 8-bit Index on read only Index

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0066
Name	TCD_DLIC_ISDU_WRITE8ROINDEX
Purpose (short)	Response of 8-bit Write request upon read only Index
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device ISDU test: test to fail
Specification (clause)	[6], see 7.3.6, and Annex C, Table C.1; see 10.3.4 and Table 97
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master sends 8-bit Write request on read only index "VendorName". A negative Write response "Access denied" is expected.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Write request on Index 0x10 with 1 octet data length b) Receive response "busy" (0x01) until Device is ready c) Receive response when ready
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check Write response in c)
Test passed	Negative Write response with ErrorCode = 0x80 and AdditionalCode = 0x23 "Access denied"
Test not passed (examples)	Negative response ≠ "Access denied", or Positive Write response
Report	Negative Write response upon read only Index: <ok | nok >

6.5.17 Read 8-bit Index with aborted request

Table 86 defines the test conditions for this test case.

Table 86 – Read 8-bit Index with aborted request

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0067
Name	TCD_DLIC_ISDU_ABORTREADREQ
Purpose (short)	Response of aborted 8-bit Read request
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4, Figure 52 (T9; for Devices with more than two octets On-request Data within one message: T10 or T11), and Annex A.2, Table A.10
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends and aborts an 8-bit Read request for "VendorName" (0x10). Device switches to Idle mode.
Precondition	DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends first ISDU segment in first message b) DTU sends ISDU abort (R, FlowCTRL = ABORT = 0x1F, ISDU) in next message c) DTU sends idle message (R, FlowCTRL = IDLE 1 = 0x11, ISDU)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response
Test passed	Read response with I-Service = 0b0000 ("No Service")
Test not passed (examples)	Incorrect Device response to incomplete ISDU
Report	Device works properly after aborted Read request: <ok | nok >

6.5.18 Read 8-bit Index with aborted response

Table 87 defines the test conditions for this test case.

Table 87 – Read 8-bit Index with aborted response

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0068
Name	TCD_DLIC_ISDU_ABORTREADRESP
Purpose (short)	Reaction of aborted 8-bit Read response
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], see 7.3.6.4, Figure 52 (T11), Table 50, and Annex A.2, Table A.10
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master sends Read request for VendorName (0x0010) and receives from the Device the expected "VendorName". Master aborts reading the complete "VendorName".
Precondition	DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request to Index 0x0010 (16) b) Receive response "BUSY" (0x01) until Device is ready c) Receive first ISDU segment of the Read response "VendorName" d) DTU sends ISDU abort (R, FlowCTRL = ABORT = 0x1F, ISDU) e) DTU sends idle message (R, FlowCTRL = IDLE 1 = 0x11, ISDU)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response
Test passed	Read response with I-Service = 0b0000 ("No Service")
Test not passed (examples)	Incorrect Device reply message to the incomplete Read response
Report	Device works properly after the aborted Read response: <ok | nok >

6.5.19 Master retries when ISDU transfer failed

Table 88 defines the test conditions for this test case.

Table 88 – Master retries when ISDU transfer failed

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0309
Name	TCD_DLIC_ISDU_ELEMENT_RETRY
Purpose (short)	Behavior of Device if an ISDU segment is being repeated
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device ISDU test: test to pass
Specification (clause)	[6], 7.3.6, Table 52
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	If the transmission of a single M-sequence has been corrupted, the Master can send up to three retries requesting the same ISDU segment. Thus, the Device shall repeat the segment and correctly continue the transmission of the ISDU response afterwards. The correct behavior shall be tested sending an ISDU read request using “Config3”. NOTE**:** If the read result of the ISDU contained in "Config 3" is transmitted in less than three M-sequences, as many segments as possible shall be repeated during the test.
Precondition	- Device in PREOPERATE or OPERATE respectively - DTU in PREOPERATE or OPERATE respectively
Procedure	a) DTU sends ISDU Read request using "Config3" b) Receive Read response "busy" (0x01) until Device is ready c) Receive first segment of ISDU Read response *;FlowCTRL = 0x10* d) DTU repeats the first segment e) DTU continues reading the next segment *;FlowCTRL = 0x01* f) DTU repeats the second segment g) DTU continues reading until the last segment h) DTU repeats the last segment i) DTU sends idle message (R, FlowCTRL=IDLE 1=0x11, ISDU) to finalize transfer j) DTU sends ISDU Read request using "Config3" without retries
Test parameter	"Config3" in 6.5.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read response in i) 2) Check Length in i) 3) Compare element ExtLength with element COUNT in i) 4) Calculate checksum and compare with saved checksum in i) 5) Compare data in i) and j)
Test passed	Positive Read response with I-Service = 0b1101 ("Read response (+)") Length = 0b0001 *;see ExtLength* ExtLength corresponds to COUNT Correct checksum Both data are identical
Test not passed (examples)	No, negative or incorrect response from the Device
Report	"Config3" received correctly from Device: <ok | nok >

6.6 Events

6.6.1 General

-CR012- -CR093- -CR066- -CR105-

Any of the Device applications can generate predefined "status" information when SDCI operations fail, or "technology specific" information (diagnosis) as a result from technology specific diagnostic methods. This information can be communicated via SDCI Event to upper level systems of different capability. Thus, the following tests can only verify the conformity to the SDCI standard [6] in terms of Event handling mechanisms.

Event test cases shall be executed if the Device supports one or more Events. The Number of supported events is defined by the number of entries of EventCollection in IODD. Test cases use Test Events A and B stimulated by ISDU Write to the Index defined in Config7. For each of the Test Events, Config7 defines a value EventA_Appear/EventB_Appear that stimulates "Event Appear" and a value EventA_Disappear/EventB_Disappear that stimulates "Event Disappear" of the corresponding Event.

It is mandatory for a Device to implement the required test Events. Event Codes and Event types are vendor specific and shall be associated to be a DeviceStautus of 1 to 4. EventCodes for Event A and Event B shall be different. If the Device supports only Notifications, the test Events shall be Notifications and the Event mode is "Event single shot". The Test systems shall wait 5 seconds after initiating of the event until a timeout is shown. Even if there is no time specified in [6], the assumption is that an event should not be delayed more than 5 seconds.

Event Test Procedures
In general a device is allowed to have initial events at the beginning of a test process. It is not allowed to activate additional unexpected application specific events during the test run.
Event related tests focus on testing of the change of diagnosis during the test run. IO-Link Test Master reads out the activated events 1s after the precondition is reached and acknowledges all events. Later on the IO-Link Test Master only checks for changes in the diagnosis channel.

At the end of each test case, the Device-Tester-Unit shall clean-up the test Events. For stimulated Events, the corresponding EventA_Disappear or EventB_Disappear command shall be sent, and all Events shall be acknowledged. A pause of 1 s shall be added at the end of the test to ensure the Event can be stimulated again without delay within the next test case.

6.6.2 Single Event while in OPERATE state

Table 89 defines the test conditions for this test case. For Notification test see 6.5.1.

Table 89 – Single Event while in OPERATE state

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0069
Name	TCD_DLIC_EVNT_OPERSINGLEEVENT
Purpose (short)	Test of single Event processing while in OPERATE state.
Equipment under test (EUT)	Device supporting more than one Event
Test case version	1.5
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event flag and Event buffer is serviced as specified: - Event Flag is raised once an Event occurred - StatusCode Type 2 is set respectively - Event page is frozen while Event is pending - Events are cleared as specified - Event Type & Code match
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Device-Tester to read StatusCode. Save value in tester variable "SCa”. b) Write EventA_Appear to index in Config7 c) Device-Tester to read StatusCode. Save value in tester variable "SCc”. d) Write EventB_Appear to index in Config7 e) Device-Tester to read StatusCode. Save value in tester variable "SCe”. f) Device-Tester to read EventQualifier. g) Device-Tester to read EventCode. h) Device-Tester to write "0xFF" to StatusCode. i) Wait until Event flag is set (again). j) Device-Tester to read StatusCode. k) Device-Tester to read EventQualifier. l) Device-Tester to read EventCode. m) Device-Tester to write "0xFF" to StatusCode. n) Device-Tester to read StatusCode. Save value in tester variable "SCn".
Test parameter	Config7 (Event A and B) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that Event Flag is set 2) Check after step c) that content of “SCa” and “SCc” are different 3) Check after step c) that “SCc” indicates one Event 4) Check after step e) that content of “SCc” and “SCe” is equal 5) Check after step f) that Event Type equals Event Type of test Event A 6) Check after step g) that Event Code equals Event Code of test Event A 7) Check after step k) that Event Type equals Event Type of test Event B 8) Check after step l) that Event Code equals Event Code of test Event B 9) Check after step m) that Event Flag is cleared 10) Check after step n) that SCn is clear = no events
Test passed	All evaluation steps ok.
Test not passed (examples)	Any evaluation step failed
Report	Deviations in evaluations: <yes/no> <ok | nok> a) Event Flag set: <ok | nok> b) Content of "SCa" and "SCc" are different: <ok | nok>

TEST CASE RESULTS	CHECK / REACTION	
	c) Content of "SCa": d) Event Type & Code: e) Event Flag is cleared:	<ok | nok> <ok | nok> <ok | nok>

6.6.3 Single Event while in PREOPERATE state

Table 90 defines the test conditions for this test case. For Notification test see 6.5.1.

Table 90 – Single Event while in PREOPERATE

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0070
Name	TCD_DLIC_EVNT_PROPSINGLEEVENT
Purpose (short)	Test of single Event processing while in PREOPERATE state
Equipment under test (EUT)	Device, supporting more than one Event
Test case version	1.4
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event flag and Event buffer is serviced as specified: - Event flag is raised once Event has occurred - StatusCode Type 2 is set respectively - Event page is frozen while event is pending - Events are cleared as specified
Precondition	DTU: Communication EUT: PREOPERATE
Procedure	a) Device-Tester to read StatusCode. Save value in tester variable "SCa”. b) Write EventA_Appear to index in Config7 c) Device-Tester to read StatusCode. Save value in tester variable "SCc”. d) Write EventB_Appear to index in Config7 e) Device-Tester to read StatusCode. Save value in tester variable "SCe”. f) Device-Tester to read EventQualifier. g) Device-Tester to read EventCode. h) Device-Tester to write "0xFF" to StatusCode. i) Wait until Event flag is set (again). j) Device-Tester to read StatusCode. k) Device-Tester to read EventQualifier. l) Device-Tester to read EventCode. m) Device-Tester to write "0xFF" to StatusCode. n) Device-Tester to read StatusCode. Save value in tester variable "SCn".
Test parameter	Config7 (Event A and B) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that Event Flag is set 2) Check after step c) that content of “SCa” and “SCc” are different 3) Check after step c) that “SCc” indicates one Event 4) Check after step e) that content of “SCc” and “SCe” is equal 5) Check after step f) that Event Type equals Event Type of test Event A 6) Check after step g) that Event Code equals Event Code of test Event A 7) Check after step k) that Event Type equals Event Type of test Event B 8) Check after step l) that Event Code equals Event Code of test Event B 9) Check after step m) that Event Flag is cleared 10) Check after step n) that SCn is clear = no events
Test passed	All evaluation steps ok.
Test not passed (examples)	Any evaluation step failed

<table>
<tr><th>TEST CASE RESULTS</th><th colspan="2">CHECK / REACTION</th></tr>
<tr><td>Report</td><td>Deviations in evaluations: <yes/no>
a) Event Flag set:
b) Content of "SCa" and "SCc" are different:
c) Content of "SCa":
d) Event Type & Code:
e) Event Flag is cleared:</td><td><ok | nok>
<ok | nok>
<ok | nok>
<ok | nok>
<ok | nok>
<ok | nok></td></tr>
</table>

6.6.4 Event clearance in OPERATE state

Table 91 defines the test conditions for this test case.

Table 91 – Event clearance in OPERATE state

-CR105- -CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0071
Name	TCD_DLIC_EVNT_OPEREVENTCLEAR
Purpose (short)	Test of Event clearance while in OPERATE state.
Equipment under test (EUT)	Device supporting one or more Events
Test case version	1.4
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event flag and Event buffer is serviced as specified: - Event Flag is raised once an Event occurred - Events are cleared with any value written back into the StatusCode
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to index in Config7. b) Device-Tester to write StatusCode "0x00". c) Wait 50 ms d) Write EventA_Disappear to index in Config7. e) Device-Tester to write StatusCode "0xAA". f) Wait 1 s g) Write EventA_Appear to index in Config7. h) Device-Tester to read StatusCode. Save value in tester variable "SCf”. i) Device-Tester to write StatusCode with value of tester variable "SCf”.
Test parameter	Config7 (Event A) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that Event Flag is cleared 2) Check after step e) that Event Flag is cleared 3) Check after step i) that Event Flag is cleared
Test passed	All evaluation steps ok
Test not passed (examples)	Any evaluation step failed
Report	Deviations in evaluations: <yes/no> <ok | nok> a) Event Flag in procedure step b: <ok | nok> b) Event Flag in procedure step e: <ok | nok> c) Event Flag in procedure step i: <ok | nok>

6.6.5 Event handling while communication interruption

Table 92 defines the test conditions for this test case. For Events of type Notification, which usually are not acknowledged, it should be noted that the same rules apply as for Warnings and Errors: The Event shall be resent.

Table 92 – Event handling while communication interruption

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0072
Name	TCD_DLIC_EVNT_OPERCOMMINTERRUPT
Purpose (short)	Test of Event handling while communication is interrupted.
Equipment under test (EUT)	Device supporting one or more Events
Test case version	1.5
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 10.10.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event is handled as specified once communication is cancelled or interrupted.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to index in Config7 b) DTU to read StatusCode. Save value in tester variable "SCb”. c) DTU to read the indicated EventQualifier ("Event appears"). Save value in tester variable "SCc”. d) DTU to read the indicated EventCode. Save value in tester variable "SCd”. e) DTU performs reset f) Pause of 2 s g) DTU to wake-up Device to OPERATE state h) Read out and acknowledge Events until expected Event occurred (timeout = 15 s)
Test parameter	Config7 (Event A) Hint: Messages with transmission errors shall be repeated or dropped.
Post condition	EUT is free of Events once test is completed Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step g) that Event Flag is set because the error cause from step a) was not cleared before communication was lost. 2) Check after step h): that one Event received corresponds to tester variable "SCc", and "SCd".
Test passed	All evaluation steps ok
Test not passed (examples)	Any evaluation step failed, OR timeout in h)
Report	Deviations in evaluations: <yes/no> <ok | nok> a) Event Flag in procedure step g: <ok | nok> b) Read value in step h: <ok | nok>

6.6.6 Event handling while power supply interruption

Table 93 defines the test conditions for this test case. This test case can be skipped in case of Notifications.

Table 93 – Event handling while power supply interruption

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0073
Name	TCD_DLIC_EVNT_OPERPOWERINTERRUPT
Purpose (short)	Test of Event handling while power supply of communication is interrupted.
Equipment under test (EUT)	Device supporting Events of type Warning or Error
Test case version	1.5
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event modes are handled as specified when power supply of the Device is interrupted: The “Event appears” and “Event disappears” flow must be correct and start with “Event appears” after communication is restarted. Events that are no longer active after communication restart shall not be reported with “Event disappears”.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to Index in Config7 b) Device-Tester to read StatusCode c) Device-Tester to read the indicated EventQualifier d) Device-Tester to read the EventCode. Save value in tester variable "SCd” e) Power-down of the Device (disconnect from Device-Tester) f) Pause of 15 sec g) Re-connect Device h) Device-Tester to wake-up Device to OPERATE state i) Read out and acknowledge Events (timeout = 15 s)
Test parameter	Config7 (Event A) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step d) that one entry of SCd equals EventCode of Event A 2) Check in step i) that Event A is not read with mode “Event disappear”
Test passed	All evaluation steps ok, or timeout
Test not passed (examples)	Any evaluation step failed
Report	Event appears after power off/on cycle: <yes/no> Deviations in evaluations: <yes/no> <ok | nok> a) Read value in step i: <ok | nok>

6.6.7 Event appears/disappears

Table 94 defines the test conditions for this test case.

Table 94 – Event appears/disappears

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0074
Name	TCD_DLIC_EVNT_OPERAPPEARDISAPPEAR
Purpose (short)	Test of Event handling with Errors appearing and disappearing.
Equipment under test (EUT)	Device, supporting one or more Events, test Event is of type Error or Warning
Test case version	1.4
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event modes are handled as specified in Figure 11
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to Index in Config7 b) Device-Tester to read StatusCode c) Device-Tester to read EventQualifier. Save value in tester variable "SCc”. d) Device-Tester to read EventCode. Save value in tester variable "SCd”. e) Device-Tester to write StatusCode "0xFF". f) Wait 50 ms g) Write EventA_Disappear to Index in Config7 h) Device-Tester to read StatusCode. i) Device-Tester to read EventQualifier. Save value in tester variable "SCh”. j) Device-Tester to read EventCode. Save value in tester variable "SCi”. k) Device-Tester to write StatusCode "0xFF".
Test parameter	Config7 (Event A) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step a) that Event Flag is set 2) Check after step c) that value read shows mode = "Event appeared" 3) Check after step e) that Event Flag is cleared 4) Check after step g) that Event Flag is set 5) Check after step i) that value read shows mode = "Event disappeared" 6) Check after step j) that value of "SCi" equals value of "SCd" (EventCodes). 7) Check after step k) that Event Flag is cleared
Test passed	All evaluation steps ok
Test not passed (examples)	Any evaluation step failed
Report	Deviations in evaluations: <yes/no> <ok | nok> a) Event Flag in procedure step a: <ok | nok> b) Value in procedure step c: <ok | nok> c) Event Flag in procedure step e: <ok | nok> d) Event Flag in procedure step g: <ok | nok> e) Value in procedure step i: <ok | nok> f) EventCodes in procedure step j: <ok | nok> g) Event Flag in procedure step k: <ok | nok>

Figure 11 shows the relationship of an Error and the Event Flag and its appearance and disappearance.

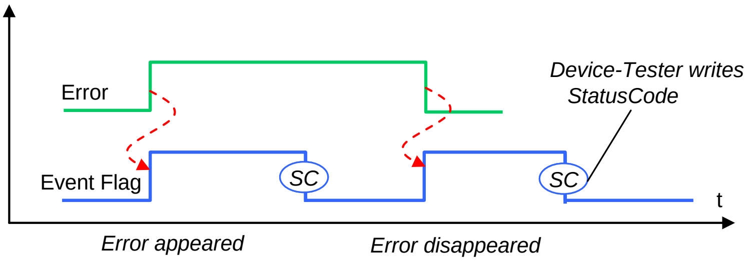


Figure 11 – Relationship of an Error and the Event Flag

6.6.8 Multi Event handling

Table 95 defines the test conditions for this test case.

Table 95 – Multi Event handling

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0075
Name	TCD_DLIC_EVNT_OPERMULTEVENT
Purpose (short)	Test of Event handling with multiple Events.
Equipment under test (EUT)	Device supporting more than one Event
Test case version	1.5
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check if Event modes are handled as specified in Figure 12
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to index in Config7 b) Device-Tester to read StatusCode c) Device-Tester to read EventQualifier. Save value in tester variable "SCc". d) Device-Tester to read EventCode. Save value in tester variable "SCd". e) Device-Tester to write Status Code "0xFF". f) Write EventB_Appear to index in Config7 g) Device-Tester to read StatusCode h) Wait 50 ms i) Write EventA_Disappear to index in Config7 j) Device-Tester to read EventQualifier. Save value in tester variable "SCj". k) Device-Tester to read EventCode. Save value in tester variable "SCk". l) Device-Tester to write Status Code "0xFF". m) Device-Tester to read StatusCode n) Write EventB_Disappear to index in Config7 o) Device-Tester to read EventQualifier. Save value in tester variable "SCo". p) Device-Tester to read EventCode. Save value in tester variable "SCp". q) Device-Tester to write StatusCode "0xFF". r) Device-Tester to read StatusCode s) Device-Tester to read EventQualifier. Save value in tester variable "SCs". t) Device-Tester to read EventCode. Save value in tester variable "SCt". u) Device-Tester to write StatusCode "0xFF".
Test parameter	Config7 (Event A and B) in 6.5.1
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step c) that value read shows mode = "Event appeared" or "Event single shot" in case of Notification 2) Check after step d) that SCd equals EventCode of Event A 3) Check after step d) that Event Flag is set 4) Check after step e) that Event Flag is cleared 5) Check after step f) that Event Flag is set 6) Check after step j) that value read shows mode = "Event appeared" or "Event single shot" in case of Notification 7) Check after step k) that SCk equals EventCode of Event B 8) Check after step k) that Event Flag is set 9) Check after step o) that value read shows mode = "Event disappeared" or "Event single shot" in case of Notification 10) Check after step p) that SCp equals EventCode of Event A 11) Check after step p) that Event Flag is set 12) Check after step s) that value read shows mode = "Event disappeared" or

	"Event single shot" in case of Notification 13) Check after step t) that SCt equals EventCode of Event B 14) Check after step t) that Event Flag is set 15) Check after step u) that Event Flag is cleared
Test passed	All evaluation steps ok
Test not passed (examples)	Any evaluation step failed
Report	Deviations in evaluations: <yes/no> <ok | nok>

Figure 12 shows the correlation of two Errors and the Event Flag and its appearance and disappearance. In case of Notifications the rising and falling edges define the time the Event is stimulated.

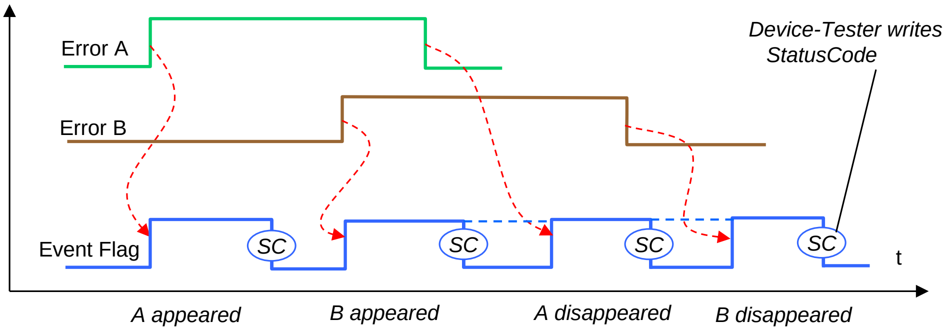


Figure 12 – Correlation of two Errors and the Event Flag

6.6.9 Short time Events

Table 96 defines the test conditions for this test case.

Table 96 – Short time Events

-CR066- -CR029-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0076
Name	TCD_DLIC_EVNT_OPERSHORTEVENT
Purpose (short)	Test of the Event handling of short time errors.
Equipment under test (EUT)	Device, supporting one or more Events, test Event is of type Error or Warning
Test case version	1.3
Category / type	Device Event test: test to pass
Specification (clause)	[6], see 7.3.8.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Event handling of short time Errors according Figure 13. "Short time Error" means that the cause of the Error is no more existent, when the acknowledgement from the Master for the appearance occurred (Write access to StatusCode (SC)). The Device shall send "Event disappeared" in this case after the acknowledgement.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write EventA_Appear to index in Config7. b) Write EventA_Disappear to index in Config7 c) Device-Tester to write StatusCode "0xFF". d) Wait for 50 ms or two times MasterCycleTime whichever is longer e) Device-Tester to write StatusCode "0xFF"
Test parameter	Config7 (Event A) in 6.5.1 and MasterCycleTime
Post condition	Test Events A and B are reset once the test is completed

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step a) that Event Flag is set 2) Check after step d) that Event Flag is set 3) Check after step e) that Event Flag is cleared
Test passed	All evaluation steps ok
Test not passed (examples)	Any evaluation step failed
Report	Deviations in evaluations: <yes/no> <ok | nok> Event has been latched: <ok | nok>

Figure 13 shows the timings of a short time error and the Event flag.

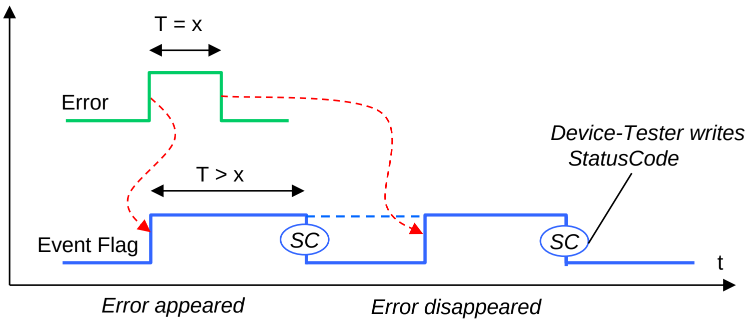


Figure 13 – Timings of a short time Error and the Event flag

6.6.10 Interconnection active Event/Device Status

Table 97 defines the test conditions for this test case.

Table 97 – Interconnection active Event/Device Status/Detailed Device Status

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0373
Name	TCD_DLIC_DEFP_EVENTDEVSTAT
Purpose (short)	Check correct interconnection between Event and (detailed) Device Status
Equipment under test (EUT)	Device supporting Events of type Warning or Error And Parameters Device Status and Detailed Device Status
Test case version	1.0
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.20 and B.2.21 and Table D.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test correct behavior regarding the interconnection of an active Event and the resulting content of parameter Device Status and Detailed Device Status.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read Parameter Device Status and store value in “devStatBase” b) Read Parameter Detailed Device Status, parse result from beginning to end in steps of 3 octets and store values in “detDevStatBase” c) Write EventA_Appear to index in Config7 d) Read Parameter Device Status e) Read Parameter Detailed Device Status, parse result from beginning to end in steps of 3 octets f) Write EventA_Disappear to index in Config7 g) Read Parameter Device Status h) Read Parameter Detailed Device Status, parse result from beginning to end in steps of 3 octets
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check values at step d) 2) Check values at step e) 3) Check values at step g) 4) Check values at step h)
Test passed	1) Device Status = 1 (maintenance required) to 4 (failure) 2) Check Detailed Device Status: One entry must be the Event Code given in Config 7 3) Device Status must be the same as stored in a) (“devStatBase”) 4) Detailed Device Status Entry given in config 7 has been deleted, content matches stored values in b) (“detDevStatBase”)
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

6.7 Data Storage (DS)

6.7.1 General

6.7.1.1 Checks on Data Storage Index

-CR013--CR038- -CR113-

Checks on different states/values shall be performed according to the notes in the "Evaluation" and "Report" field of the test cases:

- State_Property
  - value of "State of Data Storage"
  - value of "DS_UPLOAD_FLAG"
- Data_Storage_Size
  - shall be larger or equal to the memory size of the objects in the Index List, as described in "Structure of the stored DS data objects". Note: Data_Storage_Size can be larger in case Strings are transferred in condensed format.
  - check after Upload
- Parameter_checksum
  - This value shall be changed after modification of parameters listed for data storage
  - Check after parameter modification

These states are specified in [6], 10.4.2 (Data Storage state machine), and B.2.3 (Data Storage Index).

"Parameter set 1" and "Parameter set 2" are used as placeholders for two parameter sets fulfilling the following conditions:

- "Parameter set 1" and "Parameter set 2" contain parameters listed for data storage
- "Parameter set 1" and "Parameter set 2" are different in parameter values listed for data storage
- "Parameter set 1" and the parameter set of the delivered Device are different in values

6.7.1.2 Generation of "DS_UPLOAD_REQ"

It would be possible to test the generation of "DS_UPLOAD_REQ" in separate test cases. But these tests are already performed within the test cases for Upload and Download.

6.7.1.3 Different Upload test cases

Upload is tested in different states of the Device.

6.7.1.4 Different Download test cases

Download is only checked with "DS_UPLOAD_REQ" flag = "0" (false).

6.7.2 Upload without DS_UPLOAD_FLAG notification

Table 98 defines the test conditions for this test case.

Table 98 – Upload without DS_UPLOAD_FLAG notification

-CR002- -CR013- -CR023- -CR038- -CR039-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0077
Name	TCD_APPS_DSUP_NOFLAG
Purpose (short)	Explicit upload without DS_UPLOAD_FLAG notification
Equipment under test (EUT)	Device with Datastorage support
Test case version	1.1
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Table B.11, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers upload of Data Storage contents (parameter set) without DS_UPLOAD_FLAG notification
Precondition	- Device in PREOPERATE or OPERATE mode - DS_UPLOAD_FLAG is not set - Device parameterized (manufacturer to define parameter set)
Procedure	Perform upload completely as defined in DTU DS state machine: a) Switch DTU DS from deactivated to activated state. b) Stimulate upload using DS_Commands "DS_UploadStart" and "DS_UploadEnd"
Test parameter	Parameter set (manufacturer to define parameter set)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	Check whether parameter set is read without errors through Data Storage Index. Upon each of the following actions: 1) After call of the DS_UploadStart command 2) After reading/uploading the parameters 3) After call of the DS_UploadEnd command check the following: 4) State of Data Storage is correct 5) DS_UPLOAD_FLAG is not set 6) Parameter_checksum does not change 7) Verify that the value of DS Size is larger or equal with to the size of the uploaded data + 4 * number of entries in Index List (see IO-Link System Spec. Annex G)
st passed	All three checks during the three actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation action 1) and check 4): <state> <ok | nok> Result of evaluation action 1) and check 5): <flag> <ok | nok> Result of evaluation action 1) and check 6: <checksum> <ok | nok> Result of evaluation action 2) and check 4): <state> <ok | nok> Result of evaluation action 2) and check 5): <flag> <ok | nok> Result of evaluation action 2) and check 6): <checksum> <ok | nok> Result of evaluation action 3) and check 4): <state> <ok | nok> Result of evaluation action 3) and check 5): <flag> <ok | nok> Result of evaluation action 3) and check 6): <checksum> <ok | nok>

6.7.3 Upload via ParamDownloadStore

Table 99 defines the test conditions for this test case.

Table 99 – Upload via ParamDownloadStore

-CR023- -CR024-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0078
Name	TCD_APPS_DSUP_VIADOWNLOADSTORE
Purpose (short)	Explicit upload via SystemCommand "ParamDownloadStore"
Equipment under test (EUT)	Device with Datastorage support
Test case version	1.1
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.9, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers parameterization (parameter set 2) of a Device and upload of this set into the Master DS per SystemCommand "ParamDownloadStore". Manufacturer is responsible for the definition of two possible "parameter sets".
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Call SystemCommand "ParamDownloadStart" if Device supports Block Parameterization b) Write different parameter set 2 into the Device c) Call SystemCommand "ParamDownloadStore" (causes Event DS_UPLOAD_REQ) d) Wait for event DS_UPLOAD_REQ e) Perform Upload (Data Storage) completely as defined in the Master state machine (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 and parameter set 2 (defined by manufacturer)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	Check whether parameter set 2 is read without errors through Parameter_checksum. Upon each of the following actions: 1) After call of the DS_UploadStart command 2) After reading/uploading the parameters 3) After call of the DS_UploadEnd command check the following: 4) State of Data Storage is correct (active after DS_UploadStart, inactive after DS_UploadEnd) 5) DS_UPLOAD_FLAG is correct (set after "ParamDownloadStore" and cleared after DS_UploadEnd) 6) Parameter_checksum has changed to that of parameter set 2
Test passed	All three checks during the three actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation action 1) and check 4): <state> <ok | nok> Result of evaluation action 1) and check 5): <flag> <ok | nok> Result of evaluation action 1) and check 6): <checksum> <ok | nok> Result of evaluation action 2) and check 4): <state> <ok | nok> Result of evaluation action 2) and check 5): <flag> <ok | nok> Result of evaluation action 2) and check 6): <checksum> <ok | nok> Result of evaluation action 3) and check 4): <state> <ok | nok> Result of evaluation action 3) and check 5): <flag> <ok | nok> Result of evaluation action 3) and check 6): <checksum> <ok | nok>

6.7.4 Upload via ParamDownloadStore without write calls

Table 100 defines the test conditions for this test case.

Table 100 – Upload via ParamDownloadStore without write calls

-CR023- -CR025-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0079
Name	TCD_APPS_DSUP_VIADOWNLOADSTORENOWRITE
Purpose (short)	Explicit upload via "ParamDownloadStore" without write calls
Equipment under test (EUT)	Device with Datastorage support
Test case version	1.2
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers parameterization (parameter set 2) of a Device and upload of this set into the Master DS per SystemCommand "ParamDownloadStore". Manufacturer is responsible for the definition of two possible "parameter sets".
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Write different parameter set 2 into the Device b) Wait 1 s for event DS_UPLOAD_REQ c) Call SystemCommand "ParamDownloadStart" via ISDU (w/o writing parameters), if Device supports Block Parameterization d) Initiate upload via SystemCommand "ParamDownloadStore" e) Wait for Event DS_UPLOAD_REQ f) Perform upload (Data Storage) completely as defined in the Master state machine (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 and 2 (manufacturer to define possible parameter sets)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether parameter set 2 is written without errors 2) Check whether Event DS_UPLOAD_REQ was not raised after changing parameters (parameter set 2) 3) Check whether Event DS_UPLOAD_REQ was raised by the Device after SystemCommand "ParamDownloadStore" 4) Check whether parameter set 2 is read without errors through Parameter_checksum Upon each of the following actions: 5) After call of the DS_UploadStart command 6) After reading/uploading the parameters 7) After call of the DS_UploadEnd command check via Data Storage Index the following: 8) State of Data Storage is correct 9 DS_UPLOAD_FLAG is correct(set after “ParamDownloadStore” and cleared after DS_UploadEnd) 10) Parameter_checksum has changed only after "Write parameter set 2"
Test passed	All checks in 1) to 4) and all three checks during the three actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation 1): <checksum> <ok | nok> Result of evaluation 2): <flag> <ok | nok> Result of evaluation 3): <flag> <ok | nok> Result of evaluation 4): <checksum> <ok | nok> Result of evaluation action 5) and check 8): <state> <ok | nok>

TEST CASE RESULTS	CHECK / REACTION
	Result of evaluation action 5) and check 9): <flag> <ok | nok> Result of evaluation action 5) and check 10): <checksum> <ok | nok> Result of evaluation action 6) and check 8): <state> <ok | nok> Result of evaluation action 6) and check 9): <flag> <ok | nok> Result of evaluation action 6) and check 10): <checksum> <ok | nok> Result of evaluation action 7) and check 8): <state> <ok | nok> Result of evaluation action 7) and check 9): <flag> <ok | nok> Result of evaluation action 7) and check 10): <checksum> <ok | nok>

6.7.5 Upload via local parameter modification

Table 101 defines the test conditions for this test case.

Table 101 – Upload via local parameter modification

-CR023- -CR025-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0080
Name	TCD_APPS_DSUP_VIALOCALCHANGE
Purpose (short)	Implicit upload after local parameter modification
Equipment under test (EUT)	Device with local parameterization such as teach-in or panel as indicated in IODD and with Datastorage support
Test case version	1.1
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers local parameter changes of a Device and implicit upload of the parameter set into the Master DS. Manufacturer is responsible for the definition of two possible "parameter sets".
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Change parameter value(s) locally in the Device, e.g. via local menu or teach-in b) Wait for Event DS_UPLOAD_REQ c) Perform upload (Data Storage) completely as defined in the Master state machine (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 (manufacturer to define the possible parameter set)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether Event DS_UPLOAD_REQ was raised automatically by the Device after local parameter modification 2) Check whether parameter set 2 is read without errors through Parameter_checksum Upon each of the following actions: 3) After call of the DS_UploadStart command 4) After reading/uploading the parameters 5) After call of the DS_UploadEnd command check via Data Storage Index the following: 6) State of Data Storage is correct 7) DS_UPLOAD_FLAG is correct (set after local change of parameter values and cleared after DS_UploadEnd) 8) Parameter_checksum has changed only after 'local change of parameter values'
Test passed	All checks in 1), 2), and all three checks during the three actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation 1): <flag> <ok | nok> Result of evaluation 2): <checksum> <ok | nok> Result of evaluation action 3) and check 6): <state> <ok | nok> Result of evaluation action 3) and check 7): <flag> <ok | nok> Result of evaluation action 3) and check 8): <checksum> <ok | nok> Result of evaluation action 4) and check 6): <state> <ok | nok> Result of evaluation action 4) and check 7): <flag> <ok | nok> Result of evaluation action 4) and check 8): <checksum> <ok | nok> Result of evaluation action 5) and check 6): <state> <ok | nok> Result of evaluation action 5) and check 7): <flag> <ok | nok> Result of evaluation action 5) and check 8): <checksum> <ok | nok>

6.7.6 Call ParamBreak in different states of Upload

Table 102 defines the test conditions for this test case.

Table 102 – Call ParamBreak in different states of Upload

-CR023-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0081
Name	TCD_APPS_DSUP_PARABREAKABORT
Purpose (short)	Upload abort via SystemCommand "ParamBreak" in different states
Equipment under test (EUT)	Device with Datastorage and Block Parameterization support
Test case version	1.3
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers Upload aborts via SystemCommand "ParamBreak" in different states. Manufacturer is responsible for the definition of two possible "parameter sets".
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Call SystemCommand "ParamDownloadStart" b) Write different parameter set 2 into the Device c) Call SystemCommand "ParamDownloadStore" (causes Event DS_UPLOAD_REQ) d) Wait for Event DS_UPLOAD_REQ e) Start Upload via SystemCommand "ParamUploadStart" f) Call SystemCommand "ParamBreak" directly after "ParamUploadStart" g) Start Upload via SystemCommand "ParamUploadStart" h) Transmit first Block Parameter object of Config4 i) Call SystemCommand "ParamBreak" j) Start Upload via SystemCommand "ParamUploadStart" k) Transmit all Block Parameter objects of Config4 l) Call SystemCommand "ParamBreak" m) Perform upload (Data Storage) completely as defined in the Master state machine (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 and 2
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether Event DS_UPLOAD_REQ was raised automatically by the Device 2) Check whether parameter set 2 is read without errors through Parameter_checksum Upon each of the following actions: 3) After call of each SystemCommand "ParamUploadStart" (3x) 4) After call of each SystemCommand "ParamBreak" (3x) check via Data Storage Index the following: 5) State of Data Storage is 0b00 (inactive) 6) DS_UPLOAD_FLAG is set 7) Parameter_checksum has changed only after 'write parameter set 2' 8) Check whether the Upload has been completed without errors
Test passed	All checks in 1), 2), 8), and all three checks during the two actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation 1): <flag> <ok | nok> Result of evaluation 2): <checksum> <ok | nok> Result of evaluation action 3) and check 5): <state> <ok | nok> Result of evaluation action 3) and check 6): <flag> <ok | nok> Result of evaluation action 3) and check 7): <checksum> <ok | nok>

TEST CASE RESULTS	CHECK / REACTION	
	Result of evaluation action 4) and check 5): <state> Result of evaluation action 4) and check 6): <flag> Result of evaluation action 4) and check 7): <checksum> Result of evaluation 8):	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

1249

6.7.7 Check structure of Data Storage Index List

Table 103 defines the test conditions for this test case.

Table 103 – Check structure of Data Storage Index List

-CR014- -CR070-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0378
Name	TCD_APPS_DSUP_INDEXLIST
Purpose (short)	Check structure of Data Storage Index List
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device DS test: test to pass
Specification (clause)	B.2.3, Table B.11
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check structure of Index_List and concatenated Index_Lists if available
Precondition	DTU: Communication EUT: PREOPERATE
Procedure	a) Read Index 3, extract Index_List (Subindex 5) b) Store length of Index_List (number of octets) in <len> c) Calculate and store <EntryCount> = (<len>-2) / 3 d) Store termination marker in <term> e) If <term> is not equal 0x00 0x00 read next Index_List from index referenced by <term> and repeat from step b)
Input parameter	
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For each loop cycle check: 1) After step c) check: <EntryCount> must be an integer value in range 0 to 70
Test passed	All evaluations with positive result
Test failed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

6.7.8 Download after modification of parameters

Table 104 defines the test conditions for this test case.

Table 104 – Download after modification of parameters

-CR023-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0082
Name	TCD_APPS_DSDN_PARAMODIFICATION
Purpose (short)	Download after modification of parameters
Equipment under test (EUT)	Device with Datastorage and Block Parameterization support
Test case version	1.2
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	After changing parameters in Device without storing them in DS, download a different parameter set into the Device.
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Call SystemCommand "ParamDownloadStart" via ISDU b) Write different parameter set 2 into the Device c) Call SystemCommand "ParamDownloadEnd" d) Perform Download (restore) of parameter set 1 completely as defined in the Master state machine only if the DS_UPLOAD_FLAG is not set (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 and 2
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether Event DS_UPLOAD_REQ was not raised by the Device Upon each of the following actions: 2) After call of SystemCommand "ParamDownloadStart" 3) After call of SystemCommand "ParamDownloadEnd" 4) After execution of the complete Download check via Data Storage Index the following: 5) State of Data Storage is 0b00 (inactive) 6) DS_UPLOAD_FLAG is not set 7) Parameter_checksum has changed only after 'write parameter set 2' 8) Check whether the Download has been completed without errors
Test passed	All checks in 1), 8), and all three checks during the three actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation 1): <flag> <ok | nok> Result of evaluation action 2) and check 5): <state> <ok | nok> Result of evaluation action 2) and check 6): <flag> <ok | nok> Result of evaluation action 2) and check 7): <checksum> <ok | nok> Result of evaluation action 3) and check 5): <state> <ok | nok> Result of evaluation action 3) and check 6): <flag> <ok | nok> Result of evaluation action 3) and check 7): <checksum> <ok | nok> Result of evaluation action 4) and check 5): <state> <ok | nok> Result of evaluation action 4) and check 6): <flag> <ok | nok> Result of evaluation action 4) and check 7): <checksum> <ok | nok> Result of evaluation 8): <ok | nok>

6.7.9 Download into the Device after reset

Table 105 defines the test conditions for this test case.

Table 105 – Download into the Device after reset

-CR023-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0083
Name	TCD_APPS_DSDN_FACTORYRESET
Purpose (short)	Download into the Device after reset to factory settings
Equipment under test (EUT)	Device with "Reset to factory settings" support and with Datastorage support
Test case version	1.0
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, 10.7, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test of Download of the stored parameter set into the Device after reset to factory settings. Manufacturer to provide parameter set 1 different to factory settings.
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Call SystemCommand "Restore factory settings" via ISDU b) Perform Download (restore) of parameter set 1 completely as defined in the Master state machine only if the DS_UPLOAD_FLAG is not set (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After "Restore factory settings" check whether Event DS_UPLOAD_REQ was not raised by the Device. Upon the following action: 2) After "Restore factory settings" check via Data Storage Index the following: 3) State of Data Storage is correct 4) DS_UPLOAD_FLAG is not set 5) Parameter_checksum has changed 6) Check whether the Download has been completed without errors
Test passed	All checks in 1), 6), and all three checks during the action described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	Result of evaluation 1): <flag> <ok | nok> Result of evaluation action 2) and check 3): <state> <ok | nok> Result of evaluation action 2) and check 4): <flag> <ok | nok> Result of evaluation action 2) and check 5): <checksum> <ok | nok> Result of evaluation 6): <ok | nok>

6.7.10 Call ParamBreak in different states of Download

Table 106 defines the test conditions for this test case.

Table 106 – Call ParamBreak in different states of Download

-CR023-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0084
Name	TCD_APPS_DSDN_PARABREAKABORT
Purpose (short)	Download abort via SystemCommand "ParamBreak" in different states
Equipment under test (EUT)	Device with Datastorage and Block Parameterization support
Test case version	1.3
Category / type	Device DS test: test to pass
Specification (clause)	[6], see 10.4.2, Tables B.8, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers Download aborts via SystemCommand "ParamBreak" in different states. Manufacturer is responsible for the definition of two possible "parameter sets".
Precondition	- Device in PREOPERATE or OPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set - Test preparation e.g. using DS_Command "DS_UploadStart", upload of parameter set 1" using the Index List (upload), followed by a "DS_UploadEnd"
Procedure	a) Start Download via SystemCommand "ParamDownloadStart" b) Call SystemCommand "ParamBreak" directly after "ParamDownloadStart" c) Start Download via SystemCommand "ParamDownloadStart" d) Transmit first Block Parameter object of Config4 with data of parameter set 2 e) Call SystemCommand "ParamBreak" f) Start Download via SystemCommand "ParamDownloadStart" g) Transmit all Block Parameter objects of Config4 with data of parameter set 2 h) Call SystemCommand "ParamBreak" i) Perform Download of parameter set 2 completely as defined in the Master state machine only if the DS_UPLOAD_FLAG is not set in the Device (switch Master DS from deactivated to activated)
Test parameter	Parameter set 1 and 2
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	After call of each SystemCommand "ParamBreak" (3x) check via Data Storage Index the following: 1) State of Data Storage is 0b00 (inactive) 2) DS_UPLOAD_FLAG is not set 3) Parameter_checksum has not changed After i): 4) Check whether the Download has been completed without errors
Test passed	All checks during the actions described in evaluation are positive
Test not passed (examples)	Any check in evaluation failed
Report	All evaluations: <ok | nok>

6.7.11 Parameter Manager – Single Parameter and DownloadStore

Table 107 defines the test conditions for this test case.

Table 107 – Parameter Manager – Single Parameter and DownloadStore

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0321
Name	TCD_DSBP_APPL_DSSINGLEPARAM
Purpose (short)	Test of single parameter write in Parameter Manager Idle
Equipment under test (EUT)	Device with Data Storage as indicated in IODD
Test case version	1.0
Category / type	Device DS test: test to pass
Specification (clause)	[6], 10.3.2 and 10.3.4
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall not invoke DataStorage if parametrized via single parameter accesses. Device shall invoke DataStorage after receiving a SystemCommand "ParamDownloadStore".
Precondition	DTU: Communication EUT: No Event pending or active DataStorage of the Device is inactive and DS_UPLOAD_FLAG is "0"
Procedure	a) Write test parameter with different content than stored in Device b) Wait for DS_UPLOAD_REQ Event with a timeout of 1 s c) Read DataStorage state property: Index 3, Subindex 2 d) Write SystemCommand "ParamDownloadStore" e) Wait for DS_UPLOAD_REQ Event with a timeout of 1 s f) Read Data Storage state property: Index 3, Subindex 2 g) Write DS_Command "DS_UploadEnd"
Test parameter	One parameter of the parameter sets defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After step b), check that timeout occurred and no Event with code "DS_UPLOAD_REQ" has been received 2) After step c), check that State_Property.DS_UPLOAD_FLAG = "0" 3) After step e), check that no timeout occurred and Event with code "DS_UPLOAD_REQ" has been received 4) After step f), check that State_Property.DS_UPLOAD_FLAG = "1"
Test passed	All evaluations with positive result
Test not passed (examples)	No response or any evaluation failed
Report	Single Parameter without DS activity <ok | nok> DS_DownloadStore in ParameterManager state "Idle_0" <ok | nok>

6.7.12 Clear DS_UPLOAD_FLAG after DS_Up/DownloadEnd

Table 108 defines the test conditions for this test case.

Table 108 – Clear DS_UPLOAD_FLAG after DS_Up/DownloadEnd

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0322
Name	TCD_DS_APP_IDLEFLAGCLEAR
Purpose (short)	DS_UPLOAD_FLAG is cleared after TransmissionEnd while in DSIdle_2
Equipment under test (EUT)	Device with Data Storage indicated in IODD
Test case version	1.0
Category / type	Device DS test: Test to pass
Specification (clause)	[6], see 10.4.2, Table B.11, Table D.2, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	DS_UPLOAD_FLAG is cleared when DS_UploadEnd or DS_DownloadEnd command is received while Data Storage state machine is in state DSIdle_2 (Transition T11)
Precondition	DTU: OPERATE EUT: DS state machine is in state DSIdle_2
Procedure	a) Write SystemCommand “ParamDownloadStore” b) Read DataStorageIndex c) Write DS Command “DS_UploadEnd” d) Read DataStorageIndex e) Write SystemCommand “ParamDownloadStore” f) Read DataStorageIndex g) Write DS Command “DS_DownloadEnd h) Read DataStorageIndex
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "DS_UPLOAD_FLAG" in step b), d), f), and h)
Test passed	In b) = "1" In d) = "0" In f) = "1" In h) = "0"
Test not passed (examples)	Any check failed
Report	In b): <flag> <ok | nok> In d): <flag> <ok | nok> In f): <flag> <ok | nok> In h): <flag> <ok | nok>

6.7.13 Storage of DS_UPLOAD_FLAG in non volatile memory

-CR016- -CR043-

Table 109 defines the test conditions for this test case.

Table 109 – Storage of DS_UPLOAD_FLAG in non volatile memory

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0374
Name	TCD_DS_APP_UPLOAD_FLAG_NON_VOLATILE
Purpose (short)	Storage of DS_UPLOAD_FLAG in non volatile memory
Equipment under test (EUT)	Device with Data Storage support
Test case version	1.0
Category / type	Device DS test: test to pass
Specification (clause)	[7], see 10.4.2, Tables B.8, B.9, B.11, and D.1, Figure 90
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test covers parameterization (parameter set 2) of a Device and initiation of the Upload via SystemCommand "ParamDownloadStore". Device shall store the DS_UPLOAD_FLAG in non volatile memory
Precondition	- Device in PREOPERATE mode - Parameter set 1 stored within Device (manufacturer to define parameter set 1) - DS_UPLOAD_FLAG is not set
Procedure	a) Call SystemCommand "ParamDownloadStore" (causes Event DS_UPLOAD_REQ) b) Wait for event DS_UPLOAD_REQ c) Check DS_UPLOAD_FLAG in parameter DataStorageIndex d) Apply power cycle (power off, wait 5 s, power on: Master restarts communication) e) Check if a communication startup sequence has been triggered, and Device is in PREOPERATE f) Check DS_UPLOAD_FLAG in parameter DataStorageIndex g) Reset the DS_UPLOAD_FLAG with the call of the System Command “DS_UploadStart”, followed by the system command “DSUploadEnd” h) Check DS_UPLOAD_FLAG in parameter DataStorageIndex
Test parameter	-
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check whether Event DS_UPLOAD_REQ was raised automatically by the Device after b) 2) check via Data Storage Index: the DS_UPLOAD_FLAG is set after c) 3) Startup Sequence has been detected in e), and Device is in PREOPERATE state 4) Value of procedure f) (DS_UPLOAD_FLAG) = "1" (active DS_UPLOAD_REQ) 5) Value of procedure h) (DS_UPLOAD_FLAG) = "0" (no active DS_UPLOAD_REQ)
Test passed	All checks are positive
Test failed (examples)	Any of the checks failed
Report	Device has stored the DS_UPLOAD_FLAG non volatile: <yes/no> <ok | nok>

6.8 Operation with a legacy Master ("Master 1.0")

6.8.1 General

Designers of Devices according to [6] ("Device 1.1") should be aware of the possibility that such a Device in the field can be connected to a Master ("Master 1.0") designed according to a previous specification [5]. Therefore, the conformity class requirements and the associated test cases are specified within the following clauses. It should be noted that the IODD of such a Device plays an important role in establishing the right behavior in respect to Process Data exchange and cycle times (interleave mode). The compatibility rules for IODDs are defined in [3].

6.8.2 Conformity classes

6.8.2.1 Master conformity

The "Masters 1.0" in the field are supposed to be conform with [5]. By design according to [1], the "Masters 1.1" shall be compatible to any legacy "Device 1.0". Therefore, no special compatibility rules are required for Master and no conformity classes.

6.8.2.2 "Device 1.1" without backward compatibility

The Device requires features that only a Master provides, which is designed according to [6] or a later version. Thus, usually it can deny SDCI communication with a "Master 1.0". Example is a Device with large Process Data (PD). If this Device would be used with a "Master 1.0" and an M-sequence TYPE_1, the Process Data cycle could last much longer than with a "Master 1.1" (see Annex B.5).

6.8.2.3 "Device 1.1" compatible with a "Master 1.0"

There exist two main possibilities to design a "Device 1.1" compatible to a "Master 1.0".

- A "Device 1.1" can be adjusted to a behavior according to [5] through setting of a specific parameter using ISDU services ("Device 1.0"). In this case no new test cases are required due to an SDCI communication compatible to [5].
- A "Device 1.1" will automatically detect connection to a "Master 1.0" and switch to an SDCI communication compatible to [5]. For this case, the restrictions for "Master 1.0" and "Device 1.1" are defined in the following.

The compatibility restrictions or constraints for a "Master 1.0" being able to support automatic SDCI communication version detection are:

- As defined in the state machines of [6], the "Device 1.1" will send the protocol version 1.1 (or a later one) via parameter 0x04 (RevisionID) to the "Master 1.0" during the startup phase. The "Master 1.0" ignores this version number. If the "Master 1.0" insists in protocol version 1.0, the "Device 1.1" cannot be used with this Master.
- During reading of the parameters 0x02 to 0x06 (Direct Parameter page 1) in the STARTUP phase, the "Device 1.1" cannot detect the Master version. For this reason, some of the reserved bits in the parameter 0x03 (M-sequence Capability) in [5] are set in the "Device 1.1". The "Master 1.0" ignores these bits. Otherwise, the "Device 1.1" cannot be used with this "Master 1.0".

6.8.3 From STARTUP to OPERATE (V1.0)

Table 110 defines the test conditions for this test case.

Table 110 – From STARTUP to OPERATE (V1.0)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0085
Name	TCD_DLIC_COMP_STARTUP
Purpose (short)	Establish a connection from Wakeup to OPERATE according to V1.0 SDCI protocol
Equipment under test (EUT)	Device with "V1.0" support
Test case version	1.2
Category / type	Device legacy Master test: test to pass
Specification (clause)	[5]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Establish a connection from Wakeup to OPERATE in V1.0 way of SDCI protocol. The Device-Tester initiates a Wakeup request, reads out the parameter 0x02 – 0x06 (DPP1) and then tries to turn the Device in OPERATE mode. After this, the Device shall be able to exchange Process Data and accept ISDU services. This test shall ensure that a V1.1 Device can be accessed from a V1.0 Master.
Precondition	–
Procedure	a) Master initiates a Wakeup (with reading of parameter 0x02) b) After the transmission rate detection the Master reads parameter 0x03 – 0x06 c) After reading the parameter it will write the MasterCycleTime d) Master sends MasterCommand 0x99 (OPERATE) e) Master switches to target M-sequence type and exchanges Process Data f) Master initiates a Read or Write to DPP1 to ensure response g) Master initiates an ISDU Read or Write to test the Device capability (only in case of ISDU support)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check if Device responds with the correct values during STARTUP 2) Check if the Device responds with the correct M-sequence type after OPERATE 3) Check if the Device supports ISDU (highly recommended) 4) Check if the ISDU Read or Write is responded (only if ISDU is supported)
Test passed	Evaluations 1), 2), and 4) successful
Test not passed (examples)	Evaluations 1), 2), or 4) failed
Report	Exchange of PD: <yes/no> <ok | nok> ISDU is working: <yes/no> <ok | nok> No ISDU support: <yes/no> <exeption>

6.8.4 From STARTUP to OPERATE – interleave (V1.0)

Table 111 defines the test conditions for this test case.

Table 111 – From STARTUP to OPERATE - interleave (V1.0)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0086
Name	TCD_DLIC_COMP_TYPE1INTERLEAVE
Purpose (short)	Interleave test
Equipment under test (EUT)	Device with "V1.0" support
Test case version	1.2
Category / type	Device legacy Master test: test to pass
Specification (clause)	[5]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Interleave test. In V1.1 the SDCI protocol defines new M-sequence types for large Process Data transfers (more than 2 octets). In Version V1.0, SDCI communication uses the TYPE_1_x M-sequences with interleaving of PD and OD. A compatible Device shall be able to switch to the interleave mode (TYPE_1_x) during communication with the Device-Tester.
Precondition	Device supports more than 2 octets of Process Data
Procedure	a) Master initiates a Wakeup request (with reading of parameter 0x02) b) After detection of the transmission rate the Master reads parameter 0x03 – 0x06 c) After reading the parameter it will write the MasterCycleTime to 0x01 d) Master sends MasterCommand 0x99 (OPERATE) e) Master switches to M-sequence TYPE_1_1/2 and exchanges Process Data f) Master initiates a Read or Write DPP1 to ensure a response from the Device g) Master initiates an ISDU Read or Write to ensure a response from the Device (only in case of ISDU support)
Test parameter	"PD size" taken from the IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check if Device responds with the correct values during STARTUP 2) Check if the Device responds with the correct M-sequence type after OPERATE 3) Check if the Read or Write to DPP1 is responded 4) Check if the ISDU Read or Write is responded
Test passed	Evaluations 1), 2), and 4) successful
Test not passed (examples)	Evaluations 1), 2), or 4) failed
Report	Exchange of PD in interleave mode: <yes/no> <ok | nok> Read or Write to DPP1 is working: <yes/no> <ok | nok> ISDU is working: <yes/no> <ok | nok> No ISDU support: <yes/no> <exeption>

6.8.5 Events – PDInvalid / PDValid (V1.0)

Table 112 defines the test conditions for this test case.

Table 112 – Events – PDInvalid / PDValid (V1.0)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0087
Name	TCD_DLIC_COMP_PDINVALIDEVENT
Purpose (short)	Send Event for PDVALID/INVALID
Equipment under test (EUT)	Device NOTE For Devices without PDInvalid Flag support, this test case can be skipped. No test interface required.
Test case version	1.1
Category / type	Device legacy Master test: test to pass
Specification (clause)	[5]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Send Event for PDVALID/INVALID. This test to ensure that a Device can indicate the validity of the Process Data inputs GOOD or BAD via an Event in a V1.0 SDCI communication (In V1.0, the Master does not interpret the bit in the Process Data).
Precondition	Communication is established
Procedure	a) Stimulate an error in the Device, which leads to a Process Data failure b) The Device shall send this status change via an Event
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	Check if the Device sent an Event with PDValid/Invalid to the Master
Test passed	DTU received an Event for the validity change of Process Data (GOOD and BAD)
Test not passed (examples)	Device did not send an Event
Report	Device sent "GOOD/BAD" Event: <yes/no> <ok | nok>

6.9 Direct Parameter page 1

6.9.1 MasterCycleTime

Table 113 defines the test conditions for this test case.

Table 113 – MasterCycleTime

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0089
Name	TCD_DLPC_STDP_MASTERCYCLETIME
Purpose (short)	Correct value of MasterCycleTime
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.3
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test the correct value of MasterCycleTime in DPP1. The value shall match the value transmitted by the Master (DTU).
Precondition	DTU: Port inactive EUT: –
Procedure	a) Set DTU to communication *;save MasterCycleTime(Master)* b) Read DPP1.MasterCycleTime *;returns MasterCycleTime(Device)*
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) After b), compare MasterCycleTime(Master) with MasterCycleTime(Device)
Test passed	Values match
Test not passed (examples)	Mismatch of values
Report	MasterCycleTime(Master): <value> MasterCycleTime(Device): <value> <ok | nok>

6.9.2 MinCycleTime

Table 114 defines the test conditions for this test case.

Table 114 – MinCycleTime

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0090
Name	TCD_DLPC_STDP_MINCYCLETIME
Purpose (short)	Correct setting of MinCycleTime as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.3, Figure B.2, Table B.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test value of MinCycleTime. The value shall match the Device specific default settings as indicated in IODD and it shall be valid according to specified coding.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.MinCycleTime *;returns MinCycleTime(Device)*
Test parameter	MinCycleTimeIODD = *node PhysicalLayer, attribute minCycleTime in IODD*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After a), compare MinCycleTime(Device) with MinCycleTimeIODD
Test passed	MinCycleTime(Device) = MinCycleTimeIODD and Time Base < 3
Test not passed (examples)	MinCycleTime(Device) ≠ MinCycleTimeIODD or Time Base = 3
Report	MinCycleTimeIODD: <value> MinCycleTime(DEVICE): <value> <ok | nok> Time Base: <value> <ok | nok>

6.9.3 M-sequenceCapability

Table 115 defines the test conditions for this test case.

Table 115 – M-sequenceCapability

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0091
Name	TCD_DLPC_STDP_FSEQCAPABILITY
Purpose (short)	Correct M-sequence type entries as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.2
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.4, Figure B.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test parameter M-sequenceCapability for a valid value according to specified coding. Values shall match the Device specific settings as specified in IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.M-sequenceCapability *;returns M-sequenceCapability(Device)*
Test parameter	M-sequenceCapabilityIODD = *node PhysicalLayer, attribute mSequenceCapability*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After a), check value of M-sequenceCapability
Test passed	M-sequenceCapability(Device) = M-sequenceCapabilityIODD and bit 6,7 = "0"
Test not passed (examples)	M-sequenceCapability(Device) ≠ M-sequenceCapabilityIODD or bit 6,7 ≠ "0"
Report	M-sequenceCapabilityIODD: <value> M-sequenceCapability(Device): <value> <ok | nok>

6.9.4 RevisionID

Table 116 defines the test conditions for this test case.

Table 116 – RevisionID

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0092
Name	TCD_DLPC_STDP_REVISIONID
Purpose (short)	Correct default protocol revision as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.5, Figure B.4
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test value of the protocol revision of Device. Value shall match the revision defined in IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.RevisionID *;returns RevisionID(Device)*
Test parameter	ProtocolRevisionIODD = *node CommNetworkProfile, attribute iolinkRevision*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After a), check RevisionID(Device)
Test passed	RevisionID(Device) = ProtocolRevisionIODD
Test not passed (examples)	RevisionID(Device) does not match
Report	ProtocolRevisionIDIODD: <value> RevisionID(Device): <value> <ok | nok>

6.9.5 ProcessDataIn

Table 117 defines the test conditions for this test case.

Table 117 – ProcessDataIn

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0093
Name	TCD_DLPC_STDP_PDIN
Purpose (short)	Correct default ProcessDataInput value as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.6, Figure B.5, Table B.5, Table B.6
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the values of ProcessDataInput. The value shall match the value as specified in the IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.ProcessDataIn *;returns ProcessDataIn(Device)*

Test parameter	ProcessDataInIODD = *node ProcessData.ProcessDataIn, attribute bitlength*
	SIOsupportIODD = *node PhysicalLayer, attribute sioSupported*

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After a), check ProcessDataIn(Device)
Test passed	Process Data length and SIO bit match specified values, and Process Data length unit is a valid value, and Bit 5 is "0", and ProcessDataIn(Device) = ProcessDataInIODD
Test not passed (examples)	Any of the evaluations failed
Report	ProcessDataInIODD: <value> SIOsupportIODD: <value> ProcessDataIn(Device): <value> <ok | nok>

6.9.6 ProcessDataOut

Table 118 defines the test conditions for this test case.

Table 118 – ProcessDataOut

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0094
Name	TCD_DLPC_STDP_PDOUT
Purpose (short)	Correct default ProcessDataOutput value as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.7, Figure B.5, Table B.6
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the values of Process Data Output. The value shall match the value as specified in the IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.ProcessDataOut *;returns ProcessDataOut(Device)*
Test parameter	ProcessDataOutIODD = *node ProcessData.ProcessDataOut, attribute bitLength*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After a), check ProcessDataOut(Device)
Test passed	Process Data length match specified values, and Process Data length unit is a valid value, and Bit 5 and 6 are "0", and ProcessDataOut(Device) = ProcessDataOutIODD
Test not passed (examples)	Any of the evaluations failed
Report	ProcessDataOutIODD: <value> ProcessDataOut(Device): <value> <ok | nok>

6.9.7 VendorID

Table 119 defines the test conditions for this test case.

Table 119 – VendorID

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0095
Name	TCD_DLPC_STDP_VENDORID
Purpose (short)	Correct VendorID as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the value of the VendorID. The value shall match the unique ID assigned to the vendor and the value specified in the IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.VendorID1 *;returns VendorID1* b) Read DPP1.VendorID2 *;returns VendorID2* c) Combine VendorID1 and VendorID2 to VendorID(Device)

Test parameter	VendorIDIODD = *node DeviceIdentity, attribute vendorId*
	VendorIDAssigned = *VendorID @ (https://iolink.com/share/Downloads/Vendor_ID_Table.xml)*

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After c), check VendorID(Device)
Test passed	VendorID(Device) = VendorIDIODD, and VendorID(Device) = VendorIDAssigned
Test not passed (examples)	Any of the evaluations failed
Report	VendorIDIODD: <value> VendorIDAssigned: <value> VendorID(Device): <value> <ok | nok>

6.9.8 DeviceID

Table 120 defines the test conditions for this test case.

Table 120 – DeviceID

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0096
Name	TCD_DLPC_STDP_DEVICEID
Purpose (short)	Correct default DeviceID as indicated in IODD
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the value of the DeviceID. The value shall match the ID assigned by the vendor and the value specified in the IODD.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.DeviceID1 *;returns DeviceID1* b) Read DPP1.DeviceID2 *;returns DeviceID2* c) Read DPP1.DeviceID3 *;returns DeviceID3* d) Combine DeviceID1, and DeviceID2, and DeviceID3 to DeviceID(Device)
Test parameter	DeviceIDIODD = *node DeviceIdentity, attribute deviceId*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After d), check DeviceID(Device)
Test passed	DeviceID(Device) = DeviceIDIODD, and DeviceID(Device) > 0
Test not passed (examples)	Any of the evaluations failed
Report	DeviceIDIODD: <value> DeviceID(Device): <value> <ok | nok>

6.9.9 FunctionID

Table 121 defines the test conditions for this test case.

Table 121 – FunctionID

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0097
Name	TCD_DLPC_STDP_FUNCTIONID
Purpose (short)	Correct FunctionID (reserved)
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.10
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test the value of the FunctionID. The FunctionID is not used and shall contain the default value.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.FunctionID1 *;returns FunctionID1* b) Read DPP1.FunctionID2 *;returns FunctionID2* c) Combine FunctionID1 and FunctionID2 to FunctionID(Device)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) After c), check FunctionID(Device)
Test passed	FunctionID(Device) = "0"
Test not passed (examples)	FunctionID(Device) ≠ "0"
Report	FunctionID(Device): <value> <ok | nok>

6.9.10 Reserved parameter – Read/Write

Table 122 defines the test conditions for this test case.

Table 122 – Reserved parameter – Read/Write

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0101
Name	TCD_DLPC_STDP_WRITERESPAR
Purpose (short)	Test reserved Direct Parameter write behavior
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device DPP test: test to pass
Specification (clause)	[6], see B.1.1
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test the Device behavior and read result after write access to a reserved Direct Parameter
Precondition	Device is in SDCI communication mode
Procedure	Write values 0x00 to 0xFF via the Device-Tester to reserved parameters on Direct Parameter page 1 (Address 0x0E)
Input parameter	-
Post condition	-
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	Check validity of a Write access to reserved Direct Parameter.
Test passed	If no communication errors occurred
Test failed (examples)	If communication errors or Events occurred
Report	Communication errors: <no/yes> <passed/failed>

6.9.11 Validity of MinCycleTime

-CR017- -CR094-

Table 123 defines the test conditions for this test case.

Table 123 – Validity of MinCyleTime

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0375
Name	TCD_DLPC_STDP_MINCYCLETIME_VALID
Purpose (short)	Check validity of MinCycleTime against best case timing and allowed ranges
Equipment under test (EUT)	Device with MinCycleTime <> zero
Test case version	1.0
Category / type	Device DPP test, test to pass
Specification (clause)	B.1.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test value of MinCycleTime against best case timing, means the provided time shall not be lower than the shortest possible timing calculated according clauses A.3.4 to A.3.6 with t_1, t_2, t_{idle} equal zero T_{BIT} and t_A equal one T_{BIT}. Further the ranges of Table B.3 are checked for violations.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read DPP1.MinCycleTime b) Read DPP1.M-sequenceCapability c) Read DPP1.ProcessDataIn d) Read DPP1.ProcessDataOut
Input parameter	-
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) After reading the communication parameters, calculate best timing with t_1, t_2, t_{idle} equal zero and t_A equal one based on the proposed M-sequence type and process data width. 2) Check MinCycleTime against allowed ranges defined in Table B.3
Test passed	MinCycleTime equal or greater than best timing from 1) MinCycleTime does not violate allowed ranges according 2)
Test failed (examples)	Min CycleTime lower than best timing or violating ranges according 2)
Report	MinCycleTime in range <ok | nok>

6.10 Predefined Device parameters

6.10.1 General rules

Predefined parameters shall be tested in any case. The following rules apply:

They shall be tested as specified within the test cases if they are defined within the IODD.

They shall *not* be tested as specified within the test cases if they are *not* defined within the IODD.

All optional test cases for Predefined Parameters shall be handled according to rule b)

6.10.2 System command – reserved commands

Table 124 defines the test conditions for this test case.

Table 124 – System command – reserved commands

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0104
Name	TCD_DLIC_DEFP_SYSCMDRES
Purpose (short)	SystemCommand behavior upon reserved commands (via ISDU)
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see B.2.2, and Annex C.2.1, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies the correct response values upon usage of reserved and unused SystemCommands.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write subsequently the following values to parameter SystemCommand: - 0x00 - 0x01 to 0x04, if Block Parameter transfer is not supported - 0x05, if neither Data Storage nor Block Parameter transfer is supported - 0x06, if Block Parameter transfer is not supported - 0x07 to 0x3F - 0x40 to 0x7F, if no Profile is supported - 0x80 to 0x83, if not referenced in IODD - 0x84 to 0x9F - 0xA0 to 0xFF, if not referenced in IODD
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check each Write response in a)
Test passed	All Write error responses = FUNC_NOTAVAIL (0x8035) If Write error response ≠ FUNC_NOTAVAIL (0x8035), indicate value and warning
Test not passed (examples)	Positive or no Write response
Report	Write response (reserved SystemCommand): <value> <ok | nok>

6.10.3 Data Storage Index – entire parameter structure

Table 125 defines the test conditions for this test case.

Table 125 – Data Storage Index – entire parameter structure

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0107
Name	TCD_DLIC_DEFP_DSINDEX
Purpose (short)	Behavior of parameter DataStorageIndex
Equipment under test (EUT)	Device
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.3, Table B.9; see B.2.3 (records without gaps)
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies correct implementation of parameter DataStorageIndex. Devices without Data Storage support shall support the entire parameter structure.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter DataStorageIndex (Index 0x03)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response of Read access 2) Check parameter length 3) Calculate <index entries> = (parameter length – 12)/3
Test passed	No negative response, and <index entries> has a positive integer value in the range ≥ 0 and ≤ 70
Test not passed (examples)	Negative response, or <index entries> outside the permitted range
Report	Data Storage Index, length: <index entries> <ok | nok>

6.10.4 Data Storage Index – record items

Table 126 defines the test conditions for this test case.

Table 126 – DataStorageIndex – record items

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0108
Name	TCD_DLIC_DEFP_DSRECORD
Purpose (short)	Behavior of parameter DataStorageIndex record items
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.3, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies the correct structure of parameter and record items of the DataStorageIndex.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read subsequently Subindex 1 to 5 of DataStorageIndex (Index 0x03)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response of Read access 2) Check record item length 3) Check value of record item
Test passed	No negative response on record item Read access, and Subindex 1 has a length of 1 octet and value is within range ≥ 0 to ≤ 5 Subindex 2 has a length of 1 octet and value of bit 0 and bit 3 - 6 is "0" Subindex 3 has a length of 4 octets and value is within range 0 to 2048 Subindex 4 has a length of 4 octets Subindex 5 has a length of ≥ 2 octets in increments of 3 (2,5,8,11,14, until 212)
Test not passed (examples)	Record items are deviating in length or value range
Report	DataStorageIndex, Subindex 1: <length, value> <ok | nok> DataStorageIndex, Subindex 2: <length, value> <ok | nok> DataStorageIndex, Subindex 3: <length, value> <ok | nok> DataStorageIndex, Subindex 4: <length> <ok | nok> DataStorageIndex, Subindex 5: <length> <ok | nok>

6.10.5 Data Storage Index – termination marker

Table 126 defines the test conditions for this test case.

Table 127 – DataStorageIndex – termination marker

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0323
Name	TCD_DLIC_DEFP_DSRECORDMARKER
Purpose (short)	Behavior of parameter DataStorageIndex record items – Termination marker
Equipment under test (EUT)	Device without DS support
Test case version	1.0
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.3, Table B.9, and 10.4.5
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies the correct entries in items Data_Storage_Size and DS Index_List of the parameter DataStorageIndex. Both items shall be "0".
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read subsequently Subindex 1 to 5 of DataStorageIndex (Index 0x03)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response of Read access 2) Check record item length 3) Check value of record item
Test passed	No negative response on record item Read access, and Subindex 1 has a length of 1 octet and value is within range ≥ 0 to ≤ 5 Subindex 2 has a length of 1 octet and value of bit 0 and bit 3 - 6 is "0" Subindex 3 has a length of 4 octets and value = "0" Subindex 4 has a length of 4 octets Subindex 5 has a length of 2 octets and value = "0"
Test not passed (examples)	Record items are deviating in length or value range
Report	DataStorageIndex, Subindex 1: <length, value> <ok | nok> DataStorageIndex, Subindex 2: <length, value> <ok | nok> DataStorageIndex, Subindex 3: <length, value> <ok | nok> DataStorageIndex, Subindex 4: <length> <ok | nok> DataStorageIndex, Subindex 5: <length, value > <ok | nok>

6.10.6 Device Access Locks – valid

Table 128 defines the test conditions for this test case.

Table 128 – Device Access Locks – valid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0109
Name	TCD_DLIC_DEFP_ACCESSLOCKSVAL
Purpose (short)	Behavior of DeviceAccessLocks with valid values
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], 10.6.5, 10.6.6, Annex B.2.4, Table B.12
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies that all implemented values for DeviceAccessLocks are stored and responded correctly.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write value to DeviceAccessLocks with "1" at every bit position marked as implemented and "0" at the remaining bits. b) Read value DeviceAccessLocks c) Write value 0x0000 to DeviceAccessLocks d) Read value DeviceAccessLocks
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check for response on Read and Write access 2) Set warning flag, if a positive response is received for a write access with “1” at bit positions "0" or "1" 3) Compare response value from step b) with written value from step a) 4) Compare response value from step d) with written value from step c)
Test passed	No negative response on Read or Write access, and Evaluations 3) and 4) show matching values
Test not passed (examples)	No matching values
Report	DeviceAccessLocks implemented: <written/read values> <ok | nok> DeviceAccessLocks 0x0000: <written/read values> <ok | nok> Warning "Locking implemented" (see NOTE): <exception>
NOTE If warning flag is set, text shall be displayed: "It is highly recommended not to implement the features "parameter access locking" or "Data Storage locking" since they will be removed in future releases of the specification. Vendor to inform user".	

6.10.7 Device Access Locks – invalid

Table 129 defines the test conditions for this test case.

Table 129 – Device Access Locks – invalid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0110
Name	TCD_DLIC_DEFP_ACCESSLOCKSINVAL
Purpose (short)	Behavior of DeviceAccessLocks with invalid values
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.0
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.4, Table B.11
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The test verifies that all reserved or unused values for DeviceAccessLocks are responded correctly.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write to DeviceAccessLocks subsequently the value "1" at a single bit position, which is marked as reserved and a value "0" at the remaining bit positions b) Read value DeviceAccessLocks c) Repeat from a) with next bit position
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check responses in step a) 2) Check response values in step b)
Test passed	Each response in 1) = PAR_VALOUTOFRNG (0x8030), and Each value = 0x0000
Test not passed (examples)	No negative response Any value ≠ 0x0000
Report	Device Access Locks reserved: <written/read values> <ok | nok> Device Access Locks 0x0000: <written/read values> <ok | nok>

6.10.8 Profile Characteristic

Table 130 defines the test conditions for this test case.

Table 130 – Profile Characteristic

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0111
Name	TCD_DLIC_DEFP_PROFILCHARAC
Purpose (short)	Behavior of parameter ProfileCharacteristic
Equipment under test (EUT)	Device without supported profiles
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.5; [3]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test for implementation of parameter ProfileCharacteristic
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter ProfileCharacteristic (Index 0x000D)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response
Test passed	Negative response with ErrorType = 0x8011 – *Index not available*
Test not passed (examples)	Different ErrorType or positive response
Report	Response: <negative/positive> <ok | nok> ErrorType: <value> <ok | nok>

6.10.9 PD Input Descriptor

Table 131 defines the test conditions for this test case.

Table 131 – PD Input Descriptor

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0112
Name	TCD_DLIC_DEFP_PDINDESC
Purpose (short)	Behavior of parameter PDInputDescriptor
Equipment under test (EUT)	Device without supported profiles
Test case version	1.3
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.6; [3]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test for implementation of parameter PDInputDescriptor
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter PDInputDescriptor (Index 0x000E)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response
Test passed	Negative response with ErrorType 0x8011 – *Index not available*
Test not passed (examples)	Different ErrorType or positive response
Report	Response: <negative/positive> <ok | nok> ErrorType: <value> <ok | nok>

6.10.10 PD Output Descriptor

Table 132 defines the test conditions for this test case.

Table 132 – PD Output Descriptor

TEST CASE ATTRIBUTES	**IDENTIFICATION / REFERENCE**
Identification (ID)	SDCI_TC_0113
Name	TCD_DLIC_DEFP_PDOUTDESC
Purpose (short)	Behavior of parameter PDOutputDescriptor
Equipment under test (EUT)	Device without supported profiles
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.7; [3]
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test for implementation of parameter PDOutputDescriptor
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter PDOutputDescriptor (Index 0x000F)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check response
Test passed	Negative response with ErrorType 0x8011 – *Index not available*
Test not passed (examples)	Different ErrorType or positive response
Report	Response: <negative/positive> <ok | nok> ErrorType: <value> <ok | nok>

6.10.11 Vendor Name

Table 133 defines the test conditions for this test case.

Table 133 – Vendor Name

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0114
Name	TCD_DLIC_DEFP_VENDORNAM
Purpose (short)	Behavior of parameter VendorName
Equipment under test (EUT)	Device with ISDU support
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.6
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter VendorName
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter VendorName (Index 0x0010)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length 4) Check contents of parameter	
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches registered vendor for VendorID (VID)	
Test not passed (examples)	Any of the evaluations fails	
Report	VendorName response: <negative/positive> VendorName UTF8 coding: <yes/no> VendorName length: <value> VendorName registration: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

6.10.12 Vendor Text

Table 134 defines the test conditions for this test case.

Table 134 – Vendor Text

-CR117-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0115
Name	TCD_DLIC_DEFP_VENDORTEXT
Purpose (short)	Behavior of parameter VendorText
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.7
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter VendorText
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter VendorText (Index 0x0011)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check response on Read access 2) Check coding of parameter 3) Check parameter length 4) Check contents of parameter if default value is present in IODD.	
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information	
Test not passed (examples)	Any of the evaluations fails	
Report	VendorText response: <negative/positive> VendorText UTF8 coding: <yes/no> VendorText length: <value> VendorText adequate: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

6.10.13 Product Name

Table 135 defines the test conditions for this test case.

Table 135 – Product Name

-CR117-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0116
Name	TCD_DLIC_DEFP_PRODUCTNAM
Purpose (short)	Behavior of parameter ProductName
Equipment under test (EUT)	Device with ISDU support
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter ProductName
Precondition	DTU: Communication EUT: OPERATE
Procedure	Read parameter ProductName (Index 0x0012)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length 4) Check contents of parameter if default value is present in IODD.	
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information	
Test not passed (examples)	Any of the evaluations fails	
Report	ProductName response: <negative/positive> ProductName UTF8 coding: <yes/no> ProductName length: <value> ProductName adequate: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

6.10.14 Product ID

Table 136 defines the test conditions for this test case.

Table 136 – Product ID

-CR117-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0117
Name	TCD_DLIC_DEFP_PRODUCTID
Purpose (short)	Behavior of parameter ProductID
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter ProductID.
Precondition	DTU: Communication EUT: OPERATE
Procedure	Read parameter Product ID (Index 0x0013)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length 4) Check contents of parameter if default value is present in IODD.
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information
Test not passed (examples)	Any of the evaluations fails
Report	ProductID response: <negative/positive> <ok | nok> ProductID UTF8 coding: <yes/no> <ok | nok> ProductID length: <value> <ok | nok> ProductID adequate: <yes/no> <ok | nok>

6.10.15 Product Text

Table 137 defines the test conditions for this test case.

Table 137 – Product Text

-CR117-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0118
Name	TCD_DLIC_DEFP_PRODUCTTEXT
Purpose (short)	Behavior of parameter ProductText
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.10
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter ProductText
Precondition	DTU: Communication EUT: OPERATE
Procedure	Read parameter Product Text (Index 0x0014)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length 4) Check contents of parameter if default value is present in IODD.	
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information	
Test not passed (examples)	Any of the evaluations fails	
Report	ProductText response: <negative/positive> ProductText UTF8 coding: <yes/no> ProductText length: <value> ProductText adequate: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

6.10.16 Serial Number

Table 138 defines the test conditions for this test case.

Table 138 – SerialNumber

-CR123-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0119
Name	TCD_DLIC_DEFP_SERNUM
Purpose (short)	Behavior of parameter SerialNumber
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.11
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter SerialNumber
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter SerialNumber (Index 0x0015)
Test parameter	SerialNumber of the EUT provided by the manufacturer
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length	
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 16 octets, and Contents matches vendor / Device specific information	
Test not passed (examples)	Any of the evaluation steps fails	
Report	SerialNumber response: <negative/positive> SerialNumber UTF8 coding: <yes/no> SerialNumber length: <value> SerialNumber adequate: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok>

6.10.17 Hardware Revision

Table 139 defines the test conditions for this test case.

Table 139 – HardwareRevision

-CR123-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0120
Name	TCD_DLIC_DEFP_HARDREV
Purpose (short)	Test of parameter HardwareRevision
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.12
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter HardwareRevision
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter HardwareRevision (Index 0x0016)
Test parameter	HardwareRevision of the EUT provided by the manufacturer
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information
Test not passed (examples)	Any of the evaluations fails
Report	HardwareRevision response: <negative/positive> <ok | nok> HardwareRevision UTF8 coding: <yes/no> <ok | nok> HardwareRevision length: <value> <ok | nok> HardwareRevision adequate: <yes/no> <ok | nok>

6.10.18 Firmware Revision

Table 140 defines the test conditions for this test case.

Table 140 – Firmware Revision

-CR123-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0121
Name	TCD_DLIC_DEFP_FIRMREV
Purpose (short)	Behavior of parameter Firmware Revision
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.13
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter FirmwareRevision
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter FirmwareRevision (Index 0x0017)
Test parameter	FirmwareRevision of the EUT provided by the manufacturer
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response 2) Check coding of parameter 3) Check parameter length
Test passed	No negative response, and Parameter coded in UTF8, and Parameter length ≤ 64 octets, and Contents matches vendor / Device specific information
Test not passed (examples)	Any of the evaluations fails
Report	FirmwareRevision response: <negative/positive> <ok | nok> FirmwareRevision UTF8 coding: <yes/no> <ok | nok> FirmwareRevision length: <value> <ok | nok> FirmwareRevision adequate: <yes/no> <ok | nok>

6.10.19 Application Specific Tag – valid

Table 141 defines the test conditions for this test case.

Table 141 – Application Specific Tag – valid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0122
Name	TCD_DLIC_DEFP_TAGVALID
Purpose (short)	Behavior of parameter ApplicationSpecificTag – valid strings
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.16, 10.3.4, and 10.3.5, Table 98, rule 6
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct behavior for write and read access to parameter ApplicationSpecificTag
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write a random text string with length <specified fixed length> to parameter ApplicationSpecificTag (Index 0x0018) b) Read parameter ApplicationSpecificTag (Index 0x0018) c) Power OFF/ON of the Device d) Read parameter ApplicationSpecificTag (Index 0x0018)
Test parameter	Random text string with <specified fixed length> = manufacturer dependent (minimum 16 octets, maximum 32 octets).
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response of Read and Write access 2) Compare contents and length of reading in step b) to written string in step a) 3) Compare contents and length of reading in step b) and step d)
Test passed	No negative responses, and String lengths in 2) = string lengths in 3) Contents in 2) = Contents in 3)
Test not passed (examples)	Negative response or comparison fails
Report	ApplicationSpecificTag string length: <ok | nok> ApplicationSpecificTag content: <ok | nok>

6.10.20 Application Specific Tag – invalid

Table 142 defines the test conditions for this test case.

Table 142 – Application Specific Tag – invalid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0123
Name	TCD_DLIC_DEFP_TAGINVALID
Purpose (short)	Behavior of parameter ApplicationSpecificTag – invalid string length
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.14
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct behavior for Write and Read access with invalid string length to parameter ApplicationSpecificTag.
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter ApplicationSpecificTag (Index 0x0018) b) Write a random string with length <specified fixed length + 1> to parameter Application Specific Tag (Index 0x0018) c) Read parameter ApplicationSpecificTag (Index 0x0018)
Test parameter	Random text string with <specified fixed length> = manufacturer dependent (minimum 16 octets, maximum 32 octets)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read responses 2) Check Write response 3) Compare contents and length of reading in step a) and step c)
Test passed	No negative Read responses, and Negative Write response = VAL_LENOVRRUN (0x8033), and Values in 3) are matching
Test not passed (examples)	No response
Report	ApplicationSpecificTag negative Write response: <yes/no> <ok | nok> ApplicationSpecificTag contents match: <yes/no> <ok | nok>

6.10.21 Error Count

Table 143 defines the test conditions for this test case.

Table 143 – Error Count

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0124
Name	TCD_DLIC_DEFP_ERRCOUNT
Purpose (short)	Behavior of parameter ErrorCount
Equipment under test (EUT)	Device with adequate reference in IODD (stimulation of ErrorCount possible)
Test case version	1.3
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.19
Configuration / setup	Device-Tester-Unit; user interaction for stimulation is required

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter ErrorCount
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter ErrorCount (Index 0x0020) b) Stimulate error within Device specific technology (registered for ErrorCount) c) Read parameter ErrorCount (Index 0x0020) d) Power OFF/ON of the Device *;reset of ErrorCount* e) Read parameter ErrorCount (Index 0x0020)
Test parameter	Manufacturer defined stimulation of an error
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Read responses 2) Check parameter length 3) Compare values from step a) and step c) if stimulation is possible 4) Check value of step e)
Test passed	No negative Read responses, and Parameter length = 2 octets, and ErrorCount in c) = ErrorCount in a) + 1, and ErrorCount in e) = 0x0000
Test not passed (examples)	Any of the evaluations fails
Report	ErrorCount: <length> <ok | nok> ErrorCount: <increment> <ok | nok> ErrorCount: <value> <ok | nok> No stimulation possible: <Exception>

6.10.22 Device Status

Table 144 defines the test conditions for this test case.

Table 144 – DeviceStatus

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0128
Name	TCD_DLIC_DEFP_DEVSTAT
Purpose (short)	Behavior of parameter DeviceStatus
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.20
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter DeviceStatus
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter DeviceStatus (Index 0x0024)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response 2) Check parameter length 3) Check parameter value
Test passed	No negative response, and Parameter length = 1 octet, and "0" ≤ value ≤ 4
Test not passed (examples)	No response
Report	DeviceStatus response: <positive/negative> <ok | nok> DeviceStatus length: <value> <ok | nok> DeviceStatus: <value> <ok | nok>

6.10.23 Detailed Device Status – complete object

Table 145 – **Detailed Device Status – complete object** defines the test conditions for this test case.

Table 145 – Detailed Device Status – complete object

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0129
Name	TCD_DLIC_DEFP_DETAILDEVSTAT
Purpose (short)	Behavior of entire parameter DetailedDeviceStatus
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.17 and [4]
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test of the correct length of parameter DetailedDeviceStatus
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter DetailedDeviceStatus (Index 0x0025)
Test parameter	<record item count>
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check response 2) Check parameter length 3) Calculate <record item count> = parameter length / 3
Test passed	No negative response, and Parameter length ≤ 64 x 3 octets, and 1 ≤ record item count is an integer value ≤ 64
Test not passed (examples)	No response
Report	DetailedDeviceStatus response: <positive/negative> <ok | nok> DetailedDeviceStatus length: <value> <ok | nok> DetailedDeviceStatus: <value> <ok | nok>

6.10.24

6.10.25 Detailed Device Status – Event active

Table 146 defines the test conditions for this test case.

Table 146 – Detailed Device Status – Event active

-CR066-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0131
Name	TCD_DLIC_DEFP_DETAILDEVSTATACTIVE
Purpose (short)	Record contents in parameter DetailedDeviceStatus with active Event
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.2
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.21
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test of correct entry of active Events in parameter DetailedDeviceStatus
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Stimulate an incident in Device's specific technology causing an Event , using config7 (error or warning) b) Read record of parameter DetailedDeviceStatus c) Parse result from beginning to end in steps of 3 octets and store value d) Power OFF/ON of the Device e) Read record of parameter DetailedDeviceStatus f) Parse result from beginning to end in steps of 3 octets and store value
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check value at each procedure step c) 2) Check value at each procedure step f)
Test passed	Evaluation 1) One of the parsed Events equals the initiated Event of Step a) Evaluation 2) Check parsed values of Detailed Device Status against the active Events
Test not passed (examples)	– Incorrect values or no response – an initiated event does not appear in the Detailed Device Status
Report	DetailedDeviceStatus Event active: <ok | nok>

6.10.26 Process Data Input

Table 147 defines the test conditions for this test case.

Table 147 – Process Data Input

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0132
Name	TCD_DLIC_DEFP_PDIN
Purpose (short)	Behavior of parameter ProcessDataInput
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.18
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test the correct contents and coding of parameter ProcessDataInput
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read parameter ProcessDataInput (Index 0x0028)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check Read response 2) Check parameter length 3) Check parameter contents	
Test passed	No negative response, and Parameter length matches parameter ProcessDataIn (DPP1, address 0x05), and Unused bits = "0"	
Test not passed (examples)	No response	
Report	ProcessDataInput response: <positive/negative> ProcessDataInput length: <value> ProcessDataInput unused bits = "0": <yes/no>	<ok | nok> <ok | nok> <ok | nok>

6.10.27 Process Data Output

Table 148 defines the test conditions for this test case.

Table 148 – Process Data Output

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0133
Name	TCD_DLIC_DEFP_PDOUT
Purpose (short)	Behavior of parameter ProcessDataOutput
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.19
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test of the correct contents and coding of parameter ProcessDataOutput
Precondition	DTU: Communication EUT: OPERATE
Procedure	Read parameter ProcessDataOutput (Index 0x0029)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check Read response 2) Check parameter length 3) Check parameter contents	
Test passed	No negative response, and Parameter length matches parameter ProcessDataOut (DPP1, address 0x06), and Unused bits = "0"	
Test not passed (examples)	No response	
Report	ProcessDataOutput response: <positive/negative> ProcessDataOutput length: <value> ProcessDataOutput unused bits = "0": <yes/no>	<ok | nok> <ok | nok> <ok | nok>

6.10.28 Offset Time – valid

Table 149 defines the test conditions for this test case.

Table 149 – Offset Time – valid

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0134
Name	TCD_DLIC_DEFP_OFFTIMEVALID
Purpose (short)	Behavior of OffsetTime with valid values
Equipment under test (EUT)	Device with adequate reference in IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2.20, Table B.11
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	The test verifies whether the value range for OffsetTime is correct.
Precondition	DTU: Communication EUT: OPERATE at maximum MasterCycleTime
Procedure	a) Write OffsetTime with "0" for time base and for multiplier (Index 0x0030) b) Read OffsetTime c) Repeat step a) and step b) with multiplier = "63" d) Repeat step a) to step c) with time base = "1" and time base = "2"
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check Write response in a) and Read response in b) 2) Compare written value in a) with response value in b)
Test passed	No negative responses, and Values in 2) are matching
Test not passed (examples)	No response
Report	For all three settings: OffsetTime valid response: <positive/negative> <ok | nok> OffsetTime valid matching values: <yes/no> <ok | nok>

6.10.29 Profile Parameter – Read access

Table 150 defines the test conditions for this test case.

Table 150 – Profile Parameter – Read access

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0136
Name	TCD_DLIC_DEFP_PROFILEPARREAD
Purpose (short)	Behavior of unexpected profile parameter Read access
Equipment under test (EUT)	Device supporting no profile: IODD attribute "features/ProfileCharacteristic" is not implemented within the IODD or contains no entries
Test case version	1.2
Category / type	Device application test: test to fail
Specification (clause)	[6], see Annex B.2.21
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test of Read access to Profile specific parameters
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Read Profile parameters from Index 0x0031 to 0x003F
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check each Read response
Test passed	Negative response = IDX_NOTAVAIL (0x8011) upon each profile parameter
Test not passed (examples)	No negative response on profile parameters, or Negative response ≠ IDX_NOTAVAIL (0x8011)
Report	For all Read responses: Profile parameter response: <negative/positive> <ok | nok>

6.10.30 Profile Parameter – Write access

Table 151 defines the test conditions for this test case.

Table 151 – Profile Parameter – Write access

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0137
Name	TCD_DLIC_DEFP_PROFILEPARWRITE
Purpose (short)	Behavior of unexpected profile parameter Write access
Equipment under test (EUT)	Device supporting no profile: IODD attribute "features/ProfileCharacteristic" is not implemented within the IODD or contains no entries
Test case version	1.2
Category / type	Device application test: test to fail
Specification (clause)	[6], see Annex B.2.21
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Test of Write access to Profile specific parameters
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) Write profile parameters from index 0x0031 to 0x003F with <values>
Test parameter	<values>: 0x0000
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check each Write response
Test passed	Negative response = IDX_NOTAVAIL (0x8011) upon each profile parameter
Test not passed (examples)	No negative response on profile parameters, or Negative response ≠ IDX_NOTAVAIL (0x8011)
Report	For all Write responses: Profile Parameter used response: <negative/positive> <ok | nok>

6.10.31 Write access – Read only parameter

Table 152 defines the test conditions for this test case.

Table 152 – Write access – Read only parameter

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0140
Name	TCD_DLIC_DEFP_WRITETOREADONLY
Purpose (short)	Write access to Read only standard parameter
Equipment under test (EUT)	Device with ISDU support and with adequate parameter reference in IODD
Test case version	1.3
Category / type	Device application test: test to fail
Specification (clause)	[6], see Annex B.2 and Annex C.2.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test that Write access to a Read only standard parameter shows the correct response behavior
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) First standard parameter marked "Read Only" b) Read standard parameter *;provides parameter length* c) Write <value> to standard parameter d) Repeat at b) with next standard parameter marked "Read Only"
Test parameter	<value>: any standard parameter in IODD marked as "Read Only"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Write responses
Test passed	Negative response = IDX_NOT_ACCESSIBLE (0x8023)
Test not passed (examples)	No response
Report	Upon all Write accesses to Read Only parameters: Parameter Read Only response: <ErrorType> <ok | nok>

6.10.32 Write access – Length too short

Table 153 defines the test conditions for this test case.

Table 153 – Write access – Length too short

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0141
Name	TCD_DLIC_DEFP_WRITETOOSHORT
Purpose (short)	Write access with invalid length (too short) to writable standard parameter
Equipment under test (EUT)	Device with ISDU support and writeable standard parameters are referenced within the IODD
Test case version	1.1
Category / type	Device application test: test to fail
Specification (clause)	[6], see Annex B.2 and Annex C.2.13
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test that Write access to a writable standard parameter with a too short parameter length shows the correct response behavior
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) First standard parameter with no data type StringT and length > 1 octet b) Read standard parameter *;provides parameter length* c) Write <value> with reduced length to standard parameter d) Repeat at b) with next standard parameter with no StringT and length > 1 octet
Test parameter	<value>: any writable standard parameter in IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Write responses
Test passed	Negative response = VAL_LENUNDRUN (0x8034)
Test not passed (examples)	No response or incorrect ErrorType
Report	Upon all Write accesses with too short length to writable parameters: Parameter Write response: <ErrorType> <ok | nok>

6.10.33 Write access – Length too long

Table 154 defines the test conditions for this test case.

Table 154 – Write access – Length too long

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0142
Name	TCD_DLIC_DEFP_WRITETOOLONG
Purpose (short)	Write access with invalid length (too long) to writable standard parameter
Equipment under test (EUT)	Device with ISDU support and writeable standard parameters are referenced within the IODD
Test case version	1.1
Category / type	Device application test: test to pass
Specification (clause)	[6], see Annex B.2, and Annex C.2.8
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test that Write access to a writable standard parameter with a too long parameter length shows the correct response behavior
Precondition	DTU: Communication EUT: OPERATE
Procedure	a) First standard parameter with no data type StringT and length > 1 octet b) Read standard parameter *;provides parameter length* c) Write <value> with extended length to standard parameter d) Repeat at b) with next standard parameter with no StringT and length > 1 octet
Test parameter	<value>: any writable standard parameter in IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check Write responses
Test passed	Negative response = VAL_LENOVRRUN (0x8033)
Test not passed (examples)	No response or incorrect ErrorType
Report	Upon all Write accesses with too long length to writeable parameters: Parameter Write response: <ErrorType> <ok | nok>

6.11 Block parameter

6.11.1 General

The manufacturer/vendor of a Device shall provide information about a possible Block Parameter set enabling the performance of the following tests. This Block Parameter set shall comply with the requirements of the test cases 6.11.2 through 6.11.7.

6.11.2 Block parameter – Download

Table 155 defines the test conditions for this test case.

Table 155 – Block parameter – Download

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0143
Name	TCD_DSBP_APPL_BPDOWNLOAD
Purpose (short)	Test of Block parameter Download
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.1
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Test of Block parameter Download
Precondition	DTU: SDCI communication EUT: No Block parameterization is active
Procedure	a) Write SystemCommand "ParamDownloadStart" b) Write Block parameters defined in field Test parameter c) Write SystemCommand "ParamDownloadEnd"
Test parameter	Parameter set defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check after step a) that no ErrorType has been received 2) Check during step b) that no ErrorType has been received 3) Check after step c) that no ErrorType has been received	
Test passed	All evaluation steps with positive result	
Test not passed (examples)	No response or any evaluation negative	
Report	Write "ParamDownloadStart" Write Block parameter Write "ParamDownloadEnd"	<ok | nok> <ok | nok> <ok | nok>

6.11.3 Block parameter – Break by command

Table 156 defines the test conditions for this test case.

Table 156 – Block parameter – Break by command

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0144
Name	TCD_DSBP_APPL_BPBREAKCMD
Purpose (short)	Test break of Block parameter transfer per command
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.1
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall discard any change of parameters when receiving a SystemCommand "ParamDownloadBreak"
Precondition	DTU: SDCI communication EUT: no Block parameterization is active
Procedure	a) Read one parameter (see field Test parameter) and buffer it as <value1> b) Write SystemCommand "ParamDownloadStart" c) Write parameter from step a) with <value2> ≠ <value1> d) Write SystemCommand "ParamDownloadBreak" e) Read parameter from step a) f) Write SystemCommand "ParamDownloadEnd" g Read parameter from step a)
Test parameter	Parameter set defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that no ErrorType has been received 2) Check after step d) that no ErrorType has been received 3) Check after step e) that received value matches <value1> 4) Check after step f) that ErrorType 0x8036 has been received 5) Check after step g) that received value matches <value1>
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write "ParamDownloadBreak" <ok | nok> Rollback to previous value after break <ok | nok> Write "ParamDownloadEnd" <ok | nok> Non-volatile storage after ParamDownloadEnd <ok | nok>

6.11.4 Block parameter – Break by reset

Table 157 defines the test conditions for this test case.

Table 157 – Block parameter – Break by reset

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0145
Name	TCD_DSBP_APPL_BPBREAKRESET
Purpose (short)	Test break of Block parameter transfer per reset
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.1
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall discard any change of parameters if a reset occurs during parameterization
Precondition	DTU: SDCI communication EUT: No Block parametrization is active
Procedure	a) Read one Parameter (see field Test parameter) and buffer it as <value1> b) Write SystemCommand "ParamDownloadStart" c) Write parameter from step a) with <value2> ≠ <value1> d) Reset the Device either (descending priority based on availability) - using MasterCommand "Fallback" or - using SystemCommand "DeviceReset" or - using power off/on cycle e) Set Device into SDCI communication f) Read Parameter from step a) g) Write SystemCommand "ParamDownloadEnd" h) Read Parameter from step a)
Test parameter	Parameter set defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that no ErrorType has been received 2) Check after step e) that Device communication is active 3) Check after step f) that received value matches <value1> 4) Check after step g) that ErrorType 0x8036 has been received 5) Check after step h) that received value matches <value1>
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write "ParamDownloadBreak" <ok | nok> SDCI restart <ok | nok> Rollback to previous value after communication restart <ok | nok> Write "ParamDownloadEnd" <ok | nok> Non-volatile storage after ParamDownloadEnd <ok | nok>

6.11.5 Block parameter – Illegal parameter write

Table 158 defines the test conditions for this test case.

Table 158 – Block parameter – Illegal parameter write

-CR029-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0146
Name	TCD_DSBP_APPL_BPBREAKILLPARAM
Purpose (short)	Test break of Block parameter transfer by illegal parameter
Equipment under test (EUT)	Device with option “blockParameter” indicated in IODD
Test case version	1.2
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall discard any change of parameters if an illegal parameter occurs during parameterization
Precondition	DTU: SDCI communication EUT: no Block parametrization is active
Procedure	a) Read one parameter (see Test parameter) and buffer it as <value1> b) Write SystemCommand "ParamDownloadStart" c) Write parameter from step a) with <value2> ≠ <value1> d) Write another parameter using an illegal structure (incorrect data length) e) Write SystemCommand "ParamDownloadEnd" f) Read parameter from step a)
Test parameter	Parameter set defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that no ErrorType has been received 2) Check after step d) that ErrorType 0x8033 or 0x8034 has been received 3) Check after step e) that ErrorType 0x8041 has been received 4) Check after step f) that received parameter matches <value1>
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write "ParamDownloadStart" <ok | nok> Device behavior upon illegal structure <ok | nok> Write "ParamDownloadEnd" <ok | nok> Rollback to previous value after inconsistent parameter block <ok | nok>

6.11.6 Block parameter – Break by double download

Table 159 defines the test conditions for this test case.

Table 159 – Block parameter – Break by double download

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0147
Name	TCD_DSBP_APPL_BPBREAK2DOWNLOADS
Purpose (short)	Test break of Block Parameter transfer by double SystemCommand "ParamDownloadStart"
Equipment under test (EUT)	Device with option “blockParameter” indicated in IODD
Test case version	1.1
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall discard any change of parameters if Block parameterization is restarted via a new "ParamDownloadStart" during a running Block parameter transfer
Precondition	DTU: SDCI communication EUT: no block parametrization is active
Procedure	a) Read one parameter (see field Test parameter) and buffer it as <value1> b) Write SystemCommand "ParamDownloadStart" c) Write parameter from step a) with <value2> ≠ <value1> d) Write SystemCommand "ParamDownloadStart" e) Write all parameters in field Test parameter, except those used in step a) f) Write SystemCommand "ParamDownloadEnd" g Read parameter from step a)
Test parameter	Parameter set defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that no ErrorType has been received 2) Check after step d) that no ErrorType has been received 3) Check during step e) that no ErrorType has been received 4) Check after step f) that no ErrorType has been received 5) Check after step g) that received value does not match <value1>
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write "ParamDownloadStart" <ok | nok> Write second "ParamDownloadStart" while block is unfinished <ok | nok> Write Block parameter <ok | nok> Write "ParamDownloadEnd" <ok | nok> Rollback to previous values via second “ParamDownloadStart” <ok | nok>

6.11.7 Block parameter – local locking

Table 160 defines the test conditions for this test case.

Table 160 – Block parameter – local locking

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0148
Name	TCD_DSBP_APPL_BPBREAKLOCALLOCK
Purpose (short)	Test locking of local parametrization during Block parameter transfer
Equipment under test (EUT)	Device with option “blockParameter” indicated in IODD, and local parameterization capability (on-board)
Test case version	1.1
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Local access and change of parameterization shall be locked during Block parameterization
Precondition	DTU: SDCI communication EUT: no block parametrization is active
Procedure	a) Read one parameter from A) and buffer it as <value1> b) Read parameter B) and buffer it as <value2> c) Write SystemCommand "ParamDownloadStart" d) Write parameter from step a) with <value2> ≠ <value1> e) Try to change parameter B) via local parameterization f) Write SystemCommand "ParamDownloadEnd" g) Read parameter from step a) h) Read parameter B)
Test parameter	A) Parameter set defined in 6.7.1.1 B) Parameter in A) that can be changed by local parametrization (on-board) and via SDCI communication
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step c) that no ErrorType has been received 2) Check during step e) that the Device denies the trial of local parameterization 3) Check after step f) that no ErrorType has been received 4) Check after step g) that received value does not match <value1> 5) Check after step h) that received value matches <value2>
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write "ParamDownloadStart" <ok | nok> Inhibited local parametrization <ok | nok> Write "ParamDownloadEnd" <ok | nok> Non-volatile storage after "ParamDownloadEnd" <ok | nok> Rejection of local changes <ok | nok>

6.11.8 ParameterManager – Unexpected commands in Idle

Table 161 defines the test conditions for this test case.

Table 161 – ParameterManager – Unexpected commands in Idle

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0324
Name	TCD_DSBP_APPL_UNEXPECTEDINIDLE
Purpose (short)	Test of unexpected commands in state Idle of ParameterManager
Equipment under test (EUT)	Device
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall issue the correct ErrorType after receiving an unexpected block ending command while ParameterManager is in state Idle.
Precondition	DTU: SDCI communication EUT: No block parametrization is active
Procedure	Write SystemCommand "ParamUploadEnd" Write SystemCommand "ParamBreak" Write SystemCommand "ParamDownloadEnd"
Test parameter	"blockParameter" <true / false> indicated in IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	After steps a) to c), check received ErrorType

Test passed	ErrorType = 0x8036 if blockParameter <true>, OR
	ErrorType = 0x8035 if blockParameter <false>

Test not passed (examples)	No response or incorrect ErrorTypes
Report	Unexpected actions in ParameterManager state Idle <ok | nok>

6.11.9 ParameterManager – Write request during Upload

Table 162 defines the test conditions for this test case.

Table 162 – ParameterManager – Write request during Upload

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0325
Name	TCD_DSBP_APPL_WRITEINUPLOAD
Purpose (short)	Test of reaction on write accesses during an active **B**lock Upload
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Device shall not accept any write accesses to parameters except via System-Commands during an active Block Upload. The SystemCommand is already tested in other test cases.
Precondition	DTU: SDCI communication EUT: No block parametrization is active
Procedure	a) Write SystemCommand "ParamUploadStart" b) Write any parameter which is not excluded from DataStorage c) Write SystemCommand "ParamUploadEnd" d) Write any parameter which is not excluded from DataStorage
Test parameter	One parameter of the parameter sets defined in 6.7.1.1
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check after step b) that ErrorType 0x8022 has been received 2) Check after step d) that no ErrorType has been received
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Write restriction during Block parameter Upload <ok | nok>

6.11.10 ParameterManager – Read requests during Download

Table 163 defines the test conditions for this test case.

Table 163 – ParameterManager – Read requests during Download

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0326
Name	TCD_DSBP_APPL_READINDOWNLOAD
Purpose (short)	Test of reaction on read accesses during active Download
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device shall not accept any Read accesses to parameters during an active Block Download.
Precondition	DTU: SDCI communication EUT: No Block parametrization is active
Procedure	a) Write SystemCommand "ParamDownloadStart" b) Read Test parameter c) Read DataStorageIndex, Index 3, SubIndex 0 d) Write SystemCommand "ParamDownloadEnd" e) Read Test parameter
Test parameter	One parameter of the parameter sets defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that ErrorType 0x8022 has been received 2) Check after step c) that no ErrorType has been received 3) Check after step e) that no ErrorType has been received
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Read restriction during Block parameter Download <ok | nok>

6.11.11 ParameterManager – Unexpected commands during Upload

Table 164 defines the test conditions for this test case.

Table 164 – ParameterManager – Unexpected commands during Upload

Identification (ID)	SDCI_TC_0327
Name	TCD_DSBP_APPL_UNEXPINUPLOAD
Purpose (short)	Test of reaction on unexpected SystemCommands during active Upload
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	During an active Block Upload, a Device shall react on unexpected SystemCommands, abort the ongoing process and switch to the requested state.
Precondition	DTU: SDCI communication EUT: - No Event pending or active - DataStorage of the Device is inactive and DS_UPLOAD_FLAG is "0" - No Block parametrization is active
Procedure	a) Write SystemCommand "ParamUploadStart" b) Write Test parameter A) c) Write SystemCommand "ParamUploadStart" d) Write Test parameter A) e) Write SystemCommand "ParamDownloadEnd" f) Write Test parameter A) If "dataStorage" = <true> g) Write SystemCommand "ParamUploadStart" h) Write SystemCommand "ParamDownloadStore" i) Wait for Event (max 1s) and read DataStorage Index j) Write DS_Command "DS_UploadEnd"
Test parameter	A) One parameter of the parameter sets defined in 6.7.1.1 B) dataStorage <true / false> indicated in IODD
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check after step b) that ErrorType 0x8022 has been received 2) Check after step d) that ErrorType 0x8022 has been received 3) Check after step f) that no ErrorType has been received If "dataStorage" = <true> 4) Check after step i) that Event with code DS_UPLOAD_REQ has been received and DS_UPLOADREQ = "1"
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Reaction on unexpected commands in Upload <ok | nok>

6.11.12 ParameterManager – Unexpected switches Upload/Download

Table 165 defines the test conditions for this test case.

Table 165 – ParameterManager – Unexpected switches Upload/Download

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0328
Name	TCD_DSBP_APPL_SWITCHSTATES
Purpose (short)	Test of reaction on unexpected switches between Upload and Download
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Any Upload or Download shall be interrupted by the start of the opposite action.
Precondition	DTU: SDCI communication EUT: No Block parameterization is active
Procedure	a) Write SystemCommand "ParamUploadStart" b) Write Test parameter c) Write SystemCommand "ParamDownloadStart" d) Read Test parameter e) Write SystemCommand "ParamUploadStart" f) Write Test parameter g) Write SystemCommand "ParamUploadEnd"
Test parameter	One parameter of the parameter sets defined in 6.7.1.1
Post condition	-
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check after step b) that ErrorType 0x8022 has been received 2) Check after step d) that ErrorType 0x8022 has been received 3) Check after step f) that ErrorType 0x8022 has been received
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Switching between Upload and Download <ok | nok>

6.11.13 ParameterManager – Upload interrupted by reset

Table 166 defines the test conditions for this test case.

Table 166 – ParameterManager – Upload interrupted by reset

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0329
Name	TCD_DSBP_APPL_UPBREAKRESET
Purpose (short)	Test of Block Parameter Upload transfer interrupted per reset
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Any Upload can be interrupted by a communication reset and Block parameterization shall be aborted in this case.
Precondition	DTU: SDCI communication EUT: No block parametrization is active
Procedure	a) Write SystemCommand "ParamUploadStart" b) Write Test parameter c) Reset the Device either (descending priority based on availability) - using MasterCommand "Fallback" or - using SystemCommand "DeviceReset" or - using power off/on cycle reset d) Set Device to SDCI communication e) Write Test parameter
Test parameter	One parameter of the parameter sets defined in 6.7.1.1
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check after step b) that ErrorType 0x8022 has been received 2) Check after step e) that no ErrorType has been received
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Communication interruption during Upload <ok | nok>

6.11.14 ParameterManager – UploadEnd during Download

Table 167 defines the test conditions for this test case.

Table 167 – ParameterManager – UploadEnd during Download

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0330
Name	TCD_DSBP_APPL_DNENDBYUPLOAD
Purpose (short)	Reaction on Upload commands while in Download, discarding written parameter
Equipment under test (EUT)	Device with option "blockParameter" indicated in IODD
Test case version	1.0
Category / type	Device Block parameter test: test to pass
Specification (clause)	[6], 10.3.2, 10.3.5, 10.6.11, Table 97, Table 98, Table B.9
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Parameter values of any Download shall be discarded when receiving an Upload-Start or UploadEnd command.
Precondition	DTU: SDCI communication EUT: No Block parameterization is active
Procedure	a) Read Test parameter A) b) Write SystemCommand “ParamDownloadStart” c) Write Test parameter value A) or B), whichever is different from step a) d) Write SystemCommand “ParamUploadStart” e) Read parameter from c) f) Write SystemCommand “ParamDownloadStart” g) Write Test parameter value A) or B), whichever is different from step a) h) Write SystemCommand “ParamUploadEnd” i) Read parameter from c)
Test parameter	A) One parameter of the parameter sets defined in 6.7.1.1 with value of set 1 B) Same as parameter in A) with value of set 2
Post condition	-
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check after step e) that read value equals original value of step a) 2) Check after step i) that read value equals original value of step a)
Test passed	All evaluation steps with positive result
Test not passed (examples)	No response or any evaluation negative
Report	Download break by any Upload command <ok | nok>

6.12 Test report summary of the Device protocol tests

The template is defined by the Device-Tester. The complete test report shall present at least the information of the report fields of the test cases for the Device protocol tests.

7 IODD-related tests

7.1 Overview

The IODD test focuses on four major issues. The first issue is the test of a Device's IODD file with the help of a so-called IODD checker tool. IO-Link defines the parsers such an IODD checker tool shall use for IODD schema consistency checks. It also defines a set of business rules for the IODD check (see 7.2.1).

The second issue focuses on the IODD interpreter tool. This test shall ensure, that the "Port and Device Configuration Tool" (PDCT) of the Master is able to provide all the IODD definitions in the correct manner (see 7.2.2).

The third issue is the test whether the parameters defined in the IODD are accessible within the Device (parameter verification test). These test cases are specified in 7.3.

The fourth issue is the test of complex system functions such as diverse reset possibilities that are indicated within the IODD. These test cases are specified in 7.4.

For all these tests, the Master shall be configured with inspection level "type compatible".

7.2 Schema test via an IODD checker tool and IODD interpreter tests

7.2.1 IODD Checker

The organization referenced in Annex D makes available an IODD checker tool ("Checker") for free download from its web server. It is mandatory for each and every IODD associated with a Device to pass the test with this Checker. The Device's manufacturer declaration shall state the successful result of the test.

The requirements for IODD testing had been specified in previous versions of this document but was shifted to the IODD specification [3] for practical reasons.

7.2.2 IODD interpreter tool

A Master-Tester-System shall provide a so-called fictive IODD with critical constellations of parameters, which are supported by the Master-Tester-Unit (MTU) playing the role of a Device. The IODD interpreter tools associated or related to a particular Master can be tested with the help of this IODD. Clause 4.5 describes how these tests can be performed.

7.3 Parameter verification test

7.3.1 IODD identification

Table 168 defines the test conditions for this test case.

Table 168 – IODD identification

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0149
Name	TCD_IODD_PARV_IDENT
Purpose (short)	Device matches the associated IODD
Equipment under test (EUT)	Device and associated IODD
Test case version	1.0
Category / type	IODD parameter verification test: test to pass
Specification (clause)	[3], 7.4 and 7.4.1; [6], B.1.8, B.1.9 and B.2.11
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Verify that Device matches the associated IODD. DTU reads Device parameters VendorID, DeviceID, and ProductID, and compares with IODD node descriptions. For further tests, read of DeviceID and ProductID are sufficient for an IODD association check
Precondition	DTU: SDCI communication EUT: Device is communicating, associated IODD available in machine readable form
Procedure	a) Read VendorID, DeviceID, ProductID from DPP1 of the Device b) Read corresponding entries from IODD *;see Test parameter* c) Read DeviceVariantCollection from IODD *;see Test parameter*
Test parameter	IODD: DeviceIdentity/@vendorId, DeviceIdentity/@deviceId, DeviceVariant/@productId
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check if IDs match the entries in IODD's Device identity after b) 2) Check if ProductID is listed in the DeviceVariantCollection of the IODD after c)
Test passed	All IDs from Device found in IODD
Test not passed (examples)	Any of the ID from Device not found in IODD (not matching)
Report	VendorID: <value> <ok | nok> DeviceID: <value> <ok | nok> ProductID: <value> <ok | nok> ProductID listed in DeviceVariantCollection: <yes/no> <ok | nok>

7.3.2 IODD communication parameter verification

Table 169 defines the test conditions for this test case.

Table 169 – IODD communication parameter verification

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0150
Name	TCD_IODD_PARV_COMPROFILE
Purpose (short)	Device's communication parameters match corresponding values in IODD
Equipment under test (EUT)	Device and associated IODD
Test case version	1.0
Category / type	IODD parameter verification test: test to pass
Specification (clause)	[3], 7.6; [6], B.1.3, B.1.5 and B.1.6
Configuration / setup	Device-Tester-Unit
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Verify that Device's communication parameters match values within the associated IODD. DTU reads Device parameters MinCycleTime, RevisionID, and ProcessDataIn and compares with IODD node descriptions.
Precondition	DTU: SDCI communication EUT: Device is communicating; associated IODD available in machine readable form VendorID and ProductID of the Device match values in IODD
Procedure	a) Read MinCycleTime, RevisionID, ProcessDataIn (address 0x05, bit 6 – SIO supported) from DPP1 of the Device b) Read corresponding entries from IODD *;see Test parameter*
Test parameter	IODD: CommNetworkProfile/@iolinkRevision, TransportLayers/PhysicalLayer/@minCycleTime, PhysicalLayer/@sioSupported
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Compare values in DPP1 with corresponding entries in IODD after b)
Test passed	Values are matching
Test not passed (examples)	Values do not match
Report	MinCycleTime: <value> <ok | nok> RevisionID: <value> <ok | nok> SIO supported: <yes/no> <ok | nok>

7.3.3 IODD parameter read verification

Table 170 defines the test conditions for this test case.

Table 170 – IODD parameter read verification

-CR018- -CR031- -CR034-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0151
Name	TCD_IODD_PARV_READVERIFY
Purpose (short)	Verify access rights, structures, and data contents of Read parameters
Equipment under test (EUT)	Device and associated IODD
Test case version	1.1
Category / type	IODD parameter verification test: test to pass
Specification (clause)	[3], 7.5.4; [6], Table C.1, C.2.2 to C.2.19, F.2.2 to F.2.9, F.3.2, F.3.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Verify that all parameters in the IODD with Read access can be read from the Device, have a correct structure and valid data content. The length of an acquired parameter is used as measure for its structure since the lengths of parameters depend on their data. The validity of the data content is only checked for parameters of type StringT. The Tests shall be done by access with sub index == 0, and by subindex <>0 if the type of the variable is Record_T, regardless if subindexAccessSupported is true or false.
Precondition	DTU: SDCI communication EUT: Device is communicating; associated IODD available in machine readable form; VendorID and ProductID of the Device match values in IODD
Procedure	a) Get first parameter with Read access from IODD b) Read parameter with Subindex = 0 access c) Read Parameter via all defined SubIndices d) Repeat from b) with next parameter with Read access from IODD until last one
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For each acquired parameter: 1) Check ErrorType after b) 2) b) Read parameter with Subindex = 0 access c) Read Parameter via all defined SubIndices d) Repeat from b) with next parameter with Read access from IODD until last one
Test passed	1) No ErrorType or ErrorType 0x8020, 0x8021, or 0x8022 2) No ErrorType or ErrorType 0x8020, 0x8021, or 0x8022 3) ErrorType 0x8012 or 0x8023 4) Length check OK 5) String content check OK
Test not passed (examples)	Any evaluation failed
Report	For each and every parameter with Read access in the IODD: No Error: <yes/no> <ok | nok> Permitted Error: <ErrorType> <ok | nok> String content check OK (if applicable)

7.3.4 IODD parameter write verification

Table 171 defines the test conditions for this test case.

Table 171 – IODD parameter write verification

-CR019- -CR034- -CR073-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0152
Name	TCD_IODD_PARV_WRITEVERIFY
Purpose (short)	Test verifies index space and value ranges defined within the Device's IODD
Equipment under test (EUT)	Device and associated IODD
Test case version	1.1
Category / type	IODD parameter verification test: test to pass
Specification (clause)	[3], 7.5.4; [6], Table C.1, C.2.2 to C.2.19, F.2.2 to F.2.9, F.3.2, F.3.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device-Tester writes dedicated IODD parameters to the Device. Verify that all parameters in the IODD can be accessed. Test for semantics (device specific technology) is not included. The individual rules for applicable values to be written are defined depending on the data types (for example string, integer, enum), see section "Test parameter". The Tests shall be done by access with sub index == 0, and by subindex <>0 if the type of the variable is Record_T, regardless if subindexAccessSupported is true or false.
Precondition	DTU: SDCI communication EUT: Device is communicating; associated IODD available in machine readable form VendorID and ProductID of the Device match values in IODD
Procedure	a) Store parameter content of all parameters to be tested b) Write parameter content depending on data type using subindex 0 c) Write parameter content parts depending on data type using all defined subindeces. d) Repeat b) and c) with next test parameter e) write content from a) to all tested parameters
Test parameter	Parameters with Read/Write access from IODD For every <value> to be written, the following rules for the data types shall apply: - String: filled with blanks - INT, UINT, FLOAT: maximum value of the permitted range minus one unit - BOOL: true - Time: 2011-04-18 12:00:00 UTC - Timespan: 1 s - Enum (single value): first single value
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For each written parameter: 1) Check ErrorType after b) 2) Check ErrorType after c) in case of subindexAccessSupported=”true” 3) Check ErrorType after c) in case of subindexAccessSupported=”false”
Test passed	1) No ErrorType or ErrorType 0x8020, 0x8021, 0x8022, 0x8030, 0x8031, 0x8032 2) No ErrorType or ErrorType 0x8020, 0x8021, 0x8022, 0x8030, 0x8031, 0x8032 3) ErrorType 0x8012 or 0x8023
Test not passed (examples)	Any evaluation failed

TEST CASE RESULTS	CHECK / REACTION	
Report	For every parameter with Write access in the IODD:	
	No Error: <yes/no> Permitted Error: <ErrorType>	<ok | nok> <ok | nok>

1840

7.3.5 IODD parameter Index/Subindex consistency

Table 172 defines the test conditions for this test case.

Table 172 – IODD parameter Index/Subindex consistency

-CR033-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0157
Name	TCD_IODD_PARV_INDEXCONSISTENT
Purpose (short)	Test the consistency between Indices and Subindices for IODD parameters
Equipment under test (EUT)	Device and associated IODD
Test case version	1.1
Category / type	IODD parameter verification test: test to pass
Specification (clause)	[3], 7.5.4; [6], A.5.4, F.3.2, F.3.3
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Device-Tester tests if reading or writing parameters via Subindex has the same result as reading or writing via Index. Test shall be repeated for all parameters which are: - not dynamic - readable and writeable - subindex access is allowed For each parameter, test shall be repeated for all subindices which are read-able and writeable
Precondition	DTU: SDCI communication EUT: Device is communicating; associated IODD available in machine readable form VendorID and ProductID of the Device match values in IODD This test only for non-dynamic parameters with complex data type and "subindexAccessSupported = true" within the IODD
Procedure	a) Find a valid value <value1> for subindex 0 b) Write <value1> to subindex 0 c) Read subindex 0 and store it to <value2> d) Extract value of subindex n from <value1> and store it to <value3> e) Read subindex n and store it to <value4> f) Find valid value <value5> for subindex n which is different to <value3> g) Write <value5> to subindex n h) Read subindex 0 and store it to <value6> i) Extract value of subindex n from <value6> and store it to <value7> j) Read subindex n and store it to <value8>
Test parameter	For <value1> and <value5> use values from parameter sets defined in 6.7.1.1
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	After step c) compare <value1> with <value2>. Values must be equal. After step e) compare <value3> with <value4>. Values must be equal. After step i) compare <value5> with <value7>. Values must be equal. After step j) compare <value5> with <value8>. Values must be equal.
Test passed	All evaluations are positive
Test not passed (examples)	Any evaluation failed
Report	For every parameter with Subindex access in the IODD: Parameter consistency: <value2/value1> <ok | nok> Parameter consistency: <value2/value3> <ok | nok>

7.4 Functional system tests (IODD indication)

7.4.1 General

The availability of some more complex system functions of a Device is indicated in its IODD. The corresponding functional system tests are specified in this clause.

7.4.2 IODD – functional verification of "Parameter (write) access lock"

Table 173 defines the test conditions for this test case.

Table 173 – IODD – functional verification of "Parameter (write) access lock"

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0156
Name	TCD_IODD_PARV_ACCESSLOCK
Purpose (short)	Test functional behavior of parameter access locking (conditional)
Equipment under test (EUT)	Device with feature "Parameter (write) access" indicated within IODD
Test case version	1.1
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.6.5, B.2.4, C.2.1, Table C.1
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Access to parameter write values is locked if feature "Device Access Lock" is implemented. Only parameters with attribute: accessRights = "rw" or accessRights = "wo" in IODD are tested.
Precondition	DTU: OPERATE EUT: Device is in SDCI communication mode, VendorID and DeviceID match with IODD
Procedure	a) Write value "0x0001" to Index "DeviceAccessLocks" (= parameter access locked) b) Select parameter with Write access within IODD c) Write <value> (see Test parameter) to selected parameter in b) d) Write value "0x0000" to Index "DeviceAccessLocks"
Test parameter	<value> to be defined by manufacturer
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Response is negative and ErrorType = 0x8023 (= access denied) 2) Procedure d) results in a positive response
Test passed	Evaluations 1) and 2) are true
Test not passed (example)	Evaluation 1) or 2) are false
Report	Parameter (from evaluation 1): <index>, <ErrorType> <ok | nok> DeviceAccessLocks (from evaluation 2) <ok | nok> Warning: <yes/no>

7.4.3 IODD – functional verification of "Device reset"

Table 174 defines the test conditions for this test case.

Table 174 – IODD – functional verification of "Device reset"

-CR078- -CR026- -CR020-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0316
Name	TCD_IODD_PARV_DEVICERESET
Purpose (short)	Test functional behavior of SystemCommand "Device reset" (conditional)
Equipment under test (EUT)	Device with feature "Device reset" indicated in IODD
Test case version	1.1
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.7.2, Table 101, B.2.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Parameter and reset behavior shall be as specified after a reset triggered by the SystemCommand "Device reset".
Precondition	DTU: OPERATE EUT: Device is in SDCI communication mode; VendorID and DeviceID match IODD
Procedure	a) Provide input field or configuration option for variable <time> representing the maximum Device time after System command Device Reset to restart communication (in seconds) b) Select parameter from IODD fulfilling the conditions in "Test parameter" c) Read selected parameter from b) and store values in array <parvalue_a> d) Read DID (Device ID) and store value in <didvalue_a> e) Write SystemCommand "Device reset" (128 / 0x80) f) Wait <time> g) Check if a communication startup sequence has been triggered h) Read RID (RevisionID) i) Read DID (DeviceID) and store value in <didvalue_b> j) Check if an Event has been triggered (mode "Event appears k) Read parameter DeviceStatus and store value to <status_b>") if DeviceStatus is marked as implemented in IODD l) Read selected parameter from b) and store values in array <parvalue_b>
Test parameter	Only parameters in IODD are tested - with attribute "accessRights = "rw", and - which are not marked with "excludedFromDataStorage="true" in the IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Procedure g) shows exactly one communication startup sequence 2) Result of procedure h) matches with IODD (CommNetworkProfile: iolinkRevision = "Vx.x") 3) Values of <didvalue_a> match values of <didvalue_b> 4) Procedure j) shows at least one event of mode "Event appears" or value of <status_b> is equal ‘0 (0x00)’ (Skip this step, if DeviceStatus is not marked as implemented in IODD) For each received parameter (from procedure b): 5) Response is positive 6) Values of <parvalue_a> match values of <parvalue_b>
Test passed	All evaluations 1) to 6) are true
Test not passed (example)	Any of the evaluations 1) to 6) is false
Report	Communication from evaluation 1): <value> <ok | nok> RevisionID from evaluation 2): <value> <ok | nok> DeviceID from evaluation 3): <value> <ok | nok> DeviceStatus: <value> (from procedure k), <event> (from procedure j) <ok | nok> For each received parameter (from evaluation 5) and 6)): Parameter: <Index>, <parvalue_a>, <parvalue_b> <ok | nok>

7.4.4 IODD – functional verification of "Application reset"

Table 175 defines the test conditions for this test case.

Table 175 – IODD – functional verification of "Application reset"

-CR020- -CR104- -CR121- -CR040- -CR017- -CR097-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0317
Name	TCD_IODD_PARV_APPLRESET
Purpose (short)	Test functional behavior of SystemCommand "Application reset" (conditional)
Equipment under test (EUT)	Device with feature "Application reset" indicated in IODD
Test case version	1.1
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.7.3, Table 101, B.2.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Parameter and reset behavior shall be as specified after a reset triggered by the SystemCommand "Application reset"
Precondition	DTU: Data storage is disabled EUT: Device is in SDCI communication mode, VendorID and DeviceID match IODD
Procedure	a) Select parameter from IODD fulfilling the conditions in "Test parameter" and excluding parameter ApplicationSpecificTag, FunctionTag, LocationTag b) If supported, read parameter ApplicationSpecificTag, FunctionTag and LoctionTag and store responses in <tagvalue_a> c) Read parameter DID (DeviceID) and RID (RevisionID) and store values in <idvalue_a> d) Write SystemCommand "Application reset" (129 / 0x81) e) Wait 5 s f) Check if a communication startup sequence has been triggered g) Wait for event DS_UPLOAD_REQ h) Check DS_UPLOAD_FLAG in parameter DataStorageIndex i) Read parameter DID (Device ID) and RID (Revision ID) and store values in <idvalue_b> j) If supported, read parameter ApplicationSpecificTag, FunctionTag and LocationTag and store responses in <tagvalue_b> k) Read selected parameter from a) and store values in array <parvalue_b> l) Write 0x04 (DS_DownloadEnd) to Index 3, subindex 1
Test parameter	Only parameters in IODD are tested - with attribute "accessRights = "rw", - with attribute “no defaultValue”
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Procedure f) shows no communication startup sequence 2) Values of <idvalue_a> and <idvalue_b> are equal 3) Values of <tagvalue_a> and <tagvalue_b> are equal 4) Check whether Event DS_UPLOAD_REQ was raised after SystemCommand “Application reset” 5) Value of procedure h) (DS_UPLOAD_FLAG) is equal ‘1’ (DS_UPLOAD_REQ pending) if DS is supported, or is eq ‘0’ if DS is not supported For each received parameter (from procedure a)): 6) Response is positive
Test passed	All evaluations 1) to 6) are true
Test not passed (example)	At least one of the evaluations 1) to 6) is false
Report	Communication (from evaluation 1)) <ok | nok> RevisionID/DeviceID (from evaluation 2)) <ok | nok> Identification (from evaluation 3)) <ok | nok> DataStorage (from evaluation 4)) <ok | nok>

7.4.5 IODD – functional verification of "Restore factory settings"

Table 176 defines the test conditions for this test case.

Table 176 – IODD – functional verification of "Restore factory settings"

-CR071- -CR042- -CR017- -CR028-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0155
Name	TCD_IODD_PARV_FACTORYSETTINGS
Purpose (short)	Test functional behavior of SystemCommand "Restore factory settings" (conditional)
Equipment under test (EUT)	Device with feature "Restore factory settings" indicated in IODD
Test case version	1.2
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.7.4, Table 101, B.2.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Parameter and reset behavior shall be as specified after a reset triggered by the SystemCommand "Restore factory settings"
Precondition	DTU: Data storage is disabled , Inspection level is set to “compatible” EUT: Device is in SDCI communication mode, VendorID and DeviceID match IODD
Procedure	a) Select parameter from IODD, which fulfill the conditions in "Test parameter" b) Read parameter DeviceStatus and store value in <status_a> c) Write SystemCommand " ParamDownloadStore" (5 / 0x05) d) Write SystemCommand "Restore factory setting" (130 / 0x82) e) Check if an Event has been triggered (mode "Event disappears") f) Wait 5 seconds g) Check if a communication startup sequence has been triggered h) Check DS_UPLOAD_FLAG in parameter DataStorageIndex i) Read parameter DID (DeviceID) and RID (RevisionID) and store to <idvalue_b> j) Check if an Event has been triggered (mode "Event appears") k) Read parameter DeviceStatus and store value in <status_b> l) Read selected parameter from a) and store values in array <parvalue_b>
Test parameter	Only parameters in IODD are tested - with attribute "accessRights = "rw", and - which are not "excludedFromDataStorage" and - with attribute: defaultValue
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Procedure g) shows no or exactly one communication startup sequence 2) If <status_a> is ≠ "0 (0x00)", procedure e) shows at least one Event of mode "Event disappears" ,(if a communication startup occured skip this evaluation) 3) Values of <idvalue_b> match default values of IODD RID: (CommNetworkProfile→iolinkRevision="Vx.x") 4) Value of procedure h) (DS_UPLOAD_FLAG) = "0" (no DS_UPLOAD_REQ) 5) Procedure j) shows at least one Event of mode "Event appears" or value of <status_b> = "0 (0x00)" For each received parameter (from procedure a)): 6) Response is positive 7) Values of <parvalue_b> match corresponding assigned "Value" if available in IODD
Test passed	All evaluations 1) to 7) are true (if a communication startup occured skip evaluation 2)
Test not passed (example)	At least one of the evaluations 1) to 7) is false

TEST CASE RESULTS	CHECK / REACTION	
Report	Communication (from evaluation 1))	<ok | nok>
	Diagnosis clear (from evaluation 2))	<ok | nok>
	RevisionID/DeviceID (from evaluation 3))	<ok | nok>
	DataStorage (from evaluation 4))	<ok | nok>
	DeviceStatus (from evaluation 5)): <value>, <event>	<ok | nok>
	For each received parameter (from evaluation 6) and 7)):	
	Parameter: <Index>, <parvalue_a>, <parvalue_b>	<ok | nok>

7.4.6 IODD – functional verification of "Back-to-box"

Table 177 defines the test conditions for this test case.

Table 177 – IODD – functional verification of "Back-to-box"

-CR016- -CR022- -CR023- -CR071- -CR042- -CR017- -CR078-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0318
Name	TCD_IODD_PARV_BACKTOBOX
Purpose (short)	Test functional behavior of SystemCommand "Back-to-box" (conditional)
Equipment under test (EUT)	Device with feature "Back-to-box" indicated in IODD
Test case version	1.1
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.7.5, Table 101, B.2.2
Configuration / setup	Device-Tester-Unit In case of the DeviceId is not equal to the DefaultDeviceId (see System Spec Table 101) , the DefaultIODD must be provided

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Parameter and reset behavior shall be as specified after a reset triggered by the SystemCommand "Back-to-box"
Precondition	DTU: Data storage is disabled EUT: Device is in SDCI communication mode, VendorID and DeviceID match IODD
Procedure	a) Select parameter from <DefaultIIODD> fulfilling the conditions in "Test parameter" b) Call SystemCommand "ParamDownloadStart" if Device supports Block Parameterization c) Write parameter set 1 into the Device because it is different to factory reset d) Write SystemCommand " ParamDownloadStore" (5 / 0x05) e) Write SystemCommand "Back-to-box" (131 / 0x83) f) Wait 5 s g) Check communication h) Apply power cycle (power off, wait 5 s, power on: Master restarts communication) i) Wait until communication is re-established j) Check if a communication startup sequence has been triggered k) Check DS_UPLOAD_FLAG in parameter DataStorageIndex l) Read parameter DID (Device ID) and RID (Revision ID) and store to <idvalue_b> m) Check if an Event has been triggered (mode "Event appears") n) Read parameter DeviceStatus and store value to <status_b> o) Read selected parameter from a) and store values in array <parvalue_b>
Test parameter	Only parameters in IODD are tested - with attribute "accessRights = "rw", and - which are not "excludedFromDataStorage", and - with attribute: defaultValue="<value>" In case of the DeviceId is not equal to the DefaultDeviceId, the Test system needs the following information in addition - the DefaultDeviceId to which the device is switched back - the default parameter from the DefaultIODD - The Testparameter must be taken from the DefaultIODD for comparison
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Procedure g) shows communication interruption and procedure j) shows exactly one communication startup sequence 2) Values of <idvalue_b> match the default values of the IODD RID: (CommNetworkProfile: iolinkRevision = "Vx.x") 3) Value of procedure k) (DS_UPLOAD_FLAG) = "0" (no DS_UPLOAD_REQ) 4) Procedure k) shows at least one Event of mode "Event appears" or value of <status_b> = "0 (0x00)" For each read parameter (from procedure a)): 5) Response is positive 6) Values of <parvalue_b> match assigned "defaultValue"
Test passed	All evaluations 1) to 6) are true
Test not passed	At least one of the evaluations 1) to 6) is false
Report	Communication from evaluation 1): <ok | nok> RevisionID/DeviceID from evaluation 2): <ok | nok> DataStorage from evaluation 3): <ok | nok> DeviceStatus from evaluation 4): <value>, <event> <ok | nok> For each received parameter (from evaluation 5) and 6)): Parameter: <Index>, <parvalue_a>, <parvalue_b> <ok | nok>

7.4.7 IODD – Write alternative valid DeviceID

Table 178 defines the test conditions for this test case.

Table 178 – IODD – Write alternative valid DeviceID

-CR068-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0314
Name	TCD_DLPC_DID_OVERWRITE_COMP
Purpose (short)	Device behavior with overwrite of the DeviceID (compatible)
Equipment under test (EUT)	Device with support of an additional compatible DeviceID
Test case version	1.2
Category / type	IODD functional system test: test to pass
Specification (clause)	[6], 10.6.2, B.1.9
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Device behavior with overwrite of the DeviceID with a compatible DeviceID as listed in the IODD. Master overwrites the DeviceID and the Device accepts the requested DeviceID. After a power off/on, Master checks if the previously written DeviceID has been stored in non-volatile memory in the Device.
Precondition	DTU: SIO mode EUT: Device set to ("DID 0 ")
Procedure	a) DTU establishes communication b) DTU reads DeviceID from Device (EUT) c) DTU overwrites the DeviceID with the DeviceID ("DID-2") by using the complete mechanism written in [19], Figure 76, including Restart d) Device (EUT) power switched off and on again e) DTU re-establishes communication f) DTU reads DeviceID from Device (EUT) g) DTU overwrites the DeviceID(“DID-2”) with “DID-0” to return to the previous settings for the next test by using the complete mechanism written in [19], Figure 76, including Restart
Test parameter	IODD: DeviceIdentity/@deviceId, memorized as "DID-0" DeviceIdentity/@additionalDeviceIds, one of the DeviceIds memorized as "DID-2"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check acquired DeviceID after b) 2) Check acquired DeviceID after f)

Test passed	DeviceID = “DID-0”in 1)
	DeviceID = "DID-2" in 2)

Test not passed (examples)	Any of the checks failed.
Report	Additional DeviceID stored in non-volatile memory: <yes/no> <ok | nok>

7.4.8 IODD – Write alternative invalid DeviceID

Table 179 defines the test conditions for this test case.

Table 179 – IODD – Write alternative invalid DeviceID

-CR067-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0315
Name	TCD_DLPC_DID_OVERRIDE_INCOMP
Purpose (short)	Device behavior with overwrite of the DeviceID (incompatible)
Equipment under test (EUT)	Device with or without support of an additional incompatible DeviceID
Test case version	1.2
Category / type	IODD functional system test: test to fail
Specification (clause)	[6], 10.6.2
Configuration / setup	Device-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Device behavior at overwrite of the DeviceID with incompatible value. Master overwrites the DeviceID and the Device rejects the requested DeviceID.
Precondition	DTU: SIO mode EUT: Device set to ("DID-0")
Procedure	a) DTU establishes communication b) DTU reads DeviceID from Device (EUT) c) DTU overwrites the DeviceID with any DeviceID (≠ "DID-0" or "DID-x") by using the complete mechanism written in [7], Figure 76, including Restart d) DTU reads DeviceID from Device (EUT) e) Device (EUT) power switched off and on again f) DTU re-establishes communication g) DTU reads DeviceID from Device (EUT)
Test parameter	IODD: DeviceIdentity/@deviceId, memorized as "DID-0" DeviceIdentity@additionalDeviceIds = "DID-x", the additional DeviceIDs in the IODD
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check acquired DeviceID after b) 2) Check acquired DeviceID after d) 3) Check acquired DeviceID after g)

Test passed	DeviceID = “DID-0” in 1)
	DeviceID = "DID-0" in 2)
	DeviceID = "DID-0" in 3)

Test not passed (examples)	Any of the evaluation failed
Report	Device don’t change to invalid DeviceID: <yes/no> <ok | nok>

7.5 Test report summary of the IODD based Device tests

The template is defined by the Device-Tester-System. The test report shall present at least the results of the test cases for the IODD based Device tests.

8 Master protocol tests

8.1 General

-CR019-

The protocol tests can be performed almost automatically with the help of a Master-Tester-System as defined in A.3. The test sequences are described in 4.5 together with a list of the relevant test cases for Master in Table 6. Supplementary requirements for Legacy-Masters beyond the definitions in [5] are listed in Annex B.

Procedure results not mentioned in the evaluation part are assumed to be successful for passing the tests.

Master test cases are specified using appropriate types of macros as defined in Annex A.4.

8.2 Timings

8.2.1 Delay times after WURQ and Master messages (TDMT)

Table 180 defines the test conditions for this test case.

Table 180 – Delay times after WURQ and Master messages (TDMT)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0158
Name	TCM_ PHYL_TIME_TDMT
Purpose (short)	Check delay times after WURQ and Master messages
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.2.2, Table 42
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measuring the delays between WURQ and first request (230,4 kbit/s) or between the requests in the individual transmission rates, respectively. Check if TDMT is within the tolerance of 27 to 37 TBIT of the subsequent transmission rate.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_Deactivate b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) MTU_Timing_Startup *;returning TDMT(COM3), TDMT(COM2), TDMT(COM1)* c) Evaluation 1) d) Evaluation 2) e) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TDMT(COM3) *;See Figure 31 in* [6]*, time from begin WURQ to begin start bit of COM3 minus TREN with 500 µs* 2) Check TDMT(COM2) *;See Figure 31 in* [6]*, time from begin stop bit COM3 to begin start bit COM2 minus 1 TBIT(COM3)* 3) Check TDMT(COM1) *;See Figure 31 in* [6]*, time from begin stop bit COM2 to begin start bit COM1 minus 1 TBIT(COM2)*
Test passed	In 1) 27 TBIT ≤ TDMT(COM3) ≤ 37 TBIT *;TBIT = 4,34 µs* In 2) 27 TBIT ≤ TDMT(COM2) ≤ 37 TBIT *;TBIT = 26,04 µs* In 3) 27 TBIT ≤ TDMT(COM1) ≤ 37 TBIT *;TBIT = 208,33 µs*

TEST CASE RESULTS	CHECK / REACTION	
Test not passed (examples)	Any TDMT out of tolerance	
Report	TDMT(COM3): <value> TDMT(COM2): <value> TDMT(COM1): <value>	<ok | nok> <ok | nok> <ok | nok>

1927

8.2.2 Delay time between three WURQs (TDWU)

Table 181 defines the test conditions for this test case.

Table 181 – Delay time between three WURQs (TDWU)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0159
Name	TCM_ PHYL_TIME_TDWU
Purpose (short)	Check whether delay time between wake-up retries is within tolerance
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.2.2, Table 42
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measuring the times between the three Wake-up retries. Master-Tester-Unit shall detect the beginning of all Wake-up requests and measure the time in between. It shall not react to requests.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_Deactivate a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) MTU_Timing_Startup *;returning TDWU12 and TDWU23* c) Evaluation 1) d) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check interval between first and second Wake-up: TDWU12 2) Check interval between second and third Wake-up: TDWU23
Test passed	30 ms ≤ TDWU12 ≤ 50 ms, and 30 ms ≤ TDWU23 ≤ 50 ms
Test not passed (examples)	TDWU12 or TDWU23 out of tolerance
Report	TDWU12 (interval between first and second): <value> <ok | nok> TDWU23 (interval between second and third): <value> <ok | nok>

8.2.3 Number of WURQs

Table 182 defines the test conditions for this test case.

Table 182 – Number of WURQs

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0160
Name	TCM_ PHYL_TIME_NUMOFWURQS
Purpose (short)	Check number of Wake-up retries
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.2.2, Table 42
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The number of WURQs during one Wake-up sequence is counted. The number of retries shall be exactly 2 (total number 3). Master-Tester-Unit shall detect the start of the first WURQ and then start time measurement. Another 2 WURQs shall follow within the next 100 ms. No further requests except these 3 WURQs shall follow within 500 ms after start of the first WURQ.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_Deactivate a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) MTU_Timing_Startup *;returning WURQ and WURQ101-500 counts* c) Evaluation 1) d) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check number of WURQs within first 100 ms: WURQ 2) Check number of WURQs within 101 ms to 500 ms: WURQ101-500

Test passed	WURQ = 3, and
	WURQ101-500 = 0

Test not passed (examples)	WURQ ≠ 3, or WURQ101-500 > 0
Report	WURQ: <value> <ok | nok> WURQ101-500: <value> <ok | nok>

8.2.4 Delay time between WURQ retry sequences (TSD)

Table 183 defines the test conditions for this test case.

Table 183 – Delay time between WURQ retry sequences (TSD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0161
Name	TCM_ PHYL_TIME_TSD
Purpose (short)	Check time between two WURQs is between 0,5 s and 1 s.
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.2.2, Table 42
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The time between start of two Wake-up sequences is measured. Master-Tester-Unit shall detect start of the first WURQ and start time measurement. Two more WURQs shall follow within the next 100 ms. No further requests except these 3 WURQs shall follow within 500 ms after start of the first WURQ. At the latest 1 s after the first WURQ, the Master shall start a new Wake-up sequence. Master-Tester-Unit shall check these times.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_Deactivate a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) MTU_Timing_Startup *;returning TSD* c) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TSD
Test passed	0,5 s ≤ TSD ≤ 1 s
Test not passed (examples)	TSD > 1 s
Report	TSD: <value> <ok | nok>

8.2.5 Delay time between two Master messages at STARTUP (TINITCYC)

Table 184 defines the test conditions for this test case.

Table 184 – Delay time between two Master messages at STARTUP (TINITCYC)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0162
Name	TCM_ PHYL_TIME_TINITCYC
Purpose (short)	STARTUP: The time between two message beginnings shall be ≥ 100 TBIT.
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.3.3, A.2.6, Table A.7
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	In state STARTUP, the time between two Master message beginnings is measured. This is repeated for all transmission rates supported by the Master. Master-Tester-Unit detects the start of each Master message and measures the time in between. These times shall be ≥ 100 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_Timing_GetTinitcycInStartup *;returning TINITCYC* f) Evaluation 1) g) SMI_PortConfiguration(ABPS_PORTINACTIVE) h) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TINITCYC
Test passed	Minimum of TINITCYC (COM1) ≥ 100 TBIT, and Minimum of TINITCYC (COM2) ≥ 100 TBIT, and Minimum of TINITCYC (COM3) ≥ 100 TBIT
Test not passed (examples)	Any of the minima of TINITCYC is < 100 TBIT
Report	Minimum of TINITCYC (COM1): <value> Minimum of TINITCYC (COM2): <value> Minimum of TINITCYC (COM3): <value> <ok | nok>

8.2.6 Delay time of Master messages with 2 octets OD in PREOP (TINITCYC)

Table 185 defines the test conditions for this test case.

Table 185 – Delay time of Master messages with 2 octets OD in PREOP (TINITCYC)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0331
Name	TCM_ PHYL_TIME_TINITCYC_PREOP_2
Purpose (short)	PREOPERATE: TINITCYC of messages with 2 octets OD shall be ≥ 100 TBIT
Equipment under test (EUT)	Master
Test case version	1.0
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.3.3, A.2.6, Table A.8
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	In state PREOPERATE, the time between the start of two Master messages with 2 octets On-request Data is measured. This is repeated for all transmission rates supported by the Master. Master-Tester-Unit detects the start of each Master message and measures the time in between. These times shall be ≥ 100 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE *;2 octets OD*
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_Timing_GetTinitcycInPreoperate *;returning TINITCYC* f) Evaluation 1) g) SMI_PortConfiguration(ABPS_PORTINACTIVE) h) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TINITCYC
Test passed	Minimum of TINITCYC (COM1) ≥ 100 TBIT, and Minimum of TINITCYC (COM2) ≥ 100 TBIT, and Minimum of TINITCYC (COM3) ≥ 100 TBIT
Test not passed (examples)	Any of the minima of TINITCYC is at least < 100 TBIT
Report	Minimum of TINITCYC (COM1): <value> Minimum of TINITCYC (COM2): <value> Minimum of TINITCYC (COM3): <value> <ok | nok>

8.2.7 Delay time of Master messages with 8 octets OD in PREOP (TINITCYC)

Table 186 defines the test conditions for this test case.

Table 186 – Delay time of Master messages with 8 octets OD in PREOP (TINITCYC)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0332
Name	TCM_ PHYL_TIME_TINITCYC_PREOP_8
Purpose (short)	PREOPERATE: TINITCYC of messages with 8 octets OD shall be ≥ 210 TBIT
Equipment under test (EUT)	Master
Test case version	1.0
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.3.3, A.2.6, Table A.8
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	In state PREOPERATE, the time between the start of two Master messages with 8 octets On-request Data is measured. This is repeated for all transmission rates supported by the Master. Master-Tester-Unit detects the start of each Master message and measures the time in between. These times shall be ≥ 210 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) MTU_DPP1_Set(M-sequenceCapability) = 0x21 *;8 octets On-request Data* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_Timing_GetTinitcycInPreoperate *;returning TINITCYC* g) Evaluation 1) h) SMI_PortConfiguration(ABPS_PORTINACTIVE) i) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TINITCYC
Test passed	Minimum of TINITCYC (COM1) ≥ 210 TBIT, and Minimum of TINITCYC (COM2) ≥ 210 TBIT, and Minimum of TINITCYC (COM3) ≥ 210 TBIT
Test not passed (examples)	Any of the minima of TINITCYC is at least < 210 TBIT
Report	Minimum of TINITCYC (COM1): <value> Minimum of TINITCYC (COM2): <value> Minimum of TINITCYC (COM3): <value> <ok | nok>

8.2.8 Delay time of Master messages with 32 octets OD in PREOP (TINITCYC)

Table 187 defines the test conditions for this test case.

Table 187 – Delay time of Master messages with 32 octets OD in PREOP (TINITCYC)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0333
Name	TCM_ PHYL_TIME_TINITCYC_PREOP_32
Purpose (short)	PREOPERATE: TINITCYC of messages with 32 octets OD shall be ≥ 550 TBIT
Equipment under test (EUT)	Master
Test case version	1.0
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.3.3, A.2.6, Table A.8
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	In state PREOPERATE, the time between the start of two Master messages with 32 octets On-request Data is measured. This is repeated for all transmission rates supported by the Master. Master-Tester-Unit detects the start of each Master message and measures the time in between. These times shall be ≥ 550 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) MTU_DPP1_Set(M-sequenceCapability) = 0x31 *;32 octets On-request Data* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_Timing_GetTinitcycInPreoperate *;returning TINITCYC* g) Evaluation 1) h) SMI_PortConfiguration(ABPS_PORTINACTIVE) i) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TINITCYC
Test passed	Minimum of TINITCYC (COM1) ≥ 550 TBIT, and Minimum of TINITCYC (COM2) ≥ 550 TBIT, and Minimum of TINITCYC (COM3) ≥ 550 TBIT
Test not passed (examples)	Any of the minima of TINITCYC is at least < 550 TBIT
Report	Minimum of TINITCYC (COM1): <value> Minimum of TINITCYC (COM2): <value> Minimum of TINITCYC (COM3): <value> <ok | nok>

8.2.9 Adjustment of the MasterCycleTime

Table 188 defines the test conditions for this test case.

Table 188 – Adjustment of the MasterCycleTime

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0163
Name	TCM_ PHYL_TIME_MASTERCYCLETIME
Purpose (short)	The Master shall adapt correctly to a too short MinCycleTime of the Device
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], B.1.3
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master-Tester-Unit shall suggest the M-sequence type for a too small MinCycleTime value. The Master shall not accept this time and shall write back a feasible Master-Cycle time. The same applies if the MinCycleTime value is "0". At each M-sequence type, MTU starts with the MinCycleTime value "0" and with times below the time the Master can achieve. The Master shall correct these times by writing back a possible MasterCycleTime value.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com", "minCT" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" *;"Device" uses COMx* c) MTU_DPP1_Set(MinCycleTime) = "minCT" *;"Device" uses too small minCT* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;Master corrects value* e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_DPP1_Get(MasterCycleTime) *;returning mstCT* g) Evaluation 1) h) SMI_PortConfiguration(ABPS_PORTINACTIVE) i) Repeat from b) with next "com", "minCT"

Test parameter	com = {COM1, COM1, COM2, COM2, COM3, COM3}
	minCT = {0x00, 0x52 (13,6 ms), 0x00, 0x11 (1,7 ms), 0x00, 0x03 (0,3 ms)}

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check all mstCT
Test passed	COM1, minCT = 0x00: mstCT ≥ 0x5B (17,2 ms) COM1, minCT = 0x52: mstCT ≥ 0x53 (14,0 ms) COM2, minCT = 0x00: mstCT ≥ 0x16 (2,2 ms) COM2, minCT = 0x11: mstCT ≥ 0x12 (1,8 ms) COM3, minCT = 0x00: mstCT ≥ 0x04 (0,4 ms) COM3, minCT = 0x03: mstCT ≥ 0x04 (0,4 ms)
Test not passed (examples)	Any of the mstCT values < Master limits (see Test passed)
Report	mstCT (COM1, minCT = 0x00): <value> <ok | nok> mstCT (COM1, minCT = 0x52): <value> <ok | nok> mstCT (COM2, minCT = 0x00): <value> <ok | nok> mstCT (COM2, minCT = 0x11): <value> <ok | nok> mstCT (COM3, minCT = 0x00): <value> <ok | nok> mstCT (COM3, minCT = 0x03): <value> <ok | nok>

8.2.10 Written MasterCycleTime corresponds to real cycle time

Table 189 defines the test conditions for this test case. It should be noted that communication load at other Ports can interfere with the performance of this test.

Table 189 – Written MasterCycleTime corresponds to real cycle time

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0164
Name	TCM_ PHYL_TIME_MASTERCYCLETIMEREAL
Purpose (short)	Written MasterCycleTime matches real cycle time
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 7.3.3.3, A.2.6, Table A.11
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Master shall read the MinCycleTime value from Master-Tester-Unit and shall write back its MasterCycleTime value. This time shall be checked by the Master-Tester-Unit. To do so, Master-Tester-Unit shall start several times with different MinCycleTimes and then check them. Master-Tester-Unit receives different values in the Direct Parameter page 1 for the MinCycleTime and carries out a start-up to state OPERATE. In this state the MasterCycleTime is checked.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com", "minCT" b) MTU_Timing_SetCommunicationMode = "com *;"Device" uses COMx* c) MTU_DPP1_Set(MinCycleTime) = "minCT" *;"Device" uses too small minCT* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;Master corrects value* e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_DPP1_Get(MasterCycleTime) *;returning Texp* g) MTU_Timing_GetTcycInOperate *;returning TCYC* h) Evaluation 1) j) SMI_PortConfiguration(ABPS_PORTINACTIVE) k) Repeat from b) with next "com", "minCT"

Test parameter	com = {COM1, COM1, COM1, COM1, COM1, COM2, COM2, COM2, COM2, COM2, COM2, COM2, COM3, COM3, COM3, COM3, COM3, COM3, COM3 }
	minCT = {0x00, 0x5D, 0x7F, 0x80, 0xBF, 0x00, 0x17, 0x3F, 0x40, 0x7F, 0x80, 0xBF, 0x00, 0x04, 0x3F, 0x40, 0x7F, 0x80, 0xBF} *;see Table A.11 in* [6] *for recommended MinCycleTimes and Table B.3 in* [6] *for crossover values*

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check TCYC and Texp
Test passed	Minimum of TCYC ≥ Texp - 1% Maximum of TCYC ≤ Texp +10%
Test not passed (examples)	Minimum of TCYC < Texp - 1% Maximum of TCYC > Texp +10%
Report	Minimum of TCYC: <value> <ok | nok> Maximum of TCYC: <value> <ok | nok>

8.2.11 Master tolerates different Device response times

Table 190 defines the test conditions for this test case.

Table 190 – Master tolerates different Device response times

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0165
Name	TCM_ PHYL_TIME_DEVRESPTIMES
Purpose (short)	Master tolerates different Device response times
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.3.5
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master-Tester-Unit shall answer with different response times after receiving the Master message for all transmission rates COM1, COM2, and COM3. The Master shall be able to handle deviations (jitter). Master-Tester-Unit responds with different response times between 1 and 10 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) Assign first value to "TA" (Test parameter) f) MTU_Timing_SetReplyMessageDelay(TA) = "TA" g) MTU_State_GetMasterRetryCTviolCount *;clear RETRIES, CTVIOLS* h) SMI_Device Read(ABPS_DEVICEREAD(Index=16)) *;Vendor-Name* i) MTU_State_GetMasterRetryCTviolCount *;returning RETRIES, CTVIOLS* j) Evaluation 1) k) Repeat from f) with next "TA" l) SMI_PortConfiguration(ABPS_PORTINACTIVE) m) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3} TA = {1 TBIT, 5 TBIT, 10 TBIT} *;1 TBIT or as fast as possible*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check RETRIES, CTVIOLS
Test passed	RETRIES = 0 and CTVIOLS = 0
Test not passed (examples)	RETRIES > 0 or CTVIOLS > 0
Report	CTVIOLS: <value> <ok | nok> RETRIES: <value> <ok | nok>

8.2.12 Master tolerates different UART frame delay times (T2)

Table 191 defines the test conditions for this test case.

Table 191 – Master tolerates different UART frame delay times (T2)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0166
Name	TCM_ PHYL_TIME_UARTT2
Purpose (short)	Master tolerates different UART frame delay times (T2) of the Device.
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.3.4
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master-Tester-Unit shall respond with different delays between the octets at all transmission rates COM1, COM2, and COM3. The Master shall be able to handle this variation. Master-Tester-Unit responds with different delays between 0 and 3 TBIT.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) Assign first value to "T2" (Test parameter) f) MTU_Timing_SetReplyMessageDelay(T2) = "T2" g) MTU_State_GetMasterRetryCTviolCount *;clear RETRIES, CTVIOLS* h) SMI_Device Read(ABPS_DEVICEREAD(Index=16)) *;Vendor-Name* i) MTU_State_GetMasterRetryCTviolCount *;returning RETRIES, CTVIOLS* j) Evaluation 1) k) Repeat from f) with next "T2" l) SMI_PortConfiguration(ABPS_PORTINACTIVE) m) Repeat from b) with next "com"

Test parameter	com = {COM1, COM2, COM3}
	T2 = {0 TBIT, 1 TBIT, 2 TBIT, 3 TBIT}

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check RETRIES, CTVIOLS
Test passed	RETRIES = 0 and CTVIOLS = 0
Test not passed (examples)	RETRIES > 0 or CTVIOLS > 0
Report	CTVIOLS: <value> <ok | nok> RETRIES: <value> <ok | nok>

8.2.13 Master sends UART frames within tolerated times (T1)

Table 192 defines the test conditions for this test case.

Table 192 – Master sends UART frames within tolerated times (T1)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0167
Name	TCM_ PHYL_TIME_UARTT1
Purpose (short)	Master sends UART frames within tolerated times (T1)
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.3.3
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Measuring the delays between the UART octets for all transmission rates COM1, COM2, and COM3. Master-Tester-Unit shall measure the delays between the end of the stop bit and the beginning of the start bit of the next octet.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" (Test parameter) b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_Timing_Get_T1 *;returning T1* f) Evaluation 1) g) SMI_PortConfiguration(ABPS_PORTINACTIVE) h) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check T1
Test passed	Maximum of T1 ≤ 1 TBIT
Test not passed (examples)	Maximum of T1 > 1 TBIT
Report	Maximum of T1: <value> <ok | nok>

8.3 Process Data (PD)

8.3.1 TYPE_2_1 for 8 bit PD input

Table 193 defines the test conditions for this test case.

Table 193 – TYPE_2_1 for 8 bit PD input

-CR086- -CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0168
Name	TCM_ DLPD_CYCC_TYPE21BIT8IN
Purpose (short)	Master uses M-sequence TYPE_2_1 for 8 bit Process Data input
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 5)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 8 bit and output length = 0 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_1 in OPERATE and propagate this information to SMI_PDin, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x08 *;"8" Bit* c) MTU_DPP1_Set(PDOut length) = 0x00 *;"0" Bit* d) MTU_PD_Set = 0x12 *;PDIn = 0x12* e) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;Switch to OPERATE* f) TM_AWAIT_PORT_STATUS(OPERATE) *;Wait for OPERATE* g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_PDIn *;returns "ArgBlock PDIn"* j) Evaluation 2) k) SMI_PDInOut *;returns "ArgBlock PDInOut"* l) Evaluation 3) m) SMI_PortStatus *;returns "ArgBlock PortStatusList"* n) Evaluation 4) o) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) p) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn" 3) Check "ArgBlock PDInOut" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 1, PDI0 = [0x12], and PDInOut: PQI = 0xA0, OE = 0x00, InputDataLength = 1, PDI0 = [0x12], OutputDataLength = 0, and PortStatusList: PortQualityInfo = 0x02, RevisionID = 0x11, InputDataLength = 1, OutputDataLength = 0, and <vendorname> = IO-Link Community
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.2 TYPE_2_2 for 16 bit PD input

Table 194 defines the test conditions for this test case.

Table 194 – TYPE_2_2 for 16 bit PD input

-CR086- -CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0169
Name	TCM_ DLPD_CYCC_TYPE22BIT16IN
Purpose (short)	Master uses M-sequence TYPE_2_2 for 16 bit Process Data input
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 6)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 16 bit and output length = 0 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_2 in OPERATE and propagate this information to SMI_PDin, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x10 *;"16" Bit* c) MTU_DPP1_Set(PDOut length) = 0x00 *;"0" Bit* d) MTU_PD_Set = 0x1234 *;PDIn = 0x1234* e) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;Switch to OPERATE* f) TM_AWAIT_PORT_STATUS (OPERATE) *;Wait for OPERATE* g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_PDIn *;returns "ArgBlock PDIn"* j) Evaluation 2) k) SMI_PDInOut *;returns "ArgBlock PDInOut"* l) Evaluation 3) m) SMI_PortStatus *;returns "ArgBlock PortStatusList"* n) Evaluation 4) o) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) p) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn" 3) Check "ArgBlock PDInOut" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 2, PDI = [0x12, 0x34], and PDInOut: PQI = 0xA0, OE = 0x00, InputDataLength = 2, PDI = [0x12, 0x34], OutputDataLength = 0, and PortStatusList: PortQualityInfo = 0x02, RevisionID = 0x11, InputDataLength = 2, OutputDataLength = 0, and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.3 TYPE_2_3 for 8 bit PD output

Table 195 defines the test conditions for this test case.

Table 195 – TYPE_2_3 for 8 bit PD output

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0170
Name	TCM_ DLPD_CYCC_TYPE23BIT8OUT
Purpose (short)	Master uses M-sequence TYPE_2_3 for 8 bit Process Data output
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 7)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 0 bit and output length = 8 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_3 in OPERATE and propagate this information to SMI_PDIn, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x00 *;"0" Bit* c) MTU_DPP1_Set(PDOut length) = 0x08 *;"8" Bit* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;Switch to OPERATE* e) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE state* f) MTU_State_CheckOperate *;returns "MTU in OPERATE"* g) Evaluation 1) h) SMI_PDOut(ABPS_PDOUT<PDO0=0x12>) *;PDO0 = 0x12* i) SMI_PDInOut *;returns "ArgBlock PDInOut"* j) Evaluation 2) k) MTU_PD_Get *;returns "PDOut values"* l) Evaluation 3) m) SMI_PortStatus *;returns "ArgBlock PortStatusList"* n) Evaluation 4) o) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) p) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDInOut" 3) Check "PDout values" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDInOut: PQI = 0xA0, OE = 1, InputDataLength = 0, OutputDataLength = 1, PDO0 = [0x12], and "PDout values" = 0x12, and PortStatusList: PortQualityInfo = 0x00, RevisionID = 0x11, InputDataLength = 0, OutputDataLength = 1, and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.4 TYPE_2_4 for 16 bit PD output

Table 196 defines the test conditions for this test case.

Table 196 – TYPE_2_4 for 16 bit PD output

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0171
Name	TCM_ DLPD_CYCC_TYPE24BIT16OUT
Purpose (short)	Master uses M-sequence TYPE_2_4 for 16 bit Process Data output
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 8)
Configuration / setup	Master-Tester-System
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master reads PD input length = 0 bits, Output length = 16 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_4 in OPERATE and propagate this information to SMI_PDIn, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x00 *;"0" Bit* c) MTU_DPP1_Set(PDOut length) = 0x10 *;"16" Bit* d) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;switch to OPERATE* e) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* f) MTU_State_CheckOperate *;returns "MTU in OPERATE"* g) Evaluation 1) h) SMI_PDOut(ABPS_PDOUT<PDO0=0x12, PDO1=0x34>) *;set PDOut = 0x1234* i) SMI_PDInOut *;returns "ArgBlock PDInOut"* j) Evaluation 2) k) MTU_PD_Get *;returns "PDOut values"* l) Evaluation 3) m) SMI_PortStatus *;returns "ArgBlock PortStatusList"* n) Evaluation 4) o) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDInOut" 3) Check "PDout values" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDInOut: PQI = 0xA0, OE = 1, InputDataLength = 0, OutputDataLength = 2, PDO = [0x12, 0x34], and "PDout values" = 0x12, 0x34, and PortStatusList: PortQualityInfo = 0x00, RevisionID = 0x11, InputDataLength = 0, OutputDataLength = 2, and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.5 TYPE_2_5 for 8/8 bit PD in/output

Table 197 defines the test conditions for this test case.

Table 197 – TYPE_2_5 for 8/8 bit PD in/output

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0172
Name	TCM_ DLPD_CYCC_TYPE25BIT8INBIT8OUT
Purpose (short)	Master uses M-sequence TYPE_2_5 for 8/8 bit Process Data in/output
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 9)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 8 bits, Output length = 8 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_5 in OPERATE and propagate this information to the SMI_PDIn, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x08 *;”8” Bit* c) MTU_DPP1_Set(PDOut length) = 0x08 *;”8” Bit* d) MTU_PD_Set = 0x12 *;PDIn = 0x12* e) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;Switch to OPERATE* f) TM_AWAIT_PORT_STATUS(OPERATE) *;Wait for OPERATE* g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_PDOut(ABPS_PDOUT<PDO0=0x34>) *;PDO0 = 0x34* j) SMI_PDIn *;returns "ArgBlock PDIn"* k) Evaluation 2) l) SMI_PDInOut *;returns "ArgBlock PDInOut"* m) Evaluation 3) n) MTU_PD_Get *;returns "PDOut values"* o) Evaluation 4) p) SMI_PortStatus *;returns "ArgBlock PortStatusList"* q) Evaluation 5) r) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) s) Evaluation 6)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn" 3) Check "ArgBlock PDInOut"	4) Check "PDout values" 5) Check "ArgBlock PortStatusList" 6) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 1, PDI0 = [0x12], and PDInOut: PQI = 0xA0, OE = 1, InputDataLength = 1, PDI0 = [0x12], OutputDataLength = 1, PDO0 = [0x34], and "PDout values" = 0x34, and PortStatusList: PortQualityInfo = 0x00, RevisionID = 0x11, InputDataLength = 1, OutputDataLength = 1, and <vendorname> = "IO-Link Community"	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	

TEST CASE RESULTS	CHECK / REACTION
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.6 TYPE_2_V for 16/16 bit PD in/output and 1 octet OD

Table 197 defines the test conditions for this test case.

Table 198 – TYPE_2_ V for 16/16 bit PD in/output and 1 octet OD

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0298
Name	TCM_ DLPD_CYCC_TYPE2VBIT16INBIT16OUT
Purpose (short)	Master uses M-sequence TYPE_2_V for 16/16 bit Process Data in/output
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 10 and 11)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 16 bit, Output length = 16 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_V with 1 octet OD in OPERATE and propagate information to SMI_PDIn, SMI_PDInOut, and SMI_PortStatus.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x10 *;"16" Bit* c) MTU_DPP1_Set(PDOut length) = 0x10 *;"16" Bit* d) MTU_PD_Set = 0x1234 *;PDIn = 0x1234* e) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;switch to OPERATE* f) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_PDOut(ABPS_PDOUT<PDO0=0x56, PDO1=0x78>) *;PDO = 0x5678* j) SMI_PDIn *;returns "ArgBlock PDIn"* k) Evaluation 2) l) SMI_PDInOut *;returns "ArgBlock PDInOut"* m) Evaluation 3) n) MTU_PD_Get *;returns "PDOut values"* o) Evaluation 4) p) SMI_PortStatus *;returns "ArgBlock PortStatusList"* q) Evaluation 5) r) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) s) Evaluation 6)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn" 3) Check "ArgBlock PDInOut" 4) Check "PDout values" 5) Check "ArgBlock PortStatusList" 6) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 2, PDI = [0x12, 0x34], and PDInOut: PQI = 0xA0, OE = 0x01, InputDataLength = 2, PDI = [0x12, 0x34], OutputDataLength = 2, PDO = [0x56, 0x78], and "PDout values" = 0x56, 0x78, and PortStatusList: PortQInfo = 0x00, RevID = 0x11, InDLen = 2, OutDLen = 2, and <vendorname> = "IO-Link Community"

TEST CASE RESULTS	CHECK / REACTION	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	
Report	<vendorname> correct: <yes/no>	<ok | nok>

2055

8.3.7 TYPE_1_1 for 32 octets PD in and 2 octets OD

Table 199 defines the test conditions for this test case.

Table 199 – TYPE_1_1 for 32 octets PD in and 2 octets OD

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0173
Name	TCM_ DLPD_CYCC_TYPE1OCTET32IN
Purpose (short)	Master uses M-sequence TYPE_1_1 for 32/0 octets PD input and 2 octets OD
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 3)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads PD input length = 256 bits, Output length = 0 bit during STARTUP from Device. Master shall select M-sequence TYPE_1_1 with 2 octets OD in OPERATE and propagate this information to SMI_PDIn, SMI_PDInOut and SMI_PortStatus services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x13 *;see B.1.4 in [6]* b) MTU_DPP1_Set(RevisionID) = 0x10 *;see B.1.1 in [6]* c) MTU_DPP1_Set(PDIn length) = 0x100 *;"256" Bit = 32 octets* d) MTU_DPP1_Set(PDOut length) = 0x00 *;"0" Bit* e) MTU_PD_Set = 2^256-1 *;all Bits "1"* f) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;switch to OPERATE* g) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* h) MTU_State_CheckOperate *;returns "MTU in OPERATE"* i) Evaluation 1) j) SMI_PDIn *;returns "ArgBlock PDIn"* k) Evaluation 2) l) SMI_PDInOut *;returns "ArgBlock PDInOut"* m) Evaluation 3) n) SMI_PortStatus *;returns "ArgBlock PortStatusList"* o) Evaluation 4) p) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) q) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn" 3) Check "ArgBlock PDInOut" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 0x20, PDI[0 … 31] = 0xFF, and PDInOut: PQI = 0xA0, OE = 0x00, InputDataLength = 0x20, PDI[0 … 31] = 0xFF, OutputDataLength = 0, and PortStatusList: PortQualityInfo = 0x02, RevisionID = 0x10, InputDataLength = 0x20, OutputDataLength = 0, and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.8 TYPE_2_V for 0/32 octets PD in/out and variable octets OD

Table 200 defines the test conditions for this test case.

Table 200 – TYPE_2_V for 0/32 octets PD in/out and variable octets OD

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0334
Name	TCM_ DLPD_CYCC_ TYPE2VOD1IN0OUT32OCTET
Purpose (short)	Master uses M-sequence TYPE_2_V for 0/32 octets PD in/out and variable ODs
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (rows 12, 15, 17, and 19)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads OD length = 1 and PD input length = 0 bits, Output length = 256 bits during STARTUP from Device. Master shall select M-sequence TYPE_2_V in OPERATE and propagate this information to SMI_PDIn, SMI_PDInOut and SMI_PortStatus services. Different OD lengths are tested within the test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "Mscap" (Test parameter) b) MTU_DPP1_Set(M-sequenceCapability) = "Mscap" *;see B.1.4 in [6]* c) MTU_DPP1_Set(PDIn length) = 0x00 *;"0" Bit* d) MTU_DPP1_Set(PDOut length) = 0x100 *;"256" Bit* e) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;switch to OPERATE* f) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_PDOut(ABPS_PDOUT<PDO[0..31]=0xFF>) *;PDO all bits "1"* j) SMI_PDInOut *;returns "ArgBlock PDInOut"* k) Evaluation 2) l) MTU_PD_Get *;returns "PDOut values"* m) Evaluation 3) n) SMI_PortStatus *;returns "ArgBlock PortStatusList"* o) Evaluation 4) p) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) q) Evaluation 5) r) SMI_PortConfiguration (ABPS_PORT_INACTIVE) *;SIO mode* s) Repeat from b) with "Mscap" = next value from Test parameter
Test parameter	Mscap = {0x19, 0x1B, 0x1D, 0x1F} *;M-sequence capability*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock PDInOut"	3) Check "PDout values" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDInOut: PQI = 0xA0,, OE = 0x01, InputDataLength = 0x00, OutputDataLength = 0x20, PDO[0 … 31] = 0xFF, and PortStatusList: PortQualityInfo = 0x00, RevisionID = 0x11, InputDataLength = 0x20, OutputDataLength = 0, and <vendorname> = "IO-Link Community"	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	
Report	<vendorname> correct: <yes/no> <ok | nok>	

8.3.9 TYPE_2_V for 32/0 octets PD in/out and variable octets OD

Table 201 defines the test conditions for this test case.

Table 201 – TYPE_2_V for 32/0 octets PD in/out and variable octets OD

-CR072-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0335
Name	TCM_ DLPD_CYCC_ TYPE2VOD1IN32OUT0OCTET
Purpose (short)	Master uses M-sequence TYPE_2_V for 32/0 octets PD in/out and variable ODs
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5,11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (rows 13, 14, 16, and 18)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master reads OD length = 1 and PD input length = 256 bits, Output length = 0 bit during STARTUP from Device. Master shall select M-sequence TYPE_2_V in OPERATE and propagate this information to SMI_PDIn, SMI_PDInOut, and SMI_PortStatus services. Different OD lengths are tested within the test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "Mscap" (Test parameter) b) MTU_DPP1_Set(M-sequenceCapability) = "Mscap" *;see B.1.4 in [6]* c) MTU_DPP1_Set(PDIn length) = 0x100 *;"256" Bit* d) MTU_DPP1_Set(PDOut length) = 0x00 *;"0" Bit* e) MTU_PD_Set = 2^256-1 *;PDIn = all bits set* f) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* g) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* h) MTU_State_CheckOperate *;returns "MTU in OPERATE"* i) Evaluation 1) j) SMI_PDIn *;returns "ArgBlock PDIn"* k) Evaluation 2) l) SMI_PDInOut *;returns "ArgBlock PDInOut"* m) Evaluation 3) n) SMI_PortStatus *;returns "ArgBlock PortStatusList"* o) Evaluation 4) p) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>) q) Evaluation 5) r) SMI_PortConfiguration (ABPS_PORT_INACTIVE) *;SIO mode* s) Repeat from b) with "Mscap" = next value from Test parameter
Test parameter	Mscap = {0x19, 0x1B, 0x1D, 0x1F} *;M-sequence capability*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock PDIn"	3) Check "ArgBlock PDInOut" 4) Check "ArgBlock PortStatusList" 5) Check <vendorname>
Test passed	"MTU in OPERATE" = TRUE, and PDIn: PQI = 0xA0, InputDataLength = 0x20, PDI[0 … 31] = 0xFF, and PDInOut: PQI = 0xA0, OE = 0x00, InputDataLength = 0x20, PDI[0 … 31] = 0xFF, OutputDataLength = 0, and PortStatusList: PortQualityInfo = 0x02, RevisionID = 0x11, InputDataLength = 0x20, OutputDataLength = 0, and <vendorname> = "IO-Link Community"	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	

TEST CASE RESULTS	CHECK / REACTION
Report	<vendorname> correct: <yes/no> <ok | nok>

8.3.10 Master reads mirrored PD in/out from Device

Table 202 defines the test conditions for this test case.

Table 202 – Master reads mirrored PD in/out from Device

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0176
Name	TCM_ DLPD_CYCC_MIRROREDPD
Purpose (short)	Master reads mirrored Process Data in/out from Device
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 7.3.3.2, 9.2.3.5
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Device (MTU) mirrors its PD such that the Master can check consistency.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(PDIn length) = 0x10 *;"16" Bit* c) MTU_DPP1_Set(PDOut length) = 0x10 *;"16" Bit* d) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;switch to OPERATE* e) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* f) MTU_State_MirrorPD *;perform PD mirroring* g) SMI_PDOut(ABPS_PDOUT<OutputDataLength=2, PDO0=0x12, PDO1=0x34>) h) SMI_PDIn *;returns "ArgBlock PDIn1"* i) Evaluation 1) j) SMI_PDOut (ABPS_PDOUT<OutputDataLength=2, PDO0=0x56, PDO1=0x78>) k) SMI_PDIn *;returns "ArgBlock PDIn2"* l) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "ArgBlock PDIn1" 2) Check "ArgBlock PDIn2"
Test passed	PDIn1: PDI = [0x12, 0x34], and PDIn2: PDI = [0x56, 0x78]
Test not passed (examples)	Inconsistency between transmitted and received process data
Report	Sent and received PD match: <ok | nok>

8.3.11 Master propagates "PD invalid" indication in a correct manner

Table 203 defines the test conditions for this test case.

Table 203 – Master propagates "PD invalid" indication in a correct manner

-CR063-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0177
Name	TCM_ DLPD_CYCC_PDINVALID
Purpose (short)	Master propagates "PD invalid" indication in a correct manner
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], A.1.5 and Table A.5
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	PDValid – PDInvalid transition reported correctly.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* b) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* c) SMI_PDIn *;returns "ArgBlock PDIn_1"* d) Evaluation 1) e) SMI_PortStatus *;returns "ArgBlock PortStatusList_1"* f) Evaluation 2) g) MTU_State_SetPDValidity(INVALID) *;PD invalid* h) SMI_PDIn *;returns "ArgBlock PDIn_2"* f) Evaluation 3) i) SMI_PortStatus *;returns "ArgBlock PortStatusList_2"* j) Evaluation 4) k) SMI_PortConfiguration(ABPS_PORTINACTIVE) *;switch to Deactivated* l) MTU_State_SetPDValidity(VALID) *;PD valid*
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "ArgBlock PDIn_1" 2) Check "ArgBlock PortStatusList_1" 3) Check "ArgBlock PDIn_2" 4) Check "ArgBlock PortStatusList_2"
Test passed	PDIn_1: PQI = 0xA0, and PortStatusList_1: PortQualityInfo.Bit0 = VALID, and PDIn_2: PQI = 0x20, and PortStatusList_2: PortQualityInfo.Bit0 = INVALID
Test not passed (examples)	Master defines PDIn_1 in upper-level system as invalid and/or Master defines PDIn_2 in upper-level system as valid
Report	"PDValidity" propagated correctly to upper-level system: <ok | nok>

8.3.12 Master propagates "PD valid" indication in a correct manner

Table 204 defines the test conditions for this test case.

Table 204 – Master propagates "PD valid" indication in a correct manner

-CR064-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0178
Name	TCM_ DLPD_CYCC_PDVALID
Purpose (short)	Master propagates "PD Valid" indication in a correct manner
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], A.1.5 and Table A.5
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	PDInvalid – PDValid transition reported correctly.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_SetPDValidity(INVALID) *;PD invalid* b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* c) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* d) MTU_State_SetPDValidity(VALID) *;PD valid* e) SMI_PDIn *;returns "ArgBlock PDIn"* f) Evaluation 1) g) SMI_PortStatus ;*returns "ArgBlock PortStatusList"* h) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "ArgBlock PDIn" 2) Check "ArgBlock PortStatusList"
Test passed	PDIn: PQI = 0xA0, and PortStatusList: PortQualityInfo.Bit0 = VALID
Test not passed (examples)	Master defines the PD in the upper-level system as invalid
Report	"PDInvalid" = "0" propagated to upper-level system: <ok | nok>

8.4 On-request Data (OD)

8.4.1 TYPE_2_V for different PD in/out and 1 octet OD

Table 205 defines the test conditions for this test case.

Table 205 – TYPE_2_V for different PD in/out and 1 octet OD

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0179
Name	TCM_DLOD_CYCC_TYPE2VPDXOD1
Purpose (short)	Master uses TYPE_2_V for different PD in/out and 1 octet OD
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.4, A.2.6, B.1.4, E.4, E.10, E.11, E.12, Table A.10
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master uses TYPE_2_V for several different PD in/out combinations and 1 octet OD. The Master receives the information about the target M-sequence type via address 0x03 (M-sequenceCapability), 0x05 (ProcessDataIn), and 0x06 (ProcessDataOut) out of DPP1. Different PD lengths are tested within a test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_ISDU_Add(64, 0, 0x00) *;ISDU with value 1 x "0"* b) Assign first value to "pdinlen", "pdoutlen", "mseq" *;Test parameter* c) MTU_DPP1_Set(M-sequenceCapability) = "mseq" *;Test parameter* d) MTU_DPP1_Set(PDIn length) = "pdinlen" *;"16" Bit* e) MTU_DPP1_Set(PDOut length) = "pdoutlen" *;"0" Bit* f) MTU_PD_Set = [0x5A,…, "pdinlen"-times] *;set "PDin" in MTU* g) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* h) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* i) MTU_State_CheckOperate *;returns "MTU in OPERATE"* j) Evaluation 1) k) SMI_PDOut(ABPS_PDOUT<"pdoutlen", PDO0=0x5A>) *;set "PDout"* l) SMI_DeviceWrite(ABPS_DEVICEWRITE<Index=64, OD=0xBF>) m) SMI_DeviceRead(ABPS_DEVICEREAD<Index=64>) *;returns "ArgBlock OD"* n) Evaluation 2) o) MTU_PD_Get *;returns "PDOut values"* p) Evaluation 3) q) SMI_PDin *;returns "ArgBlock PDIn"* r) Evaluation 4) s) SMI_PortConfiguration(ABPS_PORTINACTIVE) *;deactivate Port* t) Repeat from c) with next values of "pdinlen", "pdoutlen", "mseq"

Test parameter	pdinlen = {1, 3, 32, 6, 0}, pdoutlen = {3, 1, 32, 0, 20} *;variable PD lengths*
	mseq = {0x11, 0x11, 0x19, 0x19, 0x19} *;M-sequenceCapability*

Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock OD"	3) Check "PDInOut values" 4) Check "ArgBlock PDin"
Test passed	"MTU in OPERATE" = TRUE, and On-request Data = 0xBF, and PDout length = pdoutlen, all "PDOut values" = "0x5A", and PDin.InputDataLength = pdinlen, all PDin.PDIx = "0x5A"	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	
Report	All M-sequence TYPEs (PD combinations): <ok | nok>	

8.4.2 TYPE_2_V for different PD in/out and 2 octets OD

Table 206 defines the test conditions for this test case.

Table 206 – TYPE_2_V for different PD in/out and 2 octets OD

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0180
Name	TCM_DLOD_CYCC_TYPE2VPDXOD2
Purpose (short)	Master uses TYPE_2_V for different PD in/out and 2 octets OD
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.4, A.2.6, B.1.4, E.4, E.10, E.11, E.12, Table A.10
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master uses TYPE_2_V for several different PD in/out combinations and 2 octets OD. The Master receives the information about the target M-sequence type via address 0x03 (M-sequenceCapability), 0x05 (ProcessDataIn), and 0x06 (ProcessDataOut) out of DPP1. Different PD lengths are tested within a test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x2B *;see B.1.4 in [6]* b) MTU_ISDU_Add(64, 0, [0x00, 0x00]) *;ISDU with values 2 x "0"* c) Assign first value to "pdinlen" and "pdoutlen" *;Test parameter* d) MTU_DPP1_Set(PDIn length) = "pdinlen" × 8 *;in Bits* e) MTU_DPP1_Set(PDOut length) = "pdoutlen" × 8 *;in Bits* f) MTU_PD_Set = [0x5A,…, "pdinlen"-times] *;set "PDin" in MTU* g) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* h) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* i) MTU_State_CheckOperate *;returns "MTU in OPERATE"* j) Evaluation 1) k) SMI_PDOut(ABPS_PDOUT<"pdoutlen",PDO0=0x5A>) *;set "PDout"* l) SMI_DeviceWrite(ABPS_DEVICEWRITE<Index=64, OD=[0xAA, 0xBF]>) m) SMI_DeviceRead(ABPS_DEVICEREAD<Index=64>)*;returns "ArgBlock OD"* n) Evaluation 2) o) MTU_PD_Get *;returns "PDInOut values"* p) Evaluation 3) q) SMI_PDin *;returns "ArgBlock PDIn"* r) Evaluation 4) s) SMI_PortConfiguration(ABPS_PORTINACTIVE) *;deactivate Port* t) Repeat from d) with next values of "pdinlen", "pdoutlen" *;Test parameter*
Test parameter	pdinlen = {1, 32, 6, 0}, pdoutlen = {3, 32, 0, 20} *;variable PD lengths in octets*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock OD" 3) Check "PDInOut values" 4) Check "ArgBlock PDin"
Test passed	"MTU in OPERATE" = TRUE, and On-request Data = [0xAA, 0xBF], and PDout length = pdoutlen, all "PDOut values" = "0x5A", and PDin.InputDataLength = pdinlen, all PDin.PDIx = "0x5A"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	All M-sequence TYPEs (PD combinations): <ok | nok>

8.4.3 TYPE_2_V for different PD in/out and 8 octets OD

Table 207 defines the test conditions for this test case.

Table 207 – TYPE_2_V for different PD in/out and 8 octets OD

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0181
Name	TCM_DLOD_CYCC_TYPE2VPDXOD8
Purpose (short)	Master uses TYPE_2_V for different PD in/out and 8 octets OD
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.4, A.2.6, B.1.4, E.4, E.10, E.11, E.12, Table A.10
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master uses TYPE_2_V for several different PD in/out combinations and 8 octets OD. The Master receives the information about the target M-sequence type via address 0x03 (M-sequenceCapability), 0x05 (ProcessDataIn), and 0x06 (ProcessDataOut) out of DPP1. Different PD lengths are tested within a test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x2D *;see B.1.4 in [6]* b) MTU_ISDU_Add(64, 0, [0x00, …,0x00]) *;ISDU with values 8 x "0"* c) Assign first value to "pdinlen" and "pdoutlen" *;Test parameter* d) MTU_DPP1_Set(PDIn length) = "pdinlen" × 8 *;in Bits* e) MTU_DPP1_Set(PDOut length) = "pdoutlen" × 8 *;in Bits* f) MTU_PD_Set = [0x5A,…, "pdinlen"-times] *;set "PDin" in MTU* g) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* h) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* i) MTU_State_CheckOperate *;returns "MTU in OPERATE"* j) Evaluation 1) k) SMI_PDOut(ABPS_PDOUT<"pdoutlen", PDO0=0x5A>) *;set "PDout"* l) SMI_DeviceWrite(ABPS_DEVICEWRITE<Index=64, OD=[0xAA, 0xBF, 0x77, 0x40, 0x55, 0xCD, 0x33, 0xE4]>) m) SMI_DeviceRead(ABPS_DEVICEREAD<Index=64>)*;returns "ArgBlock OD"* n) Evaluation 2) o) MTU_PD_Get *;returns "PDInOut values"* p) Evaluation 3) q) SMI_PDin *;returns "ArgBlock PDIn"* r) Evaluation 4) s) SMI_PortConfiguration(ABPS_PORTINACTIVE) *;deactivate Port* t) Repeat from d) with next values of "pdinlen", "pdoutlen" *;Test parameter*
Test parameter	pdinlen = {1, 32, 6, 0}, pdoutlen = {3, 32, 0, 20} *;variable PD lengths in octets*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock OD" 3) Check "PDInOut values" 4) Check "ArgBlock PDin"
Test passed	"MTU in OPERATE" = TRUE, and On-request Data = [0xAA, 0xBF, 0x77, 0x40, 0x55, 0xCD, 0x33, 0xE4], and PDout length = pdoutlen, all "PDOut values" = "0x5A", and PDin.InputDataLength = pdinlen, all PDin.PDIx = "0x5A"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	All M-sequence TYPEs (PD combinations): <ok | nok>

8.4.4 TYPE_2_V for different PD in/out and 32 octets OD

Table 208 defines the test conditions for this test case.

Table 208 – TYPE_2_V for different PD in/out and 32 octets OD

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0182
Name	TCM_DLOD_CYCC_TYPE2VPDXOD32
Purpose (short)	Master uses TYPE_2_V for different PD in/out and 32 octets OD
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.4, A.2.6, B.1.4, E.4, E.10, E.11, E.12, Table A.10
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master uses TYPE_2_V for several different PD in/out combinations and 32 octets OD. The Master receives the information about the target M-sequence type via address 0x03 (M-sequenceCapability), 0x05 (ProcessDataIn), and 0x06 (ProcessDataOut) out of DPP1. Different PD lengths are tested within a test procedure loop (TPL).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x2F *;see B.1.4 in [6]* b) MTU_ISDU_Add(64, 0, [0x00, …,0x00]) *;ISDU with values 32 x "0"* c) Assign first value to "pdinlen" and "pdoutlen" *;Test parameter* d) MTU_DPP1_Set(PDIn length) = "pdinlen" × 8 *;in Bits* e) MTU_DPP1_Set(PDOut length) = "pdoutlen" × 8 *;in Bits* f) MTU_PD_Set = [0x5A,…, "pdinlen"-times] *;set "PDin" in MTU* g) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* h) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* i) MTU_State_CheckOperate *;returns "MTU in OPERATE"* j) Evaluation 1) k) SMI_PDOut(ABPS_PDOUT<"pdoutlen",PDO0= 0x5A>) *;set "PDout"* l) SMI_DeviceWrite(ABPS_DEVICEWRITE<Index=64, OD=[ODstring]>) *;Test parameter* m) SMI_DeviceRead(ABPS_DEVICEREAD<Index=64>)*;returns "ArgBlock OD"* n) Evaluation 2) o) MTU_PD_Get *;returns "PDInOut values"* p) Evaluation 3) q) SMI_PDin *;returns "ArgBlock PDIn"* r) Evaluation 4) s) SMI_PortConfiguration(ABPS_PORTINACTIVE) *;deactivate Port* t) Repeat from d) with next values of "pdinlen", "pdoutlen" *;Test parameter*

Test parameter	pdinlen = {1, 32, 6, 0}, pdoutlen = {3, 32, 0, 20} *;variable PD lengths in octets*
	ODstring = "Hello World, this is Master Test"

Post condition	–

TEST CASE RESULTS	CHECK / REACTION	
Evaluation	For all TPL: 1) Check "MTU in OPERATE" 2) Check "ArgBlock OD"	3) Check "PDInOut values" 4) Check "ArgBlock PDin"
Test passed	"MTU in OPERATE" = TRUE, and On-request Data = ["Hello World, this is Master Test"], and PDout length = pdoutlen, all "PDOut values" = "0x5A", and PDin.InputDataLength = pdinlen, all PDin.PDIx = "0x5A"	
Test not passed (examples)	Any of the checks in Test passed failed or communication error	
Report	All M-sequence TYPEs (PD combinations): <ok | nok>	

8.4.5 Master sends Idle after an accomplished ISDU service

Table 209 defines the test conditions for this test case.

Table 209 – Master sends Idle after an accomplished ISDU service

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0288
Name	TCM_LGCY_MANY_IDLEAFTERISDU
Purpose (short)	Master sends ISDU request "IDLE 1" after an accomplished ISDU service
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master M-sequence test: test to pass
Specification (clause)	[6], 7.3.6.3, Figure 51 (Transition T8)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	After completion of an ISDU service, the Master shall transmit at least one "IDLE 1" request to the Device as an acknowledgement of the ISDU service (I-Service).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(MasterCycleTime) = 0xBF *;cycle time = 132.8ms* b) MTU_ISDU_Add(64, 0, [0xAA]) *;ISDU with length "1"* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* d) TM_AWAIT_PORT_STATUS(OPERATE) *;wait for OPERATE* e) MTU_ISDU_TrailingIdles *;reset "NumIdles"* f) SMI_DeviceRead(ABPS_DEVICEREAD <Index=64>) *;start second Read* g) SMI_DeviceRead(ABPS_DEVICEREAD<Index=64>) *;as soon as possible* h) MTU_ISDU_TrailingIdles *;return "NumIdles"* i) Evaluation 1)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check value of "NumIdles"
Test passed	"NumIdles" > 0
Test not passed (examples)	"NumIdles" = 0
Report	IDLE 1 message: <yes/no> <ok | nok>

8.5 STARTUP

8.5.1 Master reads communication parameters (Direct Parameter)

Table 210 defines the test conditions for this test case.

Table 210 – Master reads communication parameters (Direct Parameter)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0183
Name	TCM_DLST_CHCK_COMPARAM
Purpose (short)	Check that Master starts communication and reads communication parameters
Equipment under test (EUT)	Master
Test case version	1.3
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71, 72, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check correct start-up of Master comprising "V1.0" support. Master reads communication parameter address 0x02 to 0x06 (Direct Parameter page 1).
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "com" b) MTU_Timing_SetCommunicationMode = "com" c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_Startup_GetLog *;returning MessageLog* f) Evaluation 1) g) SMI_PortConfiguration(ABPS_PORTINACTIVE) h) Repeat from b) with next "com"
Test parameter	com = {COM1, COM2, COM3}
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog
Test passed	For all COMx: MessageLog shall fulfil the following requirements: - No communication retries - Master read messages to parameters on address 0x02 to 0x06 in ascending order - One or two Master read messages to parameter on address 0x02 permitted - Only one Master read message to each parameter on addresses 0x03 to 0x06
Test not passed (examples)	Any of the requirements not fulfilled at any COMx
Report	No communication retries: <ok | nok> Master read messages in ascending order: <ok | nok> Number of read accesses to address 0x02: <value> <ok | nok> Number of read accesses to addresses 0x03 to 0x06 <ok | nok> Communication modes COM1, COM2, COM3: <ok | nok> NOTE In this test case, a Master read on address 0x02 only once is tolerated; however, the Master-Tester-System shall display and/or print out a warning.

8.5.2 Master adjusts to protocol V1.1 (Direct Parameter)

Table 211 defines the test conditions for this test case.

Table 211 – Master adjusts to protocol V1.1 (Direct Parameter)

-CR096-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0184
Name	TCM_DLST_CHCK_VIDDID
Purpose (short)	Check whether Master adjusts to protocol V1.1 and reads VendorID and DeviceID
Equipment under test (EUT)	Master
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.2.3.3, Figures 71 to 75, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master adjusts to protocol revision V1.1 and reads VendorID and DeviceID. Check for the right decision with respect to the protocol revision.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog
Test passed	MessageLog shall contain after the sequence of reading parameter addresses 0x02 to 0x06 the following sequence of messages in this order: - Write MasterCommand 0x95 (Master_Ident) - Read VendorID - Read DeviceID
Test not passed (examples)	Master did not write MasterCommand 0x95
Report	MasterCommand 0x95: <yes/no> <ok | nok> Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok>

8.5.3 Master adjusts to protocol V1.0 (Direct Parameter)

Table 212 defines the test conditions for this test case.

Table 212 – Master adjusts to protocol V1.0 (Direct Parameter)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0185
Name	TCM_DLST_CHCK_V10VIDDID
Purpose (short)	Check whether Master adjusts to protocol V1.0 and reads VendorID and DeviceID
Equipment under test (EUT)	Master
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.2.3.3, Figures 71 to 75, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check whether Master adjusts to protocol revision V1.0 and reads VendorID and DeviceID. Check for the right decision with respect to the protocol revision.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(RevisionID) = 0x10 *;Set "Device" to revision V1.0*
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog
Test passed	MessageLog shall contain after the sequence of reading parameter addresses 0x02 to 0x06 the following sequence of messages in this order: - No Write MasterCommand 0x95 (Master_Ident) - Read VendorID - Read DeviceID
Test not passed (examples)	Master did not recognize the correct protocol version and wrote MasterCommand 0x95 and/or did not read the VendorID and DeviceID.
Report	No MasterCommand 0x95: <yes/no> <ok | nok> Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok>

8.5.4 Master start-up with non configured VID and DID

Table 213 defines the test conditions for this test case.

Table 213 – Master start-up with non configured VID and DID

-CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0186
Name	TCM_DLST_CHCK_NONCONFVIDDID
Purpose (short)	Check whether Master performs start-up with non-configured VID and DID
Equipment under test (EUT)	Master
Test case version	1.3
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.2.3.3, Figures 71 to 75, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master behavior. Master establishes communication with Device and turns it into OPERATE. Configured PortMode = IOL_AUTOSTART and Validation&Backup = No Device check.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	In 1) MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident), the following sequence of messages in this order: - Read VendorID - Read DeviceID - Read FunctionID - Write MasterCommand 0x9A *;Device Preoperate* In 2) PortStatusList shall show the following values: PortStatusList.PortStatusInfo = OPERATE PortStatusList.DeviceID = 0x002BD2
Test not passed (examples)	Master does not write MasterCommand 0x9A
Report	Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> MasterCommand 0x95: <yes/no> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.5.5 Master start-up with configured VID and DID

Table 214 defines the test conditions for this test case.

Table 214 – Master start-up with configured VID and DID

-CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0187
Name	TCM_DLST_CHCK_CONFVIDDID
Purpose (short)	Check Master start-up behavior with configured VendorID and DeviceID
Equipment under test (EUT)	Master
Test case version	1.4
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.2.3.3, Figures 71 to 75, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior. Master establishes communication with Device and turns it into OPERATE. Configured PortMode = IOL_MANUAL and Validation&Backup = Type compatible Device V1.1.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP) *;type compatibility* b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	In 1) MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident), the following sequence of messages in this order: - Read VendorID - Read DeviceID - Read FunctionID - Write MasterCommand 0x9A *;Device Preoperate* In 2) PortStatusList shall show the following values: PortStatusList.PortStatusInfo = OPERATE PortStatusList.DeviceID = 0x002BD2
Test not passed (examples)	Master does not write MasterCommand 0x9A, or values in PortStatusList are not correct
Report	Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> MasterCommand 0x95: <yes/no> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.5.6 Master start-up with overwrite of the DID (compatible)

Table 215 defines the test conditions for this test case.

Table 215 – Master start-up with overwrite of the DID (compatible)

-CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0188
Name	TCM_DLST_CHCK_OVERDIDOK
Purpose (short)	Check Master start-up behavior with overwrite of DeviceID (compatible)
Equipment under test (EUT)	Master
Test case version	1.6
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with overwrite of DeviceID (compatible). Master establishes communication with the MTU, detects incorrect "Device", overwrites the DeviceID with requested DeviceID, reads communication parameters again, and turns MTU into OPERATE. Configured PortMode = IOL_MANUAL and Validation&Backup = Type compatible Device V1.1. NOTE MTU shall tolerate Masters reading communication parameter address 0x02 (Direct Parameter page 1) twice.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(DeviceID) = 0x00A439
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP) *;type compatibility* b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident) the following sequence of messages in this order: Read VendorID → Read DeviceID → Read FunctionID → Write Revision = "0x11" → Write configured DeviceID (0x002BD2) → Write MasterCommand 0x96 → Read communication parameter on address 0x02 to 0x06 → Read again VendorID → Read again DeviceID → Read again FunctionID → Write MasterCommand 0x9A, and PortStatusList: PortStatusInfo = OPERATE, DeviceID = 0x002BD2
Test not passed (examples)	Master does not write MasterCommand 0x9A, or DeviceID of MTU ≠ 0x002BD2, or values in PortStatusList are not correct
Report	Master writes Revision 0x11: <yes/no> <ok | nok> MasterCommand 0x96: <yes/no> <ok | nok> Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> MasterCommand 0x9A: <yes/no> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.5.7 Master start-up with overwrite of the DID (incompatible)

Table 216 defines the test conditions for this test case.

Table 216 – Master start-up with overwrite of the DID (incompatible)

-CR049- -CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0189
Name	TCM_DLST_CHCK_OVERDIDNOK
Purpose (short)	Check Master start-up behavior with overwrite of the DeviceID (incompatible)
Equipment under test (EUT)	Master
Test case version	1.5
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with overwrite of the DeviceID. This test case supposes an incompatible DeviceID causing MTU to initiate a corresponding Event. Master starts communication with MTU, detects incorrect "Device", overwrites the DeviceID with the requested DeviceID, reads communication parameters again, and turns MTU into PREOPERATE. MTU shall tolerate Master reading communication parameter address 0x02 (Direct Parameter page 1) twice.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(DeviceID) = 0x00AAAA *;incompatible to any other DID*
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP) *;type compatibility* b) TM_AWAIT_PORT_STATUS((PORT_DIAG) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortEvent *;returning EventCode* f) Evaluation 2) g) SMI_PortStatus *;returning PortStatusList* h) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check EventCode 3) Check PortStatusList
Test passed	MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident), the following sequence of messages in this order: Read VendorID → Read DeviceID → Read FunctionID → Write Revision=0x11 → Write configured DeviceID → Write MasterCommand 0x96 → Read communication parameter on address 0x02 to 0x06 → Read again VendorID → Read again DeviceID → Read again FunctionID → Write MasterCommand 0x9A → No Write MasterCommand 0x99, and PortEvent: EventCode = 0x1803, and *;"Incorrect DeviceID"* PortStatusList: PortStatusInfo = PORT_DIAG, DeviceID = 0x00AAAA, DiagEntry[any].EventCode = 0x1803 *;"Incorrect DeviceID"*
Test not passed (examples)	No PORT_DIAG, or MasterCommand 0x9A and thereafter a MasterCommand 0x99
Report	Master writes Revision 0x11: <yes/no> <ok | nok> Master writes configured DeviceID: <yes/no> <ok | nok> MasterCommand 0x96: <yes/no> <ok | nok> Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> MasterCommand 0x9A: <yes/no> <ok | nok> Master reads FunctionID: <value> <ok | nok> No MasterCommand 0x99: <yes/no> <ok | nok>

TEST CASE RESULTS	CHECK / REACTION
	Master indicates Event: <EventCode> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.5.8 Master start-up with overwrite of the RID (incompatible)

Table 217 defines the test conditions for this test case.

Table 217 – Master start-up with overwrite of the RID (incompatible)

-CR050- -CR044- -CR060- -CR100-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0190
Name	TCM_DLST_CHCK_OVERRIDNOK
Purpose (short)	Check Master start-up behavior with overwrite of the RevisionID (incompatible)
Equipment under test (EUT)	Master
Test case version	1.6
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with overwrite of RevisionID (RID). This test case supposes an incompatible RevisionID (>V1.1). Master starts communication with MTU, detects the "incorrect" RevisionID, and overwrites the RevisionID with the requested RevisionID. MTU in turn restores the original RID causing a corresponding Event. MTU shall tolerate Master reading communication parameter address 0x02 (Direct Parameter page 1) twice.
Precondition	EUT: PORT_INACTIVE (IOL_MANUAL, Type compatible Device V1.1) MTU: MTU_STANDARD_STATE with MTU_DPP1_Set(DeviceID) = 0x00A439, and MTU_DPP1_Set(RevisionID) = 0x12 *;incorrect RevisionID (MTU does not* *;accept any other protocol revision in this state)*
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP) *;type compatibility* b) TM_AWAIT_PORT_STATUS(PORT_DIAG)) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortEvent *;returning EventCode* f) Evaluation 2) g) SMI_PortStatus *;returning PortStatusList* h) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check EventCode 2) Check PortStatusList
Test passed	MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident), the following sequence of messages in this order: Write Revision = "0x11" → Write configured DeviceID → Write MasterCommand 0x96 → Read communication parameter on address 0x02 to 0x06 → MasterCommand 0x95 (Master_Ident) → Write MasterCommand 0x9A → No Write MasterCommand 0x99, and *;Device Operate* PortEvent: EventCode = 0x1801, and *;"Startup Parameterization error"* PortStatusList: PortStatusInfo = PORT_DIAG, DeviceID = 0x00A439, RevisionID = 0x12, DiagEntry[any].EventCode = 0x6001 *;"Startup parametrization error"*
Test not passed (examples)	Master did not send MasterCommand 0x9A, or Master leaves PREOPERATE state.

TEST CASE RESULTS	CHECK / REACTION	
Report	Master writes Revision 0x11: <yes/no> Master writes configured DeviceID: <yes/no> MasterCommand 0x96: <yes/no> MasterCommand 0x9A: <yes/no> No MasterCommand 0x99: <yes/no> Master indicates Event: <EventCode> Correct values in PortStatusList: <yes/no>	<ok | nok> <ok | nok> <ok | nok> <ok | nok> <ok | nok> <ok | nok> <ok | nok>

8.5.9 Master start-up with non configured VID and DID (V1.0)

Table 218 defines the test conditions for this test case.

Table 218 – Master start-up with non configured VID and DID (V1.0)

-CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0192
Name	TCM_DLST_CHCK_VIDDIDNONCONFIG
Purpose (short)	Check Master start-up behavior with non-configured VID and DID (V1.0)
Equipment under test (EUT)	Master
Test case version	1.5
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with non-configured VID and DID for a legacy Device (V1.0). The Master establishes communication with the "Device" (MTU), writes the MasterCycleTime (address 0x01 in Direct Parameter page 1), and sends MasterCommand 0x99 (OPERATE) turning the "Device" from STARTUP to OPERATE. Configured PortMode = IOL_AUTOSTART and Validation&Backup = No Device check.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(RevisionID) = 0x10 *;Revision V1.0*
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;no type check* b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	MessageLog shall contain after read of parameter address 0x02 to 0x06 the following sequence of messages in this order: Read VendorID → Read DeviceID → Read FunctionID → Write MasterCycleTime → Write MasterCommand 0x99 PortStatusList: PortStatusInfo = OPERATE, DeviceID = 0x002BD2
Test not passed (examples)	Master does not write the MasterCycleTime (address 0x01) or does not send MasterCommand 0x99
Report	Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> MasterCycleTime written: <yes/no> <ok | nok> MasterCommand 0x99: <yes/no> <ok | nok>

8.5.10 Master start-up with configured VID and DID (Device V1.0)

Table 219 defines the test conditions for this test case.

Table 219 – Master start-up with configured VID and DID (V1.0)

-CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0193
Name	TCM_DLST_CHCK_VIDDIDCONFIG
Purpose (short)	Check Master start-up behavior with configured VID and DID (Device V1.0)
Equipment under test (EUT)	Master
Test case version	1.5
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with configured VID and DID for a legacy "Device" (V1.0). Master establishes communication with the "Device" (MTU) and writes the MasterCycleTime (address 0x01 in Direct Parameter page 1) only if the configured values match the values in the "Device". Then, the Master shall turn the "Device" from STARTUP into OPERATE via MasterCommand 0x99. Configured PortMode = IOL_MANUAL and Validation&Backup = Type compatible Device V1.0
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(RevisionID) = 0x10 *;"Device" revision V1.0*
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP< Validation&Backup= TYPE_compatible_Device_V1.0>) *;type compatibility* b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	MessageLog shall contain after read of parameter address 0x02 to 0x06 the following sequence of messages in this order: Read VendorID → Read DeviceID → Read FunctionID → Write MasterCycleTime → Write MasterCommand 0x99 PortStatusList: PortStatusInfo = OPERATE, DeviceID = 0x002BD2
Test not passed (examples)	Master does not write the MasterCycleTime (address 0x01) or does not send MasterCommand 0x99
Report	Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> MasterCycleTime written: <yes/no> <ok | nok> MasterCommand 0x99: <yes/no> <ok | nok>

8.5.11 Master start-up with incorrect DID (Device V1.0)

Table 220 defines the test conditions for this test case.

Table 220 – Master start-up with incorrect DID (V1.0)

-CR101- -CR044-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0194
Name	TCM_DLST_CHCK_DIDWRONG
Purpose (short)	Check Master start-up behavior with incorrect DID (Device V1.0)
Equipment under test (EUT)	Master
Test case version	1.5
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with incorrect configured DID for a legacy "Device" (V1.0). Master establishes communication with MTU, recognizes a deviating DeviceID, and indicates a corresponding Event. Configured PortMode = IOL_MANUAL and Validation&Backup = Type compatible Device V1.0
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE MTU_DPP1_Set(RevisionID) = 0x10 *;Revision V1.0* MTU_DPP1_Set(DeviceID) = 0x00AAAA *;incompatible to any other DID*
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP<Validation&Backup= TYPE_compatible_Device_V1.0>) *;type compatibility* b) TM_AWAIT_PORT_STATUS(PORT_DIAG) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortEvent *;returning EventCode* f) Evaluation 2) g) SMI_PortStatus *;returning PortStatusList* h) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check EventCode 2) Check PortStatusList
Test passed	MessageLog shall contain after read of parameter address 0x02 to 0x06, the following sequence of messages in this order: Read VendorID -> Read DeviceID -> Read FunctionID -> Write MasterCycleTime -> Write MasterCommand 0x99 -> No Write MasterCommand 0x98 PortEvent: EventCode = 0x1803 *;"Incorrect DeviceID"* PortStatusList: PortStatusInfo = PORT_DIAG, DeviceID = 0x00AAAA, DiagEntry[any].EventCode = 0x1803 *;"Incorrect DeviceID"*
Test not passed (examples)	Master indicates no system specific fault information or sends MasterCommand 0x98.
Report	Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> No MasterCommand 0x98: <yes/no> <ok | nok> Master indicates Event: <EventCode> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.5.12 Master start-up with overwrite of the RID (compatible)

Table 221 defines the test conditions for this test case.

Table 221 – Master start-up with overwrite of the RID (compatible)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0307
Name	TCM_DLST_CHCK_OVERRIDOK
Purpose (short)	Check Master start-up behavior with overwrite of the Device RID (compatible)
Equipment under test (EUT)	Master
Test case version	1.3
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.2, 9.3.3, 10.6.3, Figures 71 to 76, and 81
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Check Master start-up behavior with overwrite of the RevisionID (compatible). This test supposes the active switching to protocol revision 1.0. Master overwrites the RID and the "Device" accepts the requested protocol revision. Configured PortMode = IOL_MANUAL and Validation&Backup = Type compatible Device V1.0 NOTE MTU shall tolerate Master reading communication parameter address 0x02 (Direct Parameter page 1) twice.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE *;MTU accepts protocol revision 1.0 in this state*
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP, RevisionID = "CRID") (Test param.) b) TM_AWAIT_PORT_STATUS(OPERATE) c) MTU_Startup_GetLog *;returning MessageLog* d) Evaluation 1) e) SMI_PortStatus *;returning PortStatusList* f) Evaluation 2)
Test parameter	CRID = 0x10 *;configured RevisionID V1.0*
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check MessageLog 2) Check PortStatusList
Test passed	MessageLog shall contain after writing MasterCommand 0x95 (Master_Ident) the following sequence of messages in this order: Write RevisionID = "0x10" → Write DeviceID → Write MasterCommand 0x96 → Read communication parameter on address 0x02 to 0x06 → Read VendorID → Read DeviceID → Read FunctionID Write MasterCommand 0x99 *;DeviceOperate* PortStatusList: PortStatusInfo = OPERATE, DeviceID = 0x002BD2, RevisionID = "CRID"
Test not passed (examples)	Master writes MasterCommand 0x9A or MTU does not show RevisionID 0x10
Report	Master writes RevisionID 0x10: <yes/no> <ok | nok> Master writes configured DeviceID: <yes/no> <ok | nok> MasterCommand 0x96: <yes/no> <ok | nok> Master reads DPP1 parameters 0x02 to 0x06: <yes/no> <ok | nok> Master reads VendorID: <value> <ok | nok> Master reads DeviceID: <value> <ok | nok> Master reads FunctionID: <value> <ok | nok> No MasterCommand 0x9A: <yes/no> <ok | nok> Correct values in PortStatusList: <yes/no> <ok | nok>

8.6 PREOPERATE

8.6.1 Master in PREOPERATE uses TYPE_0 to read Index 24 (1 OD)

Table 222 defines the test conditions for this test case.

Table 222 – Master in PREOPERATE uses TYPE_0 to read Index 24 (1 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0202
Name	TCM_DLOD_PREP_TYPE0READOD1
Purpose (short)	Master uses TYPE_0 to read Index 24 in PREOPERATE (1 octet OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.2, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_0 with 1 octet OD shall be used for Read access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x01 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) MTU_ISDU_Add(Index = 24, Subindex = 0, Value = testdata) *;Test parameter* f) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* g) Evaluation 1) h) SMI_DeviceRead(ABPS_DEVICEREAD(Index = 24, Subindex = 0)) *;returns "OD"* i) Evaluation 2) j) MTU_State_CheckOperate *;returns "MTU in OPERATE"* k) Evaluation 3)
Test parameter	testdata = "0x74, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x6D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x6F, 0x64, 0x64, 0x61"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check "OD" 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = testdata, and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.2 Master in PREOPERATE uses TYPE_1_2 to read Index 24 (2OD)

Table 223 defines the test conditions for this test case.

Table 223 – Master in PREOPERATE uses TYPE_1_2 to read Index 24 (2OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0203
Name	TCM_DLOD_PREP_TYPE12READOD2
Purpose (short)	Master uses TYPE_1_2 to read Index 24 in PREOPERATE (2 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_1-2 with 2 octets OD shall be used for Read access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* [6] b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) MTU_ISDU_Add(Index = 24, Subindex = 0, Value = testdata) *;Test parameter* f) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* g) Evaluation 1) h) SMI_DeviceRead(ABPS_DEVICEREAD(Index = 24, Subindex = 0)) *;returns "OD"* i) Evaluation 2) j) MTU_State_CheckOperate *;returns "MTU in OPERATE"* k) Evaluation 3)
Test parameter	testdata = "0x74, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x6D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x6F, 0x64, 0x64, 0x61"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check "OD" 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = testdata, and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.3 Master in PREOPERATE uses TYPE_1_V to read Index 24 (8 OD)

Table 224 defines the test conditions for this test case.

Table 224 – Master in PREOPERATE uses TYPE_1_V to read Index 24 (8 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0204
Name	TCM_DLOD_PREP_TYPE1VREADOD8
Purpose (short)	Master uses TYPE_1_V to read Index 24 in PREOPERATE (8 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_0_V with 8 octets OD shall be used for Read access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x21 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) MTU_ISDU_Add(Index = 24, Subindex = 0, Value = testdata) f) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* g) Evaluation 1) h) SMI_DeviceRead(ABPS_DEVICEREAD(Index = 24, Subindex = 0)) *;returns "OD"* i) Evaluation 2) j) MTU_State_CheckOperate *;returns "MTU in OPERATE"* k) Evaluation 3)
Test parameter	testdata = "0x74, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x6D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x6F, 0x64, 0x64, 0x61"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check "OD" 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = testdata, and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.4 Master in PREOPERATE uses TYPE_1_V to read Index 24 (32 OD)

Table 225 defines the test conditions for this test case.

Table 225 – Master in PREOPERATE uses TYPE_1_V to read Index 24 (32 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0205
Name	TCM_DLOD_PREP_TYPE1VREADOD32
Purpose (short)	Master uses TYPE_1_V to read Index 24 in PREOPERATE (32 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_1_V with 32 octets OD shall be used for Read access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x31 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) MTU_ISDU_Add(Index = 24, Subindex = 0, Value = testdata) f) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* g) Evaluation 1) h) SMI_DeviceRead(ABPS_DEVICEREAD(Index = 24, Subindex = 0)) *;returns "OD"* i) Evaluation 2) j) MTU_State_CheckOperate *;returns "MTU in OPERATE"* k) Evaluation 3)
Test parameter	testdata = "0x74, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x6D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x6F, 0x64, 0x64, 0x61"
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check "OD" 2) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = testdata, and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.5 Master in PREOPERATE uses TYPE_0 to write Index 24 (1 OD)

Table 226 defines the test conditions for this test case.

Table 226 – Master in PREOPERATE uses TYPE_0 to write Index 24 (1 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0206
Name	TCM_DLOD_PREP_TYPE0WRITEOD1
Purpose (short)	Master uses TYPE_0 to write to Index 24 in PREOPERATE (1 octet OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.2, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_0 with 1 octet OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x01 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* f) Evaluation 1) g) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74]) *;preset data* h) SMI_DeviceWrite(ABPS_DEVICEWRITE <Index=24, Subindex = 0, Value = [0x54]>) *;writes test data* i) MTU_ISDU_Read(Index = 24, Subindex = 0) *;returns "OD"* j) Evaluation 2) k) MTU_State_CheckOperate *;returns "MTU in OPERATE"* l) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check OD 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = [0x54], and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.6 Master in PREOPERATE uses TYPE_1_2 to write Index 24 (2 OD)

Table 227 defines the test conditions for this test case.

Table 227 – Master in PREOPERATE uses TYPE_1_2 to write Index 24 (2 OD)

-CR055-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0207
Name	TCM_DLOD_PREP_TYPE12WRITEOD2
Purpose (short)	Master uses TYPE_1_2 to write to Index 24 in PREOPERATE (2 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_1_2 with 2 octets OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* f) Evaluation 1) g) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74, 0x75]) *;preset data* h) SMI_DeviceWrite(ABPS_DEVICEWRITE <Index=24, Subindex = 0, Value = [0x54, 0x65]>) *;writes test data* i) MTU_ISDU_Read(Index = 24, Subindex = 0) *;returns "OD"* j) Evaluation 2) k) MTU_State_CheckOperate *;returns "MTU in OPERATE"* l) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check OD 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = [0x54, 0x65], and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.7 Master in PREOPERATE uses TYPE_1_V to write Index 24 (8 OD)

Table 228 defines the test conditions for this test case.

Table 228 – Master in PREOPERATE uses TYPE_1_V to write Index 24 (8 OD)

-CR055-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0208
Name	TCM_DLOD_PREP_TYPE1VWRITEOD8
Purpose (short)	Master uses TYPE_1_V to write to Index 24 in PREOPERATE (8 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_1_V with 8 octets OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x21 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) ;invalid VendorID e) TM_AWAIT_PORT_STATUS(PORT_DIAG) ;returns "Reached" f) Evaluation 1) g) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74, 0x75, 0x76, 0x77, 0x78, 0x79, 0x7A, 0x7B]) *;preset data* h) SMI_DeviceWrite(ABPS_DEVICEWRITE <Index=24, Subindex = 0, Value = [0x54, 0x65, 0x73, 0x74, 0x20, 0x38, 0x4F, 0x44]>) ;writes test data i) MTU_ISDU_Read(Index = 24, Subindex = 0) ;returns "OD" j) Evaluation 2) k) MTU_State_CheckOperate ;returns "MTU in OPERATE" l) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check OD 3) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = [0x54, 0x65, 0x73, 0x74, 0x20, 0x38, 0x4F, 0x44], and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.6.8 Master in PREOPERATE uses TYPE_1_V to write Index 24 (32 OD)

Table 229 defines the test conditions for this test case.

Table 229 – Master in PREOPERATE TYPE_1_V to write Index 24 (32 OD)

-CR055-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0209
Name	TCM_DLOD_PREP_TYPE1VWRITEOD32
Purpose (short)	Master uses TYPE_1_V to write to Index 24 in PREOPERATE (32 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.8, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for PREOPERATE. TYPE_1_V with 32 octets OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedureo	a) MTU_DPP1_Set(M-sequenceCapability) = 0x31 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength) = 0 c) MTU_DPP1_Set(OutputLength) = 0 d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns "Reached"* f) Evaluation 1) g) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74, 0x75, 0x76, 0x77, 0x78, 0x79, 0x7A, 0x7B, 0x7C, 0x7D, 0x7E, 0x7F, 0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8A, 0x8B, 0x8C, 0x8D, 0x8E, 0x8F, 0x90, 0x91, 0x92, 0x93]) *;preset data* h) SMI_DeviceWrite(ABPS_DEVICEWRITE <Index=24, Sub-index = 0, Value = [0x54, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x4D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x4F, 0x44, 0x44, 0x41]>) *;writes test data* i) MTU_ISDU_Read(Index = 24, Subindex = 0) *;returns "OD"* j) Evaluation 2) k) MTU_State_CheckOperate *;returns "MTU in OPERATE"* l) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Reached" 2) Check OD 2) Check "MTU in OPERATE"
Test passed	"Reached" = TRUE, and "OD" = [0x54, 0x65, 0x73, 0x74, 0x20, 0x6F, 0x66, 0x20, 0x74, 0x68, 0x65, 0x20, 0x4D, 0x61, 0x73, 0x74, 0x65, 0x72, 0x73, 0x20, 0x77, 0x69, 0x74, 0x68, 0x20, 0x33, 0x32, 0x20, 0x4F, 0x44, 0x44, 0x41], and "MTU in OPERATE" = FALSE
Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.7 OPERATE

8.7.1 Master in OPERATE uses TYPE_1_V for zero bit PD in/out (8 OD)

Table 230 defines the test conditions for this test case.

Table 230 – Master in OPERATE uses TYPE_1_V for zero bit PD in/out (8 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0336
Name	TCM_ DLPD_CYCC_TYPE1V_OD8BIT0INBIT0OUT
Purpose (short)	Master uses M-sequence TYPE_1_V for zero bit PD in/out (8 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 3)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Master reads OD length and PD input/output length during STARTUP through DPP1 from Device. Due to these values, the Master shall select M-sequence TYPE_1_V in OPERATE and provide this information to SMI services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x1D *;see B.1.4 in [6]* [6] b) MTU_DPP1_Set(PDIn length) = 0x00 *;"0" Bit* c) MTU_DPP1_Set(PDOut length) = 0x00 *;"0" Bit* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;switch to OPERATE* e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_State_CheckOperate *;returns "MTU in OPERATE"* g) Evaluation 1) h) SMI_PDIn *;returns "InputDataLength"* i) Evaluation 2) j) SMI_PDInOut *;returns "InputDataLength", "OutputDataLength"* k) Evaluation 3) l) SMI_PortStatus *;returns "RevisionID", "InputDataLength", "OutputDataLength"* m) Evaluation 4) n) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check value of PDIn.InputDataLength 3) Check value of PDInOut.InputDataLength and PDInOut.OutputDataLength 4) Check value of PortStatus.RevisionID, PortStatus.InputDataLength, and Portstatus.OutputDataLength 5) Check "vendorname"
Test passed	"MTU in OPERATE" = TRUE, and PDIn: InputDataLength = "0", and PDInOut: InputDataLength = "0", OutputDataLength = "0", and PortStatus: RevisionID = 0x11, InputDataLength = "0", OutputDataLength = "0", and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	vendorname ok: <yes/no> <ok | nok>

8.7.2 Master in OPERATE uses TYPE_1_V for zero bit PD in/out (32 OD)

Table 231 defines the test conditions for this test case.

Table 231 – Master in OPERATE uses TYPE_1_V for zero bit PD in/out (32 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0337
Name	TCM_ DLPD_CYCC_TYPE1V_OD32BIT0INBIT0OUT
Purpose (short)	Master uses M-sequence TYPE_1_V for zero bit PD in/out (32 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master protocol test: test to pass
Specification (clause)	[6], 9.2.3.5, 11.2.7, 11.2.17, 11.2.18, 11.2.19, A.2.6, E.4, E.10, E.11, E.12, Table A.10 (row 4)
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	The Master reads OD length and PD input/output length during STARTUP through DPP1 from Device. Due to these values, the Master shall select M-sequence TYPE_1_V in OPERATE and provide this information to SMI services.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x1F ;see B.1.4 in [6] b) MTU_DPP1_Set(PDIn length) = 0x00 ;"0" Bit c) MTU_DPP1_Set(PDOut length) = 0x00 ;"0" Bit d) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) ;switch to OPERATE e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_State_CheckOperate ;returns "MTU in OPERATE" g) Evaluation 1) h) SMI_PDIn ;returns "InputDataLength" i) Evaluation 1) j) SMI_PDInOut ;returns "InputDataLength", "OutputDataLength" k) Evaluation 2) l) SMI_PortStatus ;returns "RevisionID", "InputDataLength", "OutputDataLength" m) Evaluation 3) n) SMI_DeviceRead(ABPS_DEVICEREAD <vendorname>)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check value of PDIn.InputDataLength 3) Check value of PDInOut.InputDataLength and PDInOut.OutputDataLength 4) Check value of PortStatus.RevisionID, PortStatus.InputDataLength, and Portstatus.OutputDataLength 5) Check "vendorname"
Test passed	"MTU in OPERATE" = TRUE, and PDIn: InputDataLength = "0", and PDInOut: InputDataLength = "0", OutputDataLength = "0", and PortStatus: RevisionID = 0x11, InputDataLength = "0", OutputDataLength = "0", and <vendorname> = "IO-Link Community"
Test not passed (examples)	Any of the checks in Test passed failed or communication error
Report	vendorname ok: <yes/no> <ok | nok>

8.7.3 Master in OPERATE uses TYPE_0 to read Index 24 (1 OD)

Table 232 defines the test conditions for this test case.

Table 232 – Master in OPERATE uses TYPE_0 to read Index 24 (1 OD)

-CR107-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0210
Name	TCM_DLOD_OPER_TYPE0READOD1
Purpose (short)	Master uses TYPE_0 to read Index 24 in OPERATE (1 octet OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.2, A.5.2, B.2.1, Tables A.10, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for OPERATE. TYPE_0 with 1 octet OD shall be used for Read access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x01 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength = 0) c) MTU_DPP1_Set(OutputLength = 0) d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74]) *;test data* f) TM_AWAIT_PORT_STATUS(OPERATE) g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_DeviceRead(ABPS_DEVICEREAD(Index = 24, Subindex = 0) *;returns "OD"* j) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "MTU in OPERATE" 2) Check "OD"
Test passed	"MTU in OPERATE" = TRUE "OD" = [0x74]
Test not passed (examples)	Any evaluation failed or communication eror
Report	All evaluations <ok | nok>

8.7.4 Master in OPERATE uses TYPE_0 to write Index 24 (1 OD)

Table 233 defines the test conditions for this test case.

Table 233 – Master in OPERATE uses TYPE_0 to write Index 24 (1 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0211
Name	TCM_DLOD_OPER_TYPE0WRITEOD1
Purpose (short)	Master uses TYPE_0 to write to Index 24 in OPERATE (1 octet OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.2, A.5.2, B.2.1, Tables A.10, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for OPERATE. TYPE_0 with 1 octet OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x01 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength = 0) c) MTU_DPP1_Set(OutputLength = 0) d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74]) *;preset test data* f) TM_AWAIT_PORT_STATUS(OPERATE) g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_DeviceWrite(ABPS_DEVICEWRITE<Index=24, Subindex=0, Value=[0x54]>) *;write test data* j) MTU_ISDU_Read(Index = 24, Subindex = 0) *;returns "OD"* k) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "MTU in OPERATE" 2) Check "OD"
Test passed	"MTU in OPERATE" = TRUE "OD" = [0x54]
Test not passed (examples)	Any evaluation failed or communication eror
Report	All evaluations <ok | nok>

8.7.5 Master in OPERATE uses TYPE_1_2 to write Index 24 (2 OD)

Table 234 defines the test conditions for this test case.

Table 234 – Master in OPERATE uses TYPE_1_2 to write Index 24 (2 OD)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0212
Name	TCM_DLOD_OPER_TYPE12WRITEOD2
Purpose (short)	Master uses TYPE_1_2 to write to Index 24 in OPERATE (2 octets OD)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master protocol test: test to pass
Specification (clause)	[6], A.2.3, A.5.2, B.2.1, Tables A.10, A.13
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master uses correct M_sequence type according to Device settings in DPP1 for OPERATE. TYPE_1_2 with 2 octet OD shall be used for Write access.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x11 *;see B.1.4 in [6]* b) MTU_DPP1_Set(InputLength = 0) c) MTU_DPP1_Set(OutputLength = 0) d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) MTU_ISDU_Add(Index = 24, Subindex = 0, OD = [0x74]) *;preset test data* f) TM_AWAIT_PORT_STATUS(OPERATE) g) MTU_State_CheckOperate *;returns "MTU in OPERATE"* h) Evaluation 1) i) SMI_DeviceWrite(ABPS_DEVICEWRITE <Index=24, Subindex=0, Value=[0x54, 0x65]>) *;write test data* j) MTU_ISDU_Read(Index = 24, Subindex = 0) *;returns "OD"* k) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "MTU in OPERATE" 2) Check "OD"
Test passed	"MTU in OPERATE" = TRUE "OD" = [0x54, 0x65]
Test not passed (examples)	Any evaluation failed or communication eror
Report	All evaluations <ok | nok>

8.8 Fallback

8.8.1 Fallback from PREOPERATE

Table 235 defines the test conditions for this test case.

Table 235 – Fallback from PREOPERATE

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0213
Name	TCM_DLFB_PROP_OK
Purpose (short)	Master Fallback from PREOPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.2.3, Figure 34, 11.3.2, Figure 101, 9.2.1, Figure 69, 9.2.2.2, 5.2.2.1, Table B.2
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Transition from PREOPERATE to SIO mode of Master Port via reconfiguration and check of its correct behavior afterwards. That means, Master communicates with Device and receives only a switching signal (DI) after reconfiguration. Hint: Master reconfiguration with a different VendorID than permitted for the Device causes Master to hold on in PREOPERATE.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID=1>) *;invalid VendorID* b) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* c) Evaluation 1) d) MTU_State_CheckPreoperate *;returns "MTU in PREOPERATE"* e) Evaluation 2) f) SMI_PortConfiguration(ABPS_PORTTODI) *;switch to DI mode* g) MTU_State_CheckFallback *;returns "Master sent Fallback"* h) Evaluation 3) i) TM_AWAIT_DI_HIGH *;PDIn.PDI0="1" within MTU_Timeout* j) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "MTU in PREOPERATE" 3) Check "Master sent Fallback" 4) Check MTU_Timeout
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and "MTU in PREOPERATE" = TRUE, and "Master sent Fallback" = TRUE, and No MTU_Timeout
Test not passed (examples)	Any evaluation failed or MTU_Timeout
Report	All evaluations: <ok | nok>

8.8.2 Fallback from PREOPERATE fails

Table 236 defines the test conditions for this test case.

Table 236 – Fallback from PREOPERATE fails

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0214
Name	TCM_DLFB_PROP_FAILS
Purpose (short)	Master Fallback from PREOPERATE fails
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.2.3, 7.3.2.4, Figure 34, Table B.2
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Transition from PREOPERATE to SIO mode of Master Port via reconfiguration and check of its correct behavior afterwards. It is assumed that reconfiguration fails and Master iniates a Port restart or stops communicating. Hint: Master reconfiguration with a different VendorID than permitted for the Device causes Master to hold on in PREOPERATE.
Precondition	EUT: PORT_INACTIVE MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DPP1_Set(M-sequenceCapability) = 0x21 *;see B.1.4 in [6]* b) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* c) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "MTU in PREOPERATE"* f) Evaluation 2) g) MTU_State_BlockFallback(*n*=2) *;2 × refuse Fallback acknowledge* h) SMI_PortConfiguration(ABPS_PORTTODI) *;switch to DI mode* i) MTU_State_CheckFallback *;returns "Master sent Fallback"* j) Evaluation 3) k) TM_AWAIT_DI_HIGH *;PDIn.PDI0="1" within Test_Timeout* l) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "MTU in PREOPERATE" 3) Check "Master sent Fallback" 4) Check Test_Timeout
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and "MTU in PREOPERATE" = TRUE, and "Master sent Fallback" = TRUE, and No Test_Timeout
Test not passed (examples)	Any evaluation failed
Report	All evaluations: <ok | nok>

8.8.3 Fallback from OPERATE

Table 237 defines the test conditions for this test case.

Table 237 – Fallback from OPERATE

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0215
Name	TCM_DLFB_OPER_OK
Purpose (short)	Master Fallback from OPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.2.3, 7.3.2.4, Figure 34, Table B.2
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Transition from OPERATE to SIO mode of Master Port via reconfiguration and check of its correct behavior afterwards. That means, Master communicates with Device and receives only a switching signal (DI) after reconfiguration.
Precondition	EUT: PORT_AUTOSTART MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) SMI_PortConfiguration(ABPS_PORTTODI) *;switch to DI mode* d) MTU_State_CheckFallback *;returns "Master sent Fallback"* e) Evaluation 2) f) TM_AWAIT_DI_HIGH *;PDIn.PDI0 = 1 within MTU_Timeout* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "MTU in OPERATE" 2) Check "Master sent Fallback" 3) Check MTU_Timeout
Test passed	"MTU in OPERATE" = TRUE, and "Master sent Fallback" = TRUE, and No MTU_Timeout
Test not passed (examples)	Any evaluation failed or MTU_Timeout
Report	All evaluations: <ok | nok>

8.8.4 Fallback from OPERATE fails

Table 238 defines the test conditions for this test case.

Table 238 – Fallback from OPERATE fails

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0216
Name	TCM_DLFB_OPER_FAILS
Purpose (short)	Master Fallback from OPERATE fails
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.2.3, 7.3.2.4, Figure 34, Table B.2
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Transition from OPERATE to SIO mode of Master Port via reconfiguration and check of its correct behavior afterwards. That means, Master shall send two times the Fallback command after the reception of a second invalid response. Upon receipt of the invalid response, Master iniates a Port restart or stops communicating.
Precondition	EUT: PORT_AUTOSTART MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_State_BlockFallback(*n*=2) *;2 × refuse Fallback acknowledge* d) SMI_PortConfiguration(ABPS_PORTTODI) e) MTU_State_CheckFallback *;returns "Master sent Fallback"* f) Evaluation 2) g) TM_AWAIT_DI_HIGH *;PDIn.PDI0="1" within Test__Timeout* h) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "MTU in OPERATE" 2) Check "Master sent Fallback" 3) Check Test_Timeout
Test passed	"MTU in OPERATE" = TRUE, and "Master sent Fallback" = TRUE, and No Test_Timeout
Test not passed (examples)	Any evaluation failed
Report	All evaluations: <ok | nok>

8.9 Retry

8.9.1 Retries after 2 reply messages with incorrect Checksum (STARTUP)

Table 239 defines the test conditions for this test case.

Table 239 – Retries after 2 reply messages with incorrect Checksum (STARTUP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0338
Name	TCM_DLCC_RTRY_CKSINCORR2STRTUP
Purpose (short)	Retry behavior after 2 reply messages with incorrect checksum in STARTUP
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at STARTUP. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is correct.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) MTU_State_IncorrectChecksums(*n*=2) *;2 x reply with incorrect checksum* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_State_CountRestarts *;returns "Restarts"* f) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Restarts"
Test passed	"Restarts" = 1
Test not passed (examples)	Any evaluation failed or communication error or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.2 Retries after 2 reply messages with incorrect Checksum (PREOP)

Table 240 defines the test conditions for this test case.

Table 240 – Retries after 2 reply messages with incorrect Checksum (PREOP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0339
Name	TCM_DLCC_RTRY_ CKSINCORR2PREOP
Purpose (short)	Retry behavior after 2 reply messages with incorrect checksum in PREOPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at PREOPERATE. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is correct.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* c) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) ef) MTU_State_IncorrectChecksum(*n*=2) *;2 x reply with incorrect checksum* f) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;returns "VendorName"* g) Evaluation 2) h) MTU_State_CheckPreoperate *;returns "Reached"* i) Evaluation 3) j) MTU_State_CountRestarts *;returns "Restarts"* k) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "VendorName" 3) Check "Reached" 4) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and "VendorName" = IO-Link Community, and "Reached" = TRUE, and "Restarts" = 1
Test not passed (examples)	Any evaluation failed, or SMI_DeviceRead with negative response, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.3 Retries after 2 reply messages with incorrect Checksum (OPERATE)

Table 241 defines the test conditions for this test case.

Table 241 – Retries after 2 reply messages with incorrect Checksum (OPERATE)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0340
Name	TCM_DLCC_RTRY_ CKSINCORR2OPERATE
Purpose (short)	Retry behavior after 2 reply messages with incorrect checksum in OPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at OPERATE. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is correct.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_IncorrectChecksum(*n*=2) *;2 x reply with incorrect checksum* f) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;returns "VendorName"* g) Evaluation 2) h) MTU_State_CountRestarts *;returns "Restarts"* i) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "VendorName" 3) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and "VendorName" = IO-Link Community, and "Restarts" = 1
Test not passed (examples)	Any evaluation failed, or SMI_DeviceRead with negative response, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.4 Retries after 3 reply messages with incorrect Checksum (STARTUP)

Table 242 defines the test conditions for this test case.

Table 242 – Retries after 3 reply messages with incorrect Checksum (STARTUP)

TEST CASE ATTRIBUTES	**IDENTIFICATION / REFERENCE**
Identification (ID)	SDCI_TC_0341
Name	TCM_DLCC_RTRY_CKSINCORR3STRUP&RESTOP
Purpose (short)	Retry behavior after 3 reply messages with incorrect CKS at STARTUP (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at STARTUP. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is still incorrect.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) MTU_State_IncorrectChecksum(*n*=3) *;3 x reply with incorrect checksum* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* d) TM_AWAIT_PORT_STATUS (OPERATE) e) MTU_State_CountRestarts *;returns "Restarts"* f) Evaluation 1)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluations	1) Check "Restarts"
Test passed	"Restarts" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.5 Retries after 3 reply messages with incorrect Checksum (PREOP)

Table 243 defines the test conditions for this test case.

Table 243 – Retries after 3 reply messages with incorrect Checksum (PREOP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0342
Name	TCM_DLCC_RTRY_CKSINCORR3PREOP&RESTOP
Purpose (short)	Retry behavior after 3 reply messages with incorrect CKS at PREOP (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at PREOPERATE. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is still incorrect.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* c) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_IncorrectChecksum(n=3) *;3 x reply with incorrect checksum* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;dummy read for delay* i) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* j) Evaluation 3) k) MTU_State_CheckPreoperate *;returns "reached"* l) Evaluation 4) m) MTU_State_CountRestarts *;returns "Restarts"* n) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check ArgBlock "PortStatusList" 4) Check "reached" 5) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and *;PREOPERATE reached* "reached" = TRUE, and "PortStatusList.PortStatusInfo" = PORT_DIAG, and *;PREOPERATE reached* "reached" = TRUE, and "RESTARTS" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.6 Retries after 3 reply messages with incorrect Checksum (OPERATE)

Table 244 defines the test conditions for this test case.

Table 244 – Retries after 3 reply messages with incorrect Checksum (OPERATE)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0343
Name	TCM_DLCC_RTRY_CKSINCORR3OPER&RESTOP
Purpose (short)	Retry behavior after 3 reply messages with incorrect CKS at OPERATE (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving reply message with incorrect checksum at OPERATE. Master repeats the request message and again reply with incorrect checksum. After a third trial, reply is still incorrect.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_IncorrectChecksum(*n*=3) *;3 x reply with incorrect checksum* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;dummy read for delay* i) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* j) Evaluation 4) k) MTU_State_CheckPreoperate *;returns "reached"* l) Evaluation 5) m) MTU_State_CountRestarts *;returns "Restarts"* n) Evaluation 6)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check ArgBlock "PortStatusList" 4) Check "reached" 5) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = OPERATE, and "reached" = TRUE, and "PortStatusList.PortStatusInfo" = OPERATE, and "reached" = TRUE, and "RESTARTS" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.7 Retries after 2 missing replies and finally correct Checksum (STARTUP)

Table 245 defines the test conditions for this test case.

Table 245 – Retries after 2 missing replies and finally correct Checksum (STARTUP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0344
Name	TCM_DLCC_RTRY_MISSREP2STRTUP
Purpose (short)	Retry behavior after 2 missing replies and finally correct Checksum in STARTUP
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at STARTUP. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. Master repeats the message once more and receives a valid reply message from MTU.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) MTU_State_SkipResponse (*n*=2) *;2 x no reply message* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* d) TM_AWAIT_PORT_STATUS (OPERATE) e) MTU_State_CountRestarts *;returns "Restarts"* f) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Restarts"
Test passed	"Restarts" = 1
Test not passed (examples)	Any evaluation failed, or Communication error, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.8 Retries after 2 missing replies and finally correct Checksum (PREOP)

Table 246 defines the test conditions for this test case.

Table 246 – Retries after 2 missing replies and finally correct Checksum (PREOP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0345
Name	TCM_DLCC_RTRY_MISSREP2PREOP
Purpose (short)	Retry behavior after 2 missing replies and finally correct Checksum in PREOP
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at PREOPERATE. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. Master repeats the message once more and receives a valid reply message from MTU.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* c) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_SkipResponse (*n*=2) *;2 x no reply message* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *; dummy read for delay* i) MTU_State_CheckPreoperate" *;returns "reached"* j) Evaluation 3) k) MTU_State_CountRestarts *;returns "Restarts"* l) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check "reached" 4) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and *;PREOPERATE reached* "reached" = TRUE, and "reached" = TRUE, and "Restarts" = 1
Test not passed (examples)	Any evaluation failed, or SMI_DeviceRead with negative response, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.9 Retries after 2 missing replies and finally correct Checksum (OPERATE)

Table 247 defines the test conditions for this test case.

Table 247 – Retries after 2 missing replies and finally correct Checksum (OPERATE)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0346
Name	TCM_DLCC_RTRY_MISSREP2OPER
Purpose (short)	Retry behavior after 2 missing replies and finally correct Checksum in OPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to pass
Specification (clause)	[6], 7.3.3.4, 9.3.3.2, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at OPERATE. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. Master repeats the message once more and receives a valid reply message from MTU.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_SkipResponse (*n*=2) *;2 x no reply message* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;returns "VendorName"* i) Evaluation 3) j) MTU_State_CountRestarts *;returns "Restarts"* k) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check "VendorName" 4) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = OPERATE, and "reached" = TRUE, and "VendorName" = IO-Link Community, and "Restarts" = 1
Test not passed (examples)	Any evaluation failed or SMI_DeviceRead with negative response Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.10 Retries after 3 missing replies ending with restart/stop (STARTUP)

Table 248 defines the test conditions for this test case.

Table 248 – Retries after 3 missing replies ending with restart/stop (STARTUP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0347
Name	TCM_DLCC_RTRY_REPMISS3STRTUP&RESTOP
Purpose (short)	Retry behavior after 3 missing reply messages at STARTUP (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, 9.3.3.2, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at STARTUP. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. After a third trial, reply is still missing.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) MTU_State_SkipResponse (*n*=3) *;3 x no reply message* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* d) TM_AWAIT_PORT_STATUS (OPERATE) e) MTU_State_CountRestarts *;returns "Restarts"* f) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check "Restarts"
Test passed	"Restarts" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.11 Retries after 3 missing replies ending with restart/stop (PREOP)

Table 249 defines the test conditions for this test case.

Table 249 – Retries after 3 missing replies ending with restart/stop (PREOP)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0348
Name	TCM_DLCC_RTRY_REPMISS3PREOP&RESTOP
Purpose (short)	Retry behavior after 3 missing reply messages at PREOP (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, 9.3.3.2, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at PREOPERATE. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. After a third trial, reply is still missing.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* c) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_SkipResponse (*n*=3) *;3 x no reply message* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;dummy read for delay* i) TM_AWAIT_PORT_STATUS(PORT_DIAG) *;returns ArgBlock "PortStatusList"* j) Evaluation 3) k) MTU_State_CheckPreoperate *;returns "reached"* l) Evaluation 4) m) MTU_State_CountRestarts *;returns "Restarts"* n) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check ArgBlock "PortStatusList" 4) Check "reached" 5) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = PORT_DIAG, and *;PREOPERATE reached* "reached" = TRUE, and "PortStatusList.PortStatusInfo" = PORT_DIAG, and *;PREOPERATE reached* "reached" = TRUE, and "RESTARTS" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.12 Retries after 3 missing replies ending with restart/stop (OPERATE)

Table 250 defines the test conditions for this test case.

Table 250 – Retries after 3 missing replies ending with restart/stop (OPERATE)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0349
Name	TCM_DLCC_RTRY_REPMISS3OPER&RESTOP
Purpose (short)	Retry behavior after 3 missing reply messages at OPERATE (restart/stop)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.3.4, 9.3.3.2, Figure 40, Table 46
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Retry behavior of Master after receiving no reply messages at OPERATE. Master sends request message, MTU does not send reply message. Master sends message again, MTU again does not send reply message. After a third trial, reply is still missing.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;begin with start-up* c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* d) Evaluation 1) e) MTU_State_CheckPreoperate *;returns "reached"* f) Evaluation 2) g) MTU_State_SkipResponse (*n*=3) *;3 x no reply message* h) SMI_DeviceRead(ABPS_DEVICEREAD [Index=0x10]) *;dummy read for delay* i) TM_AWAIT_PORT_STATUS(OPERATE) *;returns ArgBlock "PortStatusList"* j) Evaluation 3) k) MTU_State_CheckPreoperate *;returns "reached"* l) Evaluation 4) m) MTU_State_CountRestarts *;returns "Restarts"* n) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check ArgBlock "PortStatusList" 2) Check "reached" 3) Check ArgBlock "PortStatusList" 4) Check "reached" 5) Check "Restarts"
Test passed	"PortStatusList.PortStatusInfo" = OPERATE, and "reached" = TRUE, and "PortStatusList.PortStatusInfo" = OPERATE, and "reached" = TRUE, and "RESTARTS" = 2
Test not passed (examples)	Any evaluation failed, or Procedure does not terminate
Report	All evaluations <ok | nok>

8.9.13 Maximum WURQs and no final success

Table 251 defines the test conditions for this test case.

Table 251 – Maximum WURQs and no final success

-CR046- -CR92-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0222
Name	TCM_DLCC_RTRY_MAXWURQNOSUCCESS
Purpose (short)	Behavior at maximum WURQ sequences and no final success
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master robustness test: test to fail
Specification (clause)	[6], 7.3.2.2, 7.3.2.4, Figures 31, 32, and 33
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master behavior with maximum WURQ sequences and no final success. Master is in STARTUP mode. Master sends at first a Wakeup pulse (WURQ) with subsequent communication requests. MTU does not respond to the subsequent communication requests. Master resends the Wakeup pulse (WURQ) with subsequent communication requests and again there is no response from MTU. Master repeats again the Wakeup pulse (WURQ) with subsequent communication requests and again there is no response from MTU. After a break, Master restarts the Wake-up requests and this time MTU responds in a correct manner.
Precondition	EUT: PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_State_CountRestarts *;reset counter* b) MTU_State_SkipResponse (*n*=3, WURQ=TRUE) *;3 x no reply message* c) SMI_PortConfiguration (ABPS_NOTYPE_CHECK) *;begin with start-up* d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_State_CheckOperate *;returns "reached"* f) Evaluations 1) g) MTU_State_CountRestarts *;returns "Restarts"* h) Evaluation 2)
Test parameter	-
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluations	1) Check “reached" 2) Check "Restarts"
Test passed	"reached" = TRUE, and "Restarts" = 4 *;4 restarts counted, last with 1 s delay*
Test not passed (examples)	Any evaluation failed, or Communication error
Report	All evaluations <ok | nok>

8.10 ISDU (Indexed Service Data Unit) – Application ErrorTypes

8.10.1 ISDU Write rejected with ErrorType

Table 252 defines the test conditions for this test case.

Table 252 – ISDU Write rejected with ErrorType

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0223
Name	TCM_ALIC_AERR_WRITEREJECT
Purpose (short)	ISDU Write service rejected with defined ErrorType, no details
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write service rejected with ErrorType, no details. Access to a supported Index of the MTU is rejected with an application error without details. The response reports an ErrorCode "0x8000" (APP_DEV).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16383,Subindex=0,Value=[0x00],ErrorType_W=0x8000) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16383,Subindex=0,OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8000
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.2 ISDU Write to unsupported Index rejected with ErrorType

Table 253 defines the test conditions for this test case.

Table 253 – ISDU Write to unsupported Index rejected with ErrorType

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0224
Name	TCM_ALIC_AERR_WRITEINDEXUNSUPPORTED
Purpose (short)	ISDU Write to unsupported Index rejected with ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write to unsupported Index rejected with ErrorType. Access to a non-supported Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8011" (IDX_NOTAVAIL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=254, Subindex=0, Value=[0x00], ErrorType_W=0x8011) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=254, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8011
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.3 ISDU Write to unsupported Subindex rejected with ErrorType

Table 254 defines the test conditions for this test case.

Table 254 – ISDU Write to unsupported Subindex rejected with ErrorType

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0225
Name	TCM_ALIC_AERR_WRITESUBINDEXNOTSUPPORTED
Purpose (short)	ISDU Write to unsupported Subindex (>0) rejected with ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	ISDU Write to unsupported Subindex (>0) rejected with ErrorType. Access to a non-supported Subindex in the MTU is rejected with an application error. The response reports an ErrorCode "0x8012" (SUBIDX_NOTAVAIL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=253, Subindex=0, Value=[0x00], ErrorType_W=0x8012) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=253, Subindex=1, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8012
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.4 ISDU Write to temporarily unavailable Index rejected with ErrorType

Table 255 defines the test conditions for this test case.

Table 255 – ISDU Write to temporarily unavailable Index rejected with ErrorType

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0226
Name	TCM_ALIC_AERR_WRITETEMPUNAV
Purpose (short)	ISDU Write to temporarily unavailable Index rejected with ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write to temporarily unavailable Index rejected with ErrorType. Access to a temporarily unavailable Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8020" (SERV_NOTAVAIL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=252, Subindex=0, Value=[0x00], ErrorType_W=0x8020) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=252, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8020
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.5 ISDU Write to temporarily unavailable Index due to local control

Table 256 defines the test conditions for this test case.

Table 256 – ISDU Write to temporarily unavailable Index due to local control

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0227
Name	TCM_ALIC_AERR_WRITEINDEXTEMPANAVLC
Purpose (short)	ISDU Write to temporarily unavailable Index due to local control
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to a temporarily unavailable Index due to local control in the MTU is rejected with an application error. The response reports an ErrorCode "0x8021" (SERV_NOTAVAIL_LOCCRTL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=251, Subindex=0, Value=[0x00], ErrorType_W=0x8021) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=251, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8021
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.6 ISDU Write to temporarily unavailable Index due to Device control

Table 257 defines the test conditions for this test case.

Table 257 – ISDU Write to temporarily unavailable Index due to Device control

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0228
Name	TCM_ALIC_AERR_WRITEINDEXTEMPANAVDC
Purpose (short)	ISDU Write to temporarily unavailable Index due to Device control
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to temporarily unavailable Index due to Device control in the MTU is rejected with an application error. The response reports an ErrorCode "0x8022" (SERV_NOTAVAIL_DEVCRTL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=250, Subindex=0, Value=[0x00], ErrorType_W=0x8022) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=250, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8022
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.7 ISDU Write to read-only Index denied

Table 258 defines the test conditions for this test case.

Table 258 – ISDU Write to read-only Index denied

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0229
Name	TCM_ALIC_AERR_WRITEINDEXRO
Purpose (short)	ISDU Write to read-only Index denied
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to read-only Index in the MTU is denied with an application error. The response reports an ErrorCode "0x8023" (IDX_NOT_ACCESSIBLE).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=249, Subindex=0, Value=[0x00], ErrorType_W=0x8023) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=249, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8023
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.8 ISDU Write with invalid Length

Table 259 defines the test conditions for this test case.

Table 259 – ISDU Write with invalid Length

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0230
Name	TCM_ALIC_AERR_WRITEINVALIDLEN
Purpose (short)	ISDU Write with invalid Length
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access with too short data length to an Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8034" (VAL_LENUNDRUN).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=248, Subindex=0, Value=[0x00], ErrorType_W=0x8034) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=248, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8034
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.9 ISDU Write with parameter value out of range

Table 260 defines the test conditions for this test case.

Table 260 – ISDU Write with parameter value out of range

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0231
Name	TCM_ALIC_AERR_WRITEPARAMOUTOFRNG
Purpose (short)	ISDU Write with parameter value out of range
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access with parameter values out of range to an Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8030" (PAR_VALOUTOFRNG).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16382, Subindex=0, Value=[0x00], ErrorType_W=0x8030) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16382, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8030
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.10 ISDU Write with parameter value above limit

Table 261 defines the test conditions for this test case.

Table 261 – ISDU Write with parameter value above limit

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0232
Name	TCM_ALIC_AERR_WRITEPARAMABOVELIMIT
Purpose (short)	ISDU Write with parameter value above limit
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access with parameter values above limit to an Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8031" (PAR_VALGTLIM).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16381, Subindex=0, Value=[0x00], ErrorType_W=0x8031) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16381, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8031
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.11 ISDU Write with parameter value below limit

Table 262 defines the test conditions for this test case.

Table 262 – ISDU Write with parameter value below limit

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0233
Name	TCM_ALIC_AERR_WRITEPARAMBELOWLIMIT
Purpose (short)	ISDU Write with parameter value below limit
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access with parameter values below limit to an Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8032" (PAR_VALLTLIM).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16380, Subindex=0, Value=[0x00], ErrorType_W=0x8032) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16380, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8032
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.12 ISDU Write with invalid parameter set

Table 263 defines the test conditions for this test case.

Table 263 – ISDU Write with invalid parameter set

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0234
Name	TCM_ALIC_AERR_WRITEPARAMINVALID
Purpose (short)	ISDU Write with invalid parameter set
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access with invalid parameter values to an Index in the MTU is rejected with an application error. For example, lower threshold value is above upper threshold value. The response reports an ErrorCode "0x8040" (PAR_SETINVALID).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16379, Subindex=0, Value=[0x00], ErrorType_W=0x8040) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16379, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8040
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.13 ISDU Write while Device application fault

Table 264 defines the test conditions for this test case.

Table 264 – ISDU Write while Device application fault

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0235
Name	TCM_ALIC_AERR_WRITEDEVICEAPPFAULT
Purpose (short)	ISDU Write while Device application fault
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU, whose technology specific application is not performing, is rejected with an application error. The response reports an Error-Code "0x8082" (APP_DEVNOTRDY).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16378, Subindex=0, Value=[0x00], ErrorType_W=0x8082) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16378, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8082
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.14 ISDU Write to reserved Indices

Table 265 defines the test conditions for this test case.

Table 265 – ISDU Write to reserved Indices

-CR099- -CR106-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0237
Name	TCM_ALIC_AERR_WRITERESERVEDINDEX
Purpose (short)	ISDU Write to reserved Indices
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Table B.7, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to a reserved Index in the MTU is rejected with an application error. The response reports an ErrorCode "0x8011" (IDX_NOTAVAIL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=65025, Subindex=0, Value=[0x00], ErrorType_W=0x8011) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=65025, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8011
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.10.15 ISDU Write to reserved Indices and no ISDU

Table 266 defines the test conditions for this test case.

Table 266 – ISDU Write to reserved Indices and no ISDU

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0238
Name	TCM_ALIC_AERR_WRITERESERVEDINDEXNOISDU
Purpose (short)	ISDU Write to reserved Indices and unavailable ISDU is rejected with ErrorType.
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to a reserved Index and ISDU not available in MTU is rejected by an application error.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_DI
Procedure	a) MTU_DPP1_Set(RevisionID=1.0) b) MTU_DPP1_Set(M-sequenceCapability=0x10) *;ISDU not supported* c) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) d) TM_AWAIT_PORT_STATUS(OPERATE) e) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16, Subindex=0, OD=[0x00])) f) Evaluation 1)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check returned ArgBlock
Test passed	ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x8011
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.11 ISDU (Indexed Service Data Unit) – Derived ErrorTypes

8.11.1 ISDU Write response without busy indication

Table 267 defines the test conditions for this test case. MTU specialty "NO_DEVICE_BUSY" causes an immediate Device response without busy indication (see A.4.8.3).

Table 267 – ISDU Write response without busy indication

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0239
Name	TCM_ALIC_DERR_WRITENOBUSY
Purpose (short)	ISDU Write response “No service” (without busy) reports Derived ErrorType.
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU. The MTU directly (without any busy response) responds with “No service” (I-Service/Length = 0x00). The response provides the Derived ErrorCode "0x1000" (COM_ERR).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16376, Subindex=0, Value=[0x00], Specialty=NO_SERVICE) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16376, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x1000
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.11.2 ISDU Write response with timeout after busy indication

Table 268 defines the test conditions for this test case. MTU specialty "TIMEOUT" causes an inappropriate delay of Device's response (see A.4.8.3).

Table 268 – ISDU Write response with timeout after busy indication

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0240
Name	TCM_ALIC_DERR_WRITEAFTERBUSYTIMEOUT
Purpose (short)	ISDU Write response with timeout after busy indication reports Derived ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], 8.3.2.3, 10.8.7, A.5.2, Annex C, Tables B.7, C.2
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU, which does not generate a response after an adequate time for the ISDU acknowledgement and despite indicating the "Device busy" bit in the ISDU Service/Length octet, is responded with an application error. The response reports a derived ErrorCode "0x1000" (COM_ERR) or "0x1100" (I-SERVICE_TIMEOUT).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16376, Subindex=0, Value=[0x00], Specialty=TIMEOUT) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16376, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x1000 or 0x1100
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.11.3 ISDU Write response with illegal service code

Table 269 defines the test conditions for this test case. MTU specialty "INCORRECT_SERVICE_CODE" causes an incorrect service code in Device's response (see A.4.8.3).

Table 269 – ISDU Write response with illegal service code

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0241
Name	TCM_ALIC_DERR_ILLSERVICECODE
Purpose (short)	ISDU Write response with illegal service code reports Derived ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C, Table C.2, Figure A.19
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU generating a response with incorrect service code, is responded with an application error. The response reports a derived ErrorCode "0x5700" (M_ISDU_ILLEGAL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16374, Subindex=0, Value=[0x00], Specialty=INCORRECT_SERVICE_CODE) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16374, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x5700
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.11.4 ISDU Write response with wrong checksum (CHKPDU)

Table 270 defines the test conditions for this test case. MTU specialty "INCORRECT_CHKPDU" causes a checksum error in Device's response (see A.4.8.3).

Table 270 – ISDU Write response with wrong checksum (CHKPDU)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0242
Name	TCM_ALIC_DERR_WRONGCHECKSUM
Purpose (short)	ISDU Write response with incorrect checksum (CHKPDU) reports Derived ErrorType.
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C, Table C.2, Figure A.19
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU generating a response with wrong CHKPDU, is responded with an application error. The response reports a derived ErrorCode "0x5600" (M_ISDU_CHECKSUM).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16374, Subindex=0, Value=[0x00], Specialty=INCORRECT_CHKPDU) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16374, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x5600
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.11.5 ISDU Write response with reserved data length

Table 271 defines the test conditions for this test case. MTU specialty "RESERVED_DATA-_LENGTH" causes reserved combinations of iService and length in Device's response (see A.4.8.3).

Table 271 – ISDU Write response with reserved data length

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0244
Name	TCM_ALIC_DERR_WRITERESERVEDDL
Purpose (short)	ISDU Write response with reserved data length reports Derived ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, Annex C, Table C.2, Figure A.19
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU is responded with "reserved data length" and results in an application error. The response reports a Derived ErrorCode "0x1000" (COM_ERR) or 0x5700 (M_ISDU_ILLEGAL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate ;returns "OPERATE-reached" b) Evaluation 1) c) MTU_ISDU_Add(Index=16372, Subindex=0, Value=[0x00], Specialty=RESERVED_DATA_LENGTH) d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16372, Subindex=0, OD=[0x00])) e) Evaluation 2)
Test parameter	-
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "Job Error" received, and JobError.ErrorCode/AdditionalCode = 0x1000 or 0x5700
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12 ISDU (Indexed Service Data Unit) – Limit checks

8.12.1 ISDU Read response without data

Table 272 defines the test conditions for this test case.

Table 272 – ISDU Read response without data

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0243
Name	TCM_ALIC_DERR_READNODATA
Purpose (short)	ISDU Read response without data reports no Derived ErrorType
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.7, Annex C, Figure A.19
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Read access to an Index in the MTU generating a positive response without data is responded without error.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16372, Subindex=0, Value=[]) *;prepare MTU* d) SMI_DeviceRead(ABPS_DEVICEREAD(Index=16372, Subindex=0)) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "On-request_Data" received, and OD = []
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.2 ISDU Write with minimum data length (zero octets)

Table 273 defines the test conditions for this test case.

Table 273 – ISDU Write with minimum data length (zero octets)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0245
Name	TCM_ALIC_LIMT_WRITEMINDATALENGTH
Purpose (short)	ISDU Write with minimum data length (zero octets)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU using the minimum data of zero octets. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16000, Subindex=0, Value=[]) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16000, Subindex=0, OD=[])) e) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.3 ISDU Write with maximum service length (238 octets)

Table 274 defines the test conditions for this test case.

Table 274 – ISDU Write with maximum service length (238 octets)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0246
Name	TCM_ALIC_LIMT_WRITEMAXDATALENGTH
Purpose (short)	ISDU Write with maximum service length (238 octets)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an Index in the MTU using the maximum service length of 238 octets. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16001, Subindex=0, Value=<data>) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16001, Subindex=0, OD=<data>)) e) Evaluation 2) f) MTU_ISDU_Read(Index=16001,Subindex=0) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	data = [0x00, 0x01 to 0xE7] (232 octets user data)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = data
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.4 ISDU Read with maximum service length (238 octets)

Table 275 defines the test conditions for this test case.

Table 275 – ISDU Read with maximum service length (238 octets)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0248
Name	TCM_ALIC_LIMT_READMAXDATALENGTH
Purpose (short)	ISDU Read with maximum service length (238 octets)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Read access to an Index in the MTU using the maximum service length of 238 octets. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16003, Subindex=0, Value=<data>) *;prepare MTU* d) SMI_DeviceRead(ABPS_DEVICEREAD(Index=16003, Subindex=0)) e) Evaluation 2)
Test parameter	data = [0x01, 0x02 to 0xE8] (232 octets user data)
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock
Test passed	"OPERATE-reached" = TRUE ArgBlock "On-request_Data" received, and OD = data
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.5 ISDU Write to 8-bit Index and no Subindex

Table 276 defines the test conditions for this test case.

Table 276 – ISDU Write to 8-bit Index and no Subindex

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0249
Name	TCM_ALIC_LIMT_WRITEINDEX8NOSUBINDEX
Purpose (short)	ISDU Write to 8-bit Index and no Subindex
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an 8-bit Index in the MTU. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=255, Subindex=0, Value=[0xAB]) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=255, Subindex=0, OD=[0x00])) e) Evaluation 2) f) MTU_ISDU_Read(Index=255, Subindex=0) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = 0x00
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.6 ISDU Write to 8-bit Index and 8-bit Subindex

Table 277 defines the test conditions for this test case.

Table 277 – ISDU Write to 8-bit Index and 8-bit Subindex

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0250
Name	TCM_ALIC_LIMT_WRITEINDEX8SUBINDEX8
Purpose (short)	ISDU Write to 8-bit Index and 8-bit Subindex
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an 8-bit Index and 8-bit Subindex in the MTU. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=255, Subindex=255, Value=[0xAB]) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=255, Subindex=255, OD=[0x00])) e) Evaluation 2) f) MTU_ISDU_Read(Index=255, Subindex=255) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = 0x00
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.7 ISDU Write to 16-bit Index and 8-bit Subindex

Table 278 defines the test conditions for this test case.

Table 278 – ISDU Write to 16-bit Index and 8-bit Subindex

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0251
Name	TCM_ALIC_LIMT_WRITEINDEX16SUBINDEX8
Purpose (short)	ISDU Write to 16-bit Index and 8-bit Subindex
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to a 16-bit Index and 8-bit Subindex in the MTU. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=16004, Subindex=1, Value=[0xAB]) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=16004, Subindex=1, OD=[0x00])) e) Evaluation 2) f) MTU_ISDU_Read(Index=16004, Subindex=1) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = 0x00
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.8 ISDU Write response without busy bit

Table 279 defines the test conditions for this test case.

Table 279 – ISDU Write response without busy bit

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0252
Name	TCM_ALIC_LIMT_IMMEDIATERESPNOBUSY
Purpose (short)	ISDU Write immediate response (without busy)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Figure A.19, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an 8-bit Index and 8-bit Subindex in the MTU. MTU generates an immediate response to the request without any busy respones (no I-Service/Length = 0x01 responses, see Table A.14).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=255, Subindex=1, Value=[0xAB], Specialty=NO_DEVICE_BUSY) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=255, Subindex=1, OD=[0x00])) e) Evaluation 2) f) MTU_ISDU_Read(Index=255, Subindex=1) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = 0x00
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.9 ISDU Write response with busy bit

Table 280 defines the test conditions for this test case.

Table 280 – ISDU Write response with busy bit

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0253
Name	TCM_ALIC_LIMT_IMMEDIATERESPWITHBUSY
Purpose (short)	ISDU Write response with at least one busy response
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Figure A.19, Annex C
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to an 8-bit Index and 8-bit Subindex in the MTU. MTU generates an ISDU response to the request with at least one busy response (I-Service/Length = 0x01, see Table A.14)
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) MTU_ISDU_Add(Index=255, Subindex=2, Value=[0xAB], Specialty=DEVICE_BUSY) *;prepare MTU* d) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=255, Subindex=2, OD=[0x00])) e) Evaluation 2) f) MTU_ISDU_Read(Index=255, Subindex=2) *;returns <value> (octet string)* g) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE ArgBlock "VoidBlock" received Returned <value> = 0x00
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.10 ISDU Write with maximum service Length (15 octets)

Table 281 defines the test conditions for this test case.

Table 281 – ISDU Write with maximum service Length (15 octets)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0254
Name	TCM_ALIC_LIMT_WRITEMAXSERVICELEN15
Purpose (short)	ISDU service (with maximum service length 15) is carried out.
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to several Index and Subindex combinations in the MTU. MTU with maximum service length 15. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) Assign first values to "index", "subindex", "init", "data" d) MTU_ISDU_Add(Index=<index>, Subindex=<subindex>, Value=<init>) *;prep MTU* e) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=<index>, Subindex=<subindex>, OD=<data>)) f) Evaluation 2) g) MTU_ISDU_Read(Index=<index>, Subindex=<subindex>) *;returns <value>* h) Evaluation 3) i) Repeat from c) with next "index", "subindex", "init", "data"

Test parameter	index = {100, 101, 16005}
	subindex = {0, 1, 1}
	init = {
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00],
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00],
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]}
	data = {
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C],
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B],
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A]}

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" For all loops: 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE For all loops: ArgBlock "VoidBlock" received Returned <value> = data
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.12.11 ISDU Write with minimum service Extended Length (17)

Table 282 defines the test conditions for this test case.

Table 282 – ISDU Write with minimum service Extended Length (17)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0255
Name	TCM_ALIC_LIMT_WRITEMINSERVICEEXTLEN17
Purpose (short)	ISDU service (with minimum Extended Length 17) is carried out.
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master ISDU test: test to pass
Specification (clause)	[5], 8.2.4.1.2; [6], A.5.2, A.5.4, A.5.7, Annex C, Table A.15, Figure A.18
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	ISDU Write access to several Index and Subindex combinations in the MTU. MTU with minimum Extended Length 17. The response shall be positive.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "OPERATE-reached"* b) Evaluation 1) c) Assign first values to "index", "subindex", "init", "data" d) MTU_ISDU_Add(Index=<index>, Subindex=<subindex>, Value=<init>) *;prep MTU* e) SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=<index>, Subindex=<subindex>, OD=<data>)) f) Evaluation 2) g) MTU_ISDU_Read(Index=<index>, Subindex=<subindex>) *;returns <value>* h) Evaluation 3) i) Repeat from c) with next "index", "subindex", "init", "data"

Test parameter	index = {100, 101, 16005}
	subindex = {0, 1, 1}
	init = {
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]
	[0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00]}
	data = {
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0x0C, 0x0D]
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B, 0c0C]
	[0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09, 0x0A, 0x0B]}

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "OPERATE-reached" For all loops: 2) Check returned ArgBlock 3) Check returned <value>
Test passed	"OPERATE-reached" = TRUE For all loops: ArgBlock "VoidBlock" received Returned <value> = data
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13 Events

8.13.1 General

The Event propagation to the upper-level system, for example a fieldbus, is not subject matter of this document. This behavior shall be defined in the corresponding "upper-level systems integration" specification. Thus, there is no immediate Event acknowledgement of the Master as with the Legacy-Master. Therefore, the timeout for waiting on the acknowledgement shall be adjustable in the MTU.

8.13.2 Master receives Events without details (notification)

Table 283 defines the test conditions for this test case.

Table 283 – Master receives Events without details (notification)

-CR083-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0256
Name	TCM_ALIC_EVNT_NODETAILSNOTIFY
Purpose (short)	Master receives Event (notification) without details (notification)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master Event test: test to pass
Specification (clause)	[5], 7.2.4.4.2.1; [6], 7.3.8.3, 8.3.3.1, 11.6, Annex A.6, Annex D
Configuration / setup	Master-Tester-System
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Master receives Event without details (StatusCode type 1, notification). Master extracts the EventCode (type 1), converts it accordingly into StatusCode (type 2) and transfers it to the upper level system. Master acknowledges the Event. MTU resets the Event flag. Five different combinations are tested in TPLs.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) c) TM_AWAIT_PORT_STATUS(OPERATE) d) Assign first values to "status_code" *;see Test parameter* e) DLL_ClearAllEvents *;clears buffer in test interface* f) MTU_Event_Clear *;removes Events in memory* g) MTU_Event_SetStatusCode(<status_code>) *;enters StatusCode in memory* h) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* i) Evaluation 2 j) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList" with converted Event* k) Evaluation 3) l) Repeat from e) with next "status_code"

Test parameter	status_code = {0x01, 0x02, 0x04, 0x08, 0x10}
	event_code = {0xFF80, 0xFF80, 0x6320, 0xFF80, 0xFF10}

Post condition	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check "EventAck" 2) Check "EventList"
Test passed	For all TPLs: "MTU in OPERATE" = TRUE, and EventAck.StatusCodeRead = TRUE, EventAck.Acknowledge = TRUE, and EventList.Entry1 (EventQualifier = 0x54, EventCode = <event_code>) *;type1 to type2*
Test not passed (examples)	Any evaluation failed

TEST CASE RESULTS	CHECK / REACTION	
Report	All evaluations	<ok | nok>

2716

8.13.3 Master receives event with details (single event)

Table 284 defines the test conditions for this test case.

Table 284 – Master receives event with details (single event)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0261
Name	TCM_ALIC_EVNT_WITHDETAILSSINGLEEVENT
Purpose (short)	Master receives Event with details (single Event)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6, Annex D
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event with details (StatusCode type 2). Master transfers the EventCode and EventQuaifier to the upper-level system. Master acknowledges the Event. MTU resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* d) MTU_Event_SetSlot(1, 0xF4, 0x1000) *;enters in slot 1 Event Q and Code* e) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* f) Evaluation 2 g) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* h) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "EventAck" 3) Check "EventList"

Test passed	"MTU in OPERATE" = TRUE, and
	EventAck.StatusCodeRead = TRUE,
	EventAck.AllSlotsRead = TRUE,
	EventAck.Acknowledge = TRUE, and
	EventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1000)

Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.4 Master receives event with details (double event)

Table 285 defines the test conditions for this test case.

Table 285 – Master receives event with details (double event)

-CR074-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0262
Name	TCM_ALIC_EVNT_WITHDETAILSDOUBLEEVENT
Purpose (short)	Master receives Event with details (double Event)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6, Annex D
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event with details (StatusCode type 2). Master transfers it to the upper level system. Master acknowledges the Event. The MTU resets the Event flag. This procedure can be carried out in loops for different Events in the Event buffer (TPL).
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) Assign first values to "status_code", "slot_a", "slot_b",”qual_a”, “qual_b” d) DLL_ClearAllEvents *;clears buffer in test interface* e) MTU_Event_Clear *;removes Events in memory* f) MTU_Event_SetStatusCode(<status_code>) *;enters StatusCode in memory* g) MTU_Event_SetSlot(<slot_a>, <qual_a>, 0x1000) *;enters Event 1 in memory* h) MTU_Event_SetSlot(<slot_b>, <qual_b>, 0x4210) *;enters Event 2 in memory* i) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* j) Evaluation 2 k) DLL_GetDeviceEvents(2) *;minED = 2, returns "EventList"* l) Evaluation 3) m) wait for 1 second ; to ensure that the event rules are respected n) Repeat from d) with next "status_code", “slot_a”, “slot_b”, “qual_a”,”qual_b”

Test parameter	status_code = {0xA1, 0x92, 0x8C, 0x8C, 0x92, 0xA1}
	slot_a = {1, 2, 3, 4, 5, 6}
	slot_b = {6, 5, 4, 3, 2, 1}
	qual_a = {0xF4, 0xB4, 0xE4, 0xA4, 0xF4, 0xB4}
	qual_b = {0xE4, 0xA4, 0xF4, 0xB4, 0xE4, 0xA4}

Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "EventAck" 3) Check "EventList"
Test passed	For all TPLs: "MTU in OPERATE" = TRUE, and EventAck: StatusCodeRead = TRUE, AllSlotsRead = TRUE, Acknowledge = TRUE, and EventList: 2 elements EventList: Entry1 (EventQualifier = “qual_a”, EventCode = 0x1000), Entry2 (EventQualifier = “qual_b”, EventCode = 0x4210)
Test not passed (examples)	Any evaluation failed

TEST CASE RESULTS	CHECK / REACTION
Report	All evaluations <ok | nok>

8.13.5 Events of type notification

Table 286 defines the test conditions for this test case.

Table 286 – Events of type notification

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0308
Name	TCD_DLIC_EVNTOPER_SINGLENOTIFICATION
Purpose (short)	Master receives Event with details (single Event, notification)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6, Annex D
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event with details (StatusCode type 2, notification). Master transfers the EventCode and EventQualifier to the upper-level system (optional). Master acknowledges the Event. MTU resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* d) MTU_Event_SetSlot(1, 0x54, 0x1800) *;enters Event in memory* e) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* f) Evaluation 2 g) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* h) Evaluation 3) i) SMI_PortStatus *;returns "ArgBlock PortStatusList"* j) Evaluation 4)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "EventAck" 3) Check "EventList" 4) Check "ArgBlock PortStatusList"

Test passed	"MTU in OPERATE" = TRUE, and
	EventAck.StatusCodeRead = TRUE,
	EventAck.AllSlotsRead = TRUE,
	EventAck.Acknowledge = TRUE, and
	EventList.Entry1 (EventQualifier = 0x54, EventCode = 0x1800), and *;"No Device"*
	PortStatusList.NumberOfDiags = 0

Test not passed	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.6 Master receives Event with details (six Events)

Table 287 defines the test conditions for this test case.

Table 287 – Master receives Event with details (six Events)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0263
Name	TCM_ALIC_EVNT_WITHDETAILSSIXEVENTS
Purpose (short)	Master receives event with details (six Events)
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master Event test: test to pass
Specification (clause)	[5], 7.2.4.4.2.1; [6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives six Events with details (StatusCode type 2). Master transfers it to the upper-level system. Master acknowledges the Event. MTU resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_Event_SetStatusCode(0xBF) *;enters StatusCode in memory* d) MTU_Event_SetSlot(1, 0xF4, 0x1800) *;enter Event appears, Error* e) MTU_Event_SetSlot(2, 0xE4, 0x1801) *;enter Event appears, Warning* f) MTU_Event_SetSlot(3, 0x54, 0x1802) *;enter Event single shot, Notification* g) MTU_Event_SetSlot(4, 0xB4, 0x1803) *;enter Event disappears, Error* h) MTU_Event_SetSlot(5, 0xA4, 0x1804) *;enter Event disappers, Warning* i) MTU_Event_SetSlot(6, 0x54, 0x1805) *;enter Event single shot, Notification* j) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* k) Evaluation 2 l) DLL_GetDeviceEvents(6) *;minED = 6, returns "EventList"* m) Evaluation 3) n) SMI_PortStatus *;returns "ArgBlock PortStatusList"* o) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "EventAck" 3) Check "EventList" 4) Check "ArgBlock PortStatusList"
Test passed	"MTU in OPERATE" = TRUE, and EventAck.StatusCodeRead = TRUE, EventAck.AllSlotsRead = TRUE, EventAck.Acknowledge = TRUE, and EventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1800), EventList.Entry2 (EventQualifier = 0xE4, EventCode = 0x1801), EventList.Entry3 (EventQualifier = 0x54, EventCode = 0x1802), EventList.Entry4 (EventQualifier = 0xB4, EventCode = 0x1803), EventList.Entry5 (EventQualifier = 0xA4, EventCode = 0x1804), EventList.Entry6 (EventQualifier = 0x54, EventCode = 0x1805), and PortStatusList.NumberOfDiags = 2, PortStatusList.DiagEntry0: Qualifier = 0xF4, Code = 0x1800 PortStatusList.DiagEntry1: Qualifier = 0xE4, Code = 0x1801
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.7 Master receives Event while in ISDU Write transfer (stopover)

Table 288 defines the test conditions for this test case.

Table 288 – Master receives Event while in ISDU Write transfer (stopover)

-CR084- -CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0264
Name	TCM_ALIC_EVNT_WRITEISDUWITH EVENT
Purpose (short)	Master receives one Event while in ISDU transfer (stopover; no details)
Equipment under test (EUT)	Master + Port
Test case version	1.3
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6, Annex I
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives one Event while in ISDU Write transfer (stopover; no details). The Event flag shall be set during transfer of an ISDU. The Master interrupts the ISDU and retrieves the Event content. After Event handling, the ISDU transfer shall be continued.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_ISDU_Add(Index=0x18,Subindex=0, Value=[0xAB,0xCD,0x12,0x34],Specialty=EVENT) *;test data* d) MTU_Event_SetStatusCode(0x01) *;enters StatusCode in memory ;enters in slot 1 Event Q and Code* e)SMI_DeviceWrite(ABPS_DEVICEWRITE(Index=0x18,Subindex=0, Value=[0x00,0x00,0x00,0x00])) *;returns "ArgBlock"* f) Evaluation 2) g) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* h) Evaluation 3) i) MTU_Event_ISDUinterrupted *;returns "Interrupted"* j) Evaluation 4) k) MTU_ISDU_Read(Index=0x18, Subindex=0) *;returns "Value"* l) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "Argblock" 3) Check "EventList" 4) Check "Interrupted" 5) Check "Value"
Test passed	"MTU in OPERATE" = TRUE, and "Argblock" = VoidBlock, and EventList.Entry1 (EventQualifier = 0x54, EventCode = 0xFF80), and "Interrupted" = TRUE, and "Value" = [0x00, 0x00, 0x00, 0x00]
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.8 Master receives Event while in ISDU Read transfer (stopover)

Table 289 defines the test conditions for this test case.

Table 289 – Master receives Event while in ISDU Read transfer (stopover)

-CR084- -CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0265
Name	TCM_ALIC_EVNT_READISDUWITH EVENT
Purpose (short)	Master receives Event while in ISDU Read transfer (stopover; no details)
Equipment under test (EUT)	Master + Port
Test case version	1.3
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives one Event while in ISDU Read transfer (stopover; no details). The Event flag shall be set during transfer of an ISDU. The Master interrupts the ISDU and retrieves the Event content. After Event handling, the ISDU transfer shall be continued.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_ISDU_Add(Index=0x18, Subindex=0, Value=[0xAB,0xCD,0x12,0x34], Specialty=EVENT) *;test data* d) MTU_Event_SetStatusCode(0x01) *;enters StatusCode in memory* e) SMI_DeviceRead(ABPS_DEVICEREAD (Index=0x18, Subindex=0)) *;returns "ArgBlock"* f) Evaluation 2) g) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* h) Evaluation 3) i) MTU_Event_ISDUinterrupted *;returns "Interrupted"* j) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "Argblock" 3) Check "EventList" 4) Check "Interrupted"
Test passed	"MTU in OPERATE" = TRUE, and ArgBlock.OD = [0xAB, 0xCD, 0x12, 0x34], and EventList.Entry1 (EventQualifier = 0x54, EventCode = 0xFF80), and "Interrupted" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.9 Master receives Event details while in ISDU Write transfer (stopover)

Table 290 defines the test conditions for this test case.

Table 290 – Master receives Event details while in ISDU Write transfer (stopover)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0266
Name	TCM_ALIC_EVNT_WRITEISDUWITH EVENTDETAILS
Purpose (short)	Master receives Event while in ISDU Write transfer (stopover; with details)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event while in ISDU Write transfer (stopover; with details). The Event flag shall be set during transfer of an ISDU. The Master interrupts the ISDU and retrieves the Event content. After Event handling, the ISDU transfer shall be continued.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_ISDU_Add(Index=0x18,Subindex=0, Value=[0xAB,0xCD,0x12,0x34],Specialty=EVENT) *;test data* d) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* e) MTU_Event_SetSlot(1, 0xF4, 0x1000) *;enter Event appears, unknown Error* f) SMI_DeviceWrite(ABPS_DEVICE_WITE(Index=0x18,Subindex=0, Value=[0x00,0x00,0x00,0x00])) *;returns "ArgBlock"* g) Evaluation 2) h) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* i) Evaluation 3) j) MTU_Event_ISDUinterrupted *;returns "Interrupted"* k) Evaluation 4)
Test parameter	–
Post condition	-

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "Argblock" 3) Check "EventList" 4) Check "Interrupted"
Test passed	"MTU in OPERATE" = TRUE, and "Argblock" = VoidBlock, and EventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1000), and "Interrupted" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.10 Master receives Event details while in ISDU Read transfer (stopover)

Table 291 defines the test conditions for this test case.

Table 291 – Master receives Event details while in ISDU Read transfer (stopover)

-CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0267
Name	TCM_ALIC_EVNT_READISDUWITH EVENTDETAILS
Purpose (short)	Master receives event while in ISDU Read transfer (stopover; with details)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master Event test: test to pass
Specification (clause)	[6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives one Event while in ISDU Read transfer (stopover; with details). The Event flag shall be set during transfer of an ISDU. The Master interrupts the ISDU and retrieves the Event content. After Event handling, the ISDU transfer shall be continued.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate *;returns "MTU in OPERATE"* b) Evaluation 1) c) MTU_ISDU_Add(Index=0x18,Subindex=0, Value=[0xAB,0xCD,0x12,0x34], Specialty=EVENT) *;test data* d) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* e) MTU_Event_SetSlot(1, 0xF4, 0x1000) *;enter Event appears, unknown Error* f) SMI_DeviceRead(ABPS_DEVICEREAD (Index=0x18,Subindex=0)) *;returns "ArgBlock"* g) Evaluation 2) h) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* i) Evaluation 3) j) MTU_Event_ISDUnterrupted *;returns "Interrupted"* k) Evaluation 4)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "Argblock" 3) Check "EventList" 4) Check "Interrupted"
Test passed	"MTU in OPERATE" = TRUE, and ArgBlock.OD = [0xAB, 0xCD, 0x12, 0x34], and EventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1000), and "Interrupted" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.11 Behavior and Port Event at Device lost (COMLOST)

Table 292 defines the test conditions for this test case.

Table 292 – Behavior and Port Event at Device lost (COMLOST)

-CR065-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0350
Name	TCM_ALIC_EVNT_NODEV
Purpose (short)	Master sends Port Event "No Device" at COMLOST
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master Event test: test to pass
Specification (clause)	[6], Annex D.3
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master is expected to send Port Event "No Device" (0x1800) and to clear Device Events in DiagEntries of ArgBlock "PortStatusList" whenever Device gets COMLOST
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_AUTOSTART
Procedure	a) MTU_State_CheckOperate ;returns "MTU in OPERATE" b) Evaluation 1) c) DLL_ClearAllEvents ;clear Event entries d) MTU_Event_SetStatusCode(0x81) ;enters StatusCode in memory e) MTU_Event_SetSlot(1, 0xF4, 0x1800) ;enters in slot 1 Event Q and Code f) MTU_Event_TriggerAndWaitForAck ;returns "EventAck" (0x1800) g) Evaluation 2) h) DLL_GetDeviceEvents(1) ;minED=1, returns "DeviceEventList1" i) Evaluation 3) j) SMI_PortStatus ;returns "ArgBlock PortStatusList1" k) Evaluation 4) l) MTU_State_Deactivate ;disable MTU response m) TM_AWAIT_PORT_STATUS(NO_DEVICE) n) DLL_GetPortEvents(1) ;minEP=1, returns "PortEventList1" o) Evaluation 5) p) SMI_PortStatus ;returns "ArgBlock PortStatusList2" q) Evaluation 6) r) MTU_State_Activate ;enable MTU response s) TM_AWAIT_PORT_STATUS(OPERATE) t) DLL_GetPortEvents(1) ;minEP=1, returns "PortEventList2" u) Evaluation 7) v) SMI_PortStatus ;returns "ArgBlock PortStatusList3" w) Evaluation 8)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "MTU in OPERATE" 2) Check "EventAck" 3) Check "DeviceEventList1" 4) Check "ArgBlock PortStatusList1" 5) Check "PortEventList1" 6) Check "ArgBlock PortStatusList2" 7) Check "PortEventList2" 8) Check "ArgBlock PortStatusList3"
Test passed	"MTU in OPERATE" = TRUE, and EventAck: StatusCodeRead = TRUE, AllSlotsRead = TRUE, Acknowledge = TRUE, and DeviceEventList1.Entry1 (EventQualifier = 0xF4, EventCode = 0x1800), and PortStatusList1: NumberOfDiags = 1, DiagEntry0 (EventQualifier = 0xF4, EventCode = 0x1800), and

	PortEventList1: Entry1 (EventQualifier = 0xFC, EventCode = 0x1800), and PortStatusList2: NumberOfDiags = 1, DiagEntry0 (EventQualifier = 0xFC, EventCode = 0x1800), and PortEventList2: Entry1 (EventQualifier = 0x5C, EventCode = 0xFF26), or Entry1 (EventQualifier = 0xBC, EventCode = 0x1800), and PortStatusList3: NumberOfDiags = 0
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.12 Event in PREOPERATE

Table 293 defines the test conditions for this test case.

Table 293 – Event in PREOPERATE

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0351
Name	TCM_ALIC_EVNT_PREOPERATE
Purpose (short)	MTU sends Event in PREOPERATE
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master Event test: test to pass
Specification (clause)	[6], Annex D.3
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event with details (StatusCode type 2) in PREOPERATE. Master transfers the EventCode and EventQuaifier to the upper-level system. Master acknowledges the Event. MTU resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_DI
Procedure	a) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* b) MTU_Event_SetSlot(1, 0xF4, 0x1000) *;enters in slot 1 Event Q and Code* c) MTU_Event_SetFlag *;set Event flag in M-sequence CKS* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) TM_AWAIT_PORT_STATUS(OPERATE) f) DLL_GetDeviceEvents(1) *;minED=1, returns "EventList"* g) Evaluation 1) h) SMI_PortStatus *;returns "ArgBlock PortStatusList"* i) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "EventList" 2) Check "ArgBlock PortStatusList"
Test passed	EventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1000), and PortStatusList.NumberOfDiags = 1, PortStatusList.DiagEntry0 (EventQualifier = 0xF4, EventCode = 0x1000)
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.13 Event in PORT_DIAG

Table 294 defines the test conditions for this test case.

Table 294 – Event in PORT_DIAG

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0352
Name	TCM_ALIC_EVNT_PORT_DIAG
Purpose (short)	MTU sends Event in PORT_DIAG
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master Event test: test to pass
Specification (clause)	[6], Annex D.3
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event with details (StatusCode type 2) in PORT_DIAG. Master transfers the EventCode and EventQuaifier to the upper level system. Master acknowledges the Event. MTU resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_DI
Procedure	a) MTU_Event_SetStatusCode(0x81) *;enters StatusCode in memory* b) MTU_Event_SetSlot(1, 0xF4, 0x1800) *;enters in slot 1 Event Q and Code* c) MTU_Event_SetFlag *;set Event flag in M-sequence CKS* d) SMI_PortConfiguration(ABPS_TYPE_COMP <VendorID = 1>) *;invalid VendorID* e) TM_AWAIT_PORT_STATUS(PORT_DIAG) f) DLL_GetDeviceEvents(1) *;minED=1, returns "DeviceEventList"* g) Evaluation 1) h) DLL_GetPortEvents(1) *;minEP=1, returns "PortEventList"* i) Evaluation 2) j) SMI_PortStatus *;returns "ArgBlock PortStatusList"* k) Evaluation 3)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "DeviceEventList" 2) Check "PortEventList" 3) Check "ArgBlock PortStatusList"
Test passed	DeviceEventList.Entry1 (EventQualifier = 0xF4, EventCode = 0x1800), and PortEventList.Entry1 (EventQualifier = 0xFC, EventCode = 0x1802), and PortStatusList.NumberOfDiags = 2, PortStatusList.DiagEntry0 (EventQualifier = 0xF4, EventCode = 0x1800), PortStatusList.DiagEntry1 (EventQualifier = 0xFC, EventCode = 0x1802)
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.13.14 Invalid cycle time requested

Table 295 defines the test conditions for this test case.

Table 295 – Invalid cycle time requested

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0353
Name	TCM_ALIC_EVNT_INVALID_CYCLE_TIME
Purpose (short)	Master sends Port Event 0x6000 due to a requested invalid cycle time
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master Event test: test to pass
Specification (clause)	[6], Annex D.3
Configuration / setup	Master-Tester-System

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	MTU is configured to a minimal cycle time of 10 ms. Master port is configured to a cycle time of 4 ms. Since the requested cycle time cannot be achieved, the Master enters PORT_DIAG and sends the Port Event 0x6000 "Invalid cycle time".
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_DI
Procedure	a) MTU_DPP1_Set(MinCycleTime = 10 ms) b) SMI_PortConfiguration(ABPS_TYPE_COMP <PortCycleTime = 4 ms>) c) TM_AWAIT_PORT_STATUS(PORT_DIAG) d) DLL_GetPortEvents(1) *;minEP=1, returns "PortEventList"* e) Evaluation 1) f) SMI_PortStatus *;returns "ArgBlock PortStatusList"* g) Evaluation 2)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "PortEventList" 2) Check "ArgBlock PortStatusList"
Test passed	EventList.Entry1 (EventQualifier = 0xFC, EventCode = 0x6000), and PortStatusList.NumberOfDiags = 1 PortStatusList.DiagEntry0 (EventQualifier = 0xFC, EventCode = 0x6000)
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14 Data Storage (DS)

8.14.1 General

Some test cases need cleared Data Storage as a precondition to perform the test. One possibility is the re-configuration of the Master Port.

8.14.2 Delete stored DS object after reconfiguration

Table 296 defines the test conditions for this test case.

Table 296 – Delete stored DS object after reconfiguration

-CR027- -CR059-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0354
Name	TCM_ALIC_STOR_DELETEDSAFTERRECONF
Purpose (short)	Delete DS data object after Port reconfiguration
Equipment under test (EUT)	Master and Master Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.3, Figure 101, Table E.3, Annex G, Table G.2
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Any relevant change of Port configuration ("PortConfigList") via SMI service causes the stored DataStorageObject to be cleared. Port reconfigurations to different PortModes are tested during test procedure loops (TPL): TPL1: Change to PortMode = DI_C/Q *;see [6], Table E.3* TPL2: Change to PortMode = DO_C/Q TPL3: Change to PortMode = IOL_AUTOSTART TPL4: Change to PortMode = IOL_MANUAL and DeviceID = 0x002BD4, Validation&Backup = "4" *;Restore* TPL5: Change to PortMode = DEACTIVATED TPL6: Change to PortMode = IOL_MANUAL and Validation&Backup = "4" *;Restore*
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "ConfigList" b) TM_MASTER_UPLOAD (PARSET1) *;prepare EUT and MTU for identical DS data* c) SMI_DSToParServ *;returns "DS_Data" (uploaded)* d) Evaluation 1) e) SMI_PortConfiguration(<ConfigList>) *;change Port configurationTC_361* f) SMI_DSToParServ *;returns "DS_Data" (cleared)* g) Evaluation 2) h) Repeat from b) with next "ConfigList"

Test parameter	ConfigList = {
	APBS_PORTTODI,
	APBS_PORTTODO,
	APBS_NOTYPE_CHECK,
	ABPS_TYPE_COMP(<DeviceID = 0x002BD4>,<Validation&Backup = "4”>), ABPS_PORTINACTIVE,
	ABPS_TYPE_COMP(<Validation&Backup = "4">)
	}

Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check ArgBlock "DS_Data" 2) Check ArgBlock "DS_Data"

TEST CASE RESULTS	CHECK / REACTION

Test passed	For all TPL:
	DataStorageObject = PARSET1 in 1), and
	DataStorageObject = no data/invalid (Header = "0") and length = 12 octets in 2)

Test not passed (examples)	Any evaluation failed or communication error
Report	All evaluations <ok | nok>

8.14.3 Data Storage size limits (quantities)

Table 297 defines the test conditions for this test case.

Table 297 – Data Storage size limits (quantities)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0355
Name	TCM_ALIC_STOR_MAXSIZELIMITS
Purpose (short)	Check the maximum size limits of DS upload/download
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Checks the maximum conditions of DS behavior with respect to Index_List and DS memory size. Using the two data sets in "Test parameter", DS works correctly during the 2 test procedure loops (TPL): TPL1: Maximum permitted size of the DS data object and maximum length of Index_List (MAXINDEXLIST) TPL2: Maximum length of data objects (MAXDATA)
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config" b) MTU_DS_SetMaxDataStorage(<config>) *;provide MTU with max DS data* c) SMI_PortConfiguration(ABPS_TYPE_COMP <Validation&Backup = "3">) d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_DS_CheckUpload *;returns "upload sequence performed"* f) Evaluation 1 g) SMI_DSToParServ *;returns "DS_Data" (uploaded)* h) Evaluation 2 i) Repeat from b) with next "config"
Test parameter	config = {MAXINDEXLIST, MAXDATA} See A.4.8
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check "upload sequence performed" 2) Check ArgBlock "DS_Data"
Test passed	TPL1: "upload sequence performed" = TRUE, and DataStorageObject = MAXINDEXLIST TPL2: "upload sequence performed" = TRUE, and DataStorageObject = MAXDATA
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.4 Write consistent DS data object from parameter server

Table 298 defines the test conditions for this test case.

Table 298 – Write consistent DS data object from parameter server

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0356
Name	TCM_ALIC_STOR_DSFROMPARAMSERV
Purpose (short)	Write consistent DS data object from upper level system
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Annex G
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Upper level system writes DS data object with header information via service SMI_ParServToDS. There is no difference between current stored DS data object and written DS data object (match of checksum). Master does not perform any upload or download activities at identical DS data.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;prepare EUT and MTU for identical DS data* b) SMI_ParServToDS (PARSET1) *;write identical DS data (checksum match)* c) MTU_DS_CheckDownload *;returns "download sequence performed"* d) Evaluation 1)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "download sequence performed"
Test passed	"download sequence performed" = FALSE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.5 Write inconsistent DS data object from parameter server

Table 299 defines the test conditions for this test case.

Table 299 – Write inconsistent DS data object from parameter server

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0357
Name	TCM_ALIC_STOR_INCONSDSFROMPARSERV
Purpose (short)	Write inconsistent DS data object from upper-level system (checksum mismatch)
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Upper-level system writes DS data object with header information via service SMI_ParServToDS. There is a difference between current stored DS data object and written DS data object (mismatch of checksum). This causes the Master to restart the Port and to download DS data.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;prepare EUT and MTU for identical DS data* b) SMI_ParServToDS(PARSET2) *;write different DS data (checksum mismatch)* c) TM_AWAIT_PORT_STATUS (OPERATE) *;wait until OPERATE* d) MTU_State_CheckPreoperate *;returns "PREOPERATE"* e) Evaluation 1) f) MTU_DS_CheckDownload *;returns "download sequence performed"* g) Evaluation 2) h) MTU_DS_CheckParameterSet(PARSET2) *;returns "PARSET2 active"* i) Evaluation 3)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "PREOPERATE" 2) Check "download sequence performed" 3) Check "PARSET2 active"
Test passed	"PREOPERATE" = PASSED, and "download sequence performed" = TRUE, and "PARSET2 active" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.6 Write DS data object to improper Port configuration

Table 300 defines the test conditions for this test case.

Table 300 – Write DS data object to improper Port configuration

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0358
Name	TCM_ALIC_STOR_DSTOIMPROPPORTCONF
Purpose (short)	Write DS data object from upper-level system to improper Port configuration
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3, Annex G
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	Write DS data from upper-level system via SMI_ParServToDS. In case of improper Port configuration (PortMode = DI_C/Q, DO_C/Q, IOL_AUTOSTART, or VendorID and/or DeviceID ≠ values in DS data header), the service returns a negative response, and no download activity occurs.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) *;improper configuration for DS* b) TM_AWAIT_PORT_STATUS(OPERATE) *;wait until OPERATE* c) SMI_ParServToDS(PARSET2) *;PortMode prevents from performance* d) Evaluation 1) e) MTU_DS_CheckDownload *;returns "download sequence performed"* f) Evaluation 2)
Test parameter	–
Post condition / next test	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check returned ArgBlock 2) Check "download sequence performed"
Test passed	ArgBlock "Job Error" received, JobError.ErrorCode/AdditionalCode = 0x4039 (INCONSISTENT_DS_DATA), and "download sequence performed" = FALSE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.7 Upload request Event in OPERATE (Backup + Restore)

Table 301 defines the test conditions for this test case.

Table 301 – Upload request Event in OPERATE (Backup + Restore)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0359
Name	TCM_ALIC_STOR_UPLOADREQEVENTOPERBACKREST
Purpose (short)	Detection of upload request Event in OPERATE and Backup + Restore
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	MTU generates a hidden "DS_UPLOAD_REQ" Event while in OPERATE. Master Port configuration is proper for DS (PortMode and Validation&Backup = "3"). Master uploads parameter values from Device.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;prepare EUT and MTU for identical DS data* b) MTU_Set_ParameterSet(PARSET2) *;MTU activates PARSET2* c) MTU_DS_SetUpload(Event) *;MTU generates DS_UPLOAD_REQ Event* d) TM_AWAIT(2000) *;wait 2 sec* e) MTU_DS_CheckUpload *;returns "upload sequence performed"* f) Evaluation 1) g) SMI_DSToParServ *;returns "DS_Data" (uploaded)* h) Evaluation 2)
Test parameter	–
Post condition / next test	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	1) Check "upload sequence performed" 2) Check ArgBlock "DS_Data"
Test passed	"upload sequence performed" = TRUE, and DataStorageObject = PARSET2
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.8 Upload request Event in OPERATE (Restore)

Table 302 defines the test conditions for this test case.

Table 302 – Upload request Event in OPERATE (Restore)

-CR086-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0360
Name	TCM_ALIC_STOR_UPLOADREQEVENTOPERREST
Purpose (short)	Detection of upload request Event in OPERATE and Restore
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	MTU generates a hidden "DS_UPLOAD_REQ" Event while in OPERATE. Master Port configuration is proper for DS (PortMode and Validation&Backup = "4"). Master downloads parameter values to Device.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;prepare EUT and MTU for identical DS data* b) SMI_PortConfiguration (ABPS_TYPE_COMP,< Validation&Backup = "4">) c) TM_AWAIT_PORT_STATUS (OPERATE) d) MTU_DS_Set_Parameter(PARSET2) *;MTU activates PARSET2* c) MTU_DS_SetUpload(Event) *;MTU generates DS_UPLOAD_REQ Event* d) TM_AWAIT(2000) *;wait 2 sec* e) MTU_DS_CheckDownload *;returns "download sequence performed"* f) Evaluation 1) g) MTU_DS_CheckParameter(PARSET1) *;returns "PARSET1 active"* h) Evaluation 2) i) SMI_DSToParServ *;returns "DS_Data" (uploaded)* j) Evaluation 3)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "download sequence performed" 2) Check "PARSET1 active" 2) Check ArgBlock "DS_Data"
Test passed	"download sequence performed" = TRUE, and "PARSET1 active" = TRUE, and DataStorageObject = PARSET1
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.9 Upload request Event in OPERATE (Backup + Restore inactive)

Table 303 defines the test conditions for this test case.

Table 303 – Upload request Event in OPERATE (Backup + Restore inactive)

-CR028- -CR029- -CR056- -CR086-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0361
Name	TCM_ALIC_STOR_UPLOADREQEVENTOPERBACKRESTINACT
Purpose (short)	Detection of upload request Event in OPERATE and Backup + Restore inactive
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	MTU generates a hidden "DS_UPLOAD_REQ" Event while in OPERATE. Master Port configuration is improper for DS (PortMode = IOL_AUTOSTART and PortMode = IOL_MANUAL) and Validation&Backup = "0" or "2"). Master does not show upload or download activities.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;prepare EUT and MTU for identical DS data* b) SMI_PortConfiguration (<ConfigList>) *;improper configuration for DS* c) TM_AWAIT_PORT_STATUS (OPERATE) d) MTU_Set_ParameterSet(PARSET2) *;MTU activates PARSET2* e) MTU_DS_SetUpload(Event) *;MTU generates DS_UPLOAD_REQ Event* f) TM_AWAIT(2000) *;wait 2 sec* g) MTU_DS_CheckUpload *;returns "upload sequence performed"* h) Evaluation 1) i) MTU_DS_CheckDownload *;returns "download sequence performed"* j) Evaluation 2) k) SMI_DSToParServ *;returns "DS_Data" (uploaded)* l) Evaluation 3) m) repeat from b) with next “ConfigList”
Test parameter	ConfigList = {APBS_NOTYPE_CHECK, ABPS_TYPE_COMP(<Validation&Backup = "0">), ABPS_TYPE_COMP(<VendorID = 0xFDE8>, <DeviceID = 0x002BD2>, <Validation&Backup = "2">)}
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For each ConfigList entry 1) Check "upload sequence performed" 2) Check "download sequence performed" 3) Check ArgBlock "DS_Data"
Test passed	For each ConfigList entry "upload sequence performed" = FALSE, and "download sequence performed" = FALSE, and DataStorageObject = no data/invalid (Header = "0") and length = 12 octets
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.10 Start-up with empty DS (DS deactivated)

Table 304 defines the test conditions for this test case.

Table 304 – Start-up with empty DS (DS deactivated)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0362
Name	TCM_ALIC_STOR_STARTUPEMPTDSDEACTIV
Purpose (short)	Start-up with empty/invalid DS data object and deactivated Data Storage
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	For Port start-up an empty or invalid DS data object in Master is assumed as well as two different improper Port configurations. No upload takes place in both Test Procedure Loops: TPL1: PortMode = IOL_AUTOSTART (no type compatibility check) TPL2: PortMode = IOL_MANUAL (no Backup, no Restore)
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config" b) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* c) MTU_DS_SetUpload(Flag_ON) *;set DS_UPLOAD_FLAG* d) SMI_PortConfiguration(<config>) e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_DS_CheckDSCommands *;returns "DS_Commands performed"* g) Evaluation 1) h) SMI_DSToParServ *;returns "DS_Data"* i) Evaluation 2) j) Repeat from b) with next "config"
Test parameter	config = {APBS_NOTYPE_CHECK, ABPS_TYPE_COMP(< Backup&Restore = "2">)}
Post condition / next test	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	For all TPL: 1) Check "DS_Commands performed" 2) Check ArgBlock "DS_Data"
Test passed	For all TPL: "DS_Commands performed" = FALSE, and DataStorageObject = no data/invalid (Header = "0") and length = 12 octets
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.11 Start-up with empty DS (DS activated – DS Upload)

Table 305 defines the test conditions for this test case.

Table 305 – Start-up with empty DS (DS activated – DS Upload)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0363
Name	TCM_ALIC_STOR_STRTUPEMPTYDSDSACTIV
Purpose (short)	Start-up with empty/invalid DS data object and activated Data Storage
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	For Port start-up an empty or invalid DS data object in Master is assumed as well as four different proper Port configurations. Upload takes place in all four Test Procedure Loops, where PortMode = IOL_MANUAL: TPL1: Validation&Backup = "3", DS_UPLOAD_FLAG = "0" *;Backup+Restore* TPL2: Validation&Backup = "3", DS_UPLOAD_FLAG = "1" *;Backup+Restore* TPL3: Validation&Backup = "4", DS_UPLOAD_FLAG = "0" *;Restore* TPL4: Validation&Backup = "4", DS_UPLOAD_FLAG = "1" *;Restore*
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config", "flag" b) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* c) MTU_DS_SetUpload(<flag>) *;Test parameter* d) SMI_PortConfiguration(ABPS_TYPE_COMP, <Validation&Backup = "config">) *;Test parameter* e) TM_AWAIT_PORT_STATUS(OPERATE) f) MTU_DS_CheckUpload *;returns "upload sequence performed"* g) Evaluation 1) h) SMI_DSToParServ *;returns "DS_Data"* i) Evaluation 2) j) Repeat from b) with next "config", "flag"

Test parameter	config = {3, 3, 4, 4} *;Type compatiple Device V1.1 (Backup+Restore or Restore)*
	flag = {Flag_off, Flag_on, Flag_off, Flag_on} *;DS_UPLOAD_FLAG*

Post condition / next test	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	For all TPL: 1) Check "upload sequence performed" 2) Check ArgBlock "DS_Data"
Test passed	For all TPL: "upload sequence performed" = TRUE, and DataStorageObject = PARSET2
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.12 Start-up with stored DS (Device replacement – DS Download)

Table 306 defines the test conditions for this test case.

Table 306 – Start-up with stored DS (Device replacement – DS Download)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0364
Name	TCM_ALIC_STOR_STRTUPDEVREPLACEDSDOWN
Purpose (short)	Start-up after Device replacement with DS Download (checksum mismatch)
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU
TEST CASE	**CONDITIONS / PERFORMANCE**
Purpose (detailed)	For Port start-up a stored or valid DS data object in Master is assumed as well as three different proper Port configurations. Download takes place in all three Test Procedure Loops, where PortMode = IOL_MANUAL, DS = activated, and DS check-sum = mismatch: TPL1: Validation&Backup = "3", DS_UPLOAD_FLAG = "0" *;Backup+Restore* TPL2: Validation&Backup = "4", DS_UPLOAD_FLAG = "0" *;Restore* TPL3: Validation&Backup = "4", DS_UPLOAD_FLAG = "1" *;Restore*
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config", "flag" b) MTU_Set_Parameter(PARSET1) *;MTU activates PARSET1* c) SMI_PortConfiguration(ABPS_TYPE_COMP, <Validation&Backup = "config**"**>) *;Test parameter* d) TM_AWAIT_PORT_STATUS(OPERATE) e) SMI_DSToParServ *;returns DataStorageObject* f) MTU_State_Deactivate *;disable response to Master request* g) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* h) MTU_DS_SetUpload(<flag>) *;Test parameter* i) MTU_ State_Activate *;enable response to Master request* j) TM_AWAIT_PORT_STATUS(OPERATE) k) MTU_DS_CheckDownload *;returns "download sequence performed"* l) Evaluation 1 m) MTU_DS_CheckParameter(PARSET1) *;returns "PARSET1 active"* n) Evaluation 2 o) Repeat from b) with next "config", " flag"

Test parameter	config = {3, 4, 4} *;Type compatiple Device V1.1 (Backup+Restore or Restore)*
	flag = {Flag_off, Flag_off, Flag_on} *;DS_UPLOAD_FLAG*

Post condition / next test	–
TEST CASE RESULTS	**CHECK / REACTION**
Evaluation	For all TPL 1) Check "download sequence performed" 2) Check "PARSET1 active"
Test passed	For all TPL "download sequence performed" = TRUE, and "PARSET1 active" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.13 Start-up with stored DS (Device replacement – no DS Download)

Table 307 defines the test conditions for this test case.

Table 307 – Start-up with stored DS (Device replacement – no DS Download)

-CR079-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0365
Name	TCM_ALIC_STOR_STRTUPDEVREPLACENODSDOWN
Purpose (short)	Start-up after Device replacement without DS Download (checksum match)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Table E.3
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	For Port start-up a stored or valid DS data object in Master is assumed and the Device is replaced by a type compatible one. No Download takes place in all four Test Procedure Loops: TPL1: PortMode = IOL_MANUAL, Validation&Backup = "3" *;Backup+Restore* TPL2: PortMode = IOL_MANUAL, Validation&Backup = "4" *;Restore* TPL3: PortMode = IOL_AUTOSTART *;no type compatibility check* TPL4: PortMode = IOL_MANUAL, Validation&Backup = "2" *;no Backup, no Restore*
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config" b) MTU_Set_Parameter(PARSET1) *;MTU activates PARSET1* c) SMI_PortConfiguration(<config>) *;Test parameter* d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_State_Deactivate *;disable response to Master request* f) MTU_State_Activate *;enable response to Master request* g) TM_AWAIT_PORT_STATUS(OPERATE) h) MTU_DS_CheckDSComands *;returns "DS_Commands performed"* i) Evaluation 1 j) MTU_DS_CheckParameter(PARSET1) *;returns "PARSET1 active"* k) Evaluation 2 l) Repeat from b) with next "config"
Test parameter	config= {ABPS_TYPE_COMP, <Validation&Backup = "3">, ABPS_TYPE_COMP, <Validation&Backup = "4">, APBS_NO_TYPE_CHECK, ABPS_TYPE_COMP, <Validation&Backup = "2">}
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check "DS_Commands *performed* " 2) Check "PARSET1 active"
Test passed	"DS_Commands *performed*" = FALSE, and "PARSET1 active" = TRUE
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.14 Start-up with stored DS (Device replacement – DS Upload)

Table 308 defines the test conditions for this test case.

Table 308 – Start-up with stored DS (Device replacement – DS Upload)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0366
Name	TCM_ALIC_STOR_STRTUPDEVREPLACEDSUPFLAG
Purpose (short)	Start-up after Device replacement with raised DS upload flag
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	For Port start-up a stored or valid DS data object in Master is assumed and the Device is replaced by a type compatible one setting DS_UPLOAD_FLAG. Upload takes place at proper PortConfiguration (PortMode = IOL_MANUAL, Validation&Backup = "3")
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DS_SetParameter(PARSET1) *;MTU activates PARSET1* b) SMI_PortConfiguration(ABPS_TYPE_COMP, Validation&Backup = "3") c) TM_AWAIT_PORT_STATUS(OPERATE) d) MTU_State_Deactivate *;disable response to Master request* e) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* f) MTU_DS_SetUpload(<Flag_on>) *;set DS_UPLOAD_FLAG* g) f) MTU_State_Activate *;enable response to Master request* h) TM_AWAIT _PORT_STATUS(OPERATE) i) MTU_DS_CheckUpload *;returns "upload sequence performed"* j) Evaluation 1) k) SMI_DSToParServ *;returns "DS_Data"* l) Evaluation 2)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "upload sequence performed" 2) Check ArgBlock "DS_Data"
Test passed	"upload sequence performed" = TRUE, and DataStorageObject = PARSET2
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.15 Start-up with stored DS (Device replacement – no DS Upload)

Table 309 defines the test conditions for this test case.

Table 309 – Start-up with stored DS (Device replacement – no DS Upload)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0367
Name	TCM_ALIC_STOR_STRTUPDEVREPLACEDSFLAGNOUPLOAD
Purpose (short)	Start-up after Device replacement with raised DS upload flag – DS upload blocked
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	For Port start-up a stored or valid DS data object in Master is assumed and the Device is replaced by a type compatible one setting DS_UPLOAD_FLAG. No Upload takes place in all three Test Procedure Loops: TPL1: PortMode = IOL_MANUAL, Validation&Backup = "4" ;Restore TPL2: PortMode = IOL_MANUAL, Validation&Backup = "2" ;no Backup, no Restore TPL3: PortMode = IOL_AUTOSTART ;no type compatibility check
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) Assign first value to "config" b) MTU_DS_SetParameter(PARSET1) ;MTU activates PARSET1 c) SMI_PortConfiguration(<config>) ;Test parameter d) TM_AWAIT_PORT_STATUS(OPERATE) e) MTU_State_Deactivate ;disable response to Master request f) MTU_DS_SetParameter(PARSET2) ;MTU activates PARSET2 g) MTU_DS_SetUpload(<Flag_on>) ;set DS_UPLOAD_FLAG h) f) MTU_State_Activate ;enable response to Master request i) TM_AWAIT_PORT_STATUS(OPERATE) j) MTU_DS_CheckUpload ;returns "upload sequence performed" k) Evaluation 1) l) SMI_DSToParServ ;returns "DS_Data" m) Evaluation 2) n) Repeat from b) with next "config"
Test parameter	config = {ABPS_TYPE_COMP, <Validation&Backup = "4">, ABPS_TYPE_COMP, <Validation&Backup = "2">, APBS_NO_TYPE_CHECK}
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	For all TPL: 1) Check "upload sequence performed" 2) Check ArgBlock "DS_Data"
Test passed	For all TPL: "upload sequence performed" = FALSE, and DataStorageObject ≠ PARSET2
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.16 Corrupted DS Index_List

Table 310 defines the test conditions for this test case.

Table 310 – Corrupted DS Index_List

-CR062--CR075-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0368
Name	TCM_ALIC_STOR_DSINDLISTCORRUPT
Purpose (short)	DS error in case of read Index_List fault during Port start-up
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Annex E.16
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	An error occurs while reading an Index_List during Port start-up. This leads to the following consequences: - No Upload/Download occurs (user view) - MTU is on hold in PREOPERATE - PortEvent is generated
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;PARSET1 uploaded to Master DS* b) DLL_GetPortEvents *;clear Event entries* c) MTU_State_Deactivate *;disable response to Master request* d) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* e) MTU_DS_SetUpload(<Flag_on>) *;set DS_UPLOAD_FLAG* f) MTU_DS_SetError(IndexList_err) *;MTU falsifies Index_List* g) MTU_State_Activate *;enable response to Master request* h) DLL_GetPortEvents(3) *;minED=3, returns "PortEventList* i) Evaluation 1) j) MTU_DS_CheckUpload *;returns "upload sequence performed"* k) Evaluation 2) l) MTU_State_CheckPreoperate *;returns "PREOPERATE"* m) Evaluation 3) n) SMI_PortStatus *;returns "PortStatusList"* o) Evaluation 4) p) SMI_PortConfiguration(ABPS_PORTTODI) *;switch Port to DI mode* q) DLL_GetPortEvents(1) *;minED=1, returns "PortEventList* r) Evaluation 5) s) SMI_PortStatus *;returns "PortStatusList"* t) Evaluation 6)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortEventList" 2) Check "upload sequence performed" 3) Check "PREOPERATE" 4) Check ArgBlock "PortStatusList" 5) Check ArgBlock "PortEventList" 6) Check ArgBlock "PortStatusList"

TEST CASE RESULTS	CHECK / REACTION
Test passed	PortEventList.Entry1 (EventQualifier: Master/Port, Error, Event appears, EventCode = 0x180C) , and "upload sequence performed" = FALSE, and "PREOERATE" = REACHED, and PortStatusList: PortStatusInfo = PORT_DIAG, DiagEntryx = result of 1), and PortEventList.Entry1 (EventQualifier: Master/Port, Error, Event disappears, EventCode = 0x180B), and PortStatusList: PortStatusInfo = DI, DiagEntryx = empty
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.17 DS Download fault

Table 311 defines the test conditions for this test case.

Table 311 – DS Download fault

-CR062-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0369
Name	TCM_ALIC_STOR_DSDOWNLOADFAULT
Purpose (short)	DS error during DS download at Port start-up
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4, Annex E.16
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	An error occurs while downloading DS data object during Port start-up, for example a read Index returns negative response. This leads to Download aborted, MTU on hold in PREOPERATE, PortEvent generated, rollback of DS data object.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1 *;PARSET1 uploaded to Master DS* b) DLL_GetPortEvents *;clear Event entries* c) MTU_State_Deactivate *;disable response to Master request* d) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* e) MTU_DS_SetError(W_Index19_err) *;MTU to return negative response* f) MTU_State_Activate *;enable response to Master request* g) MTU_DS_WAIT(DS_BREAK) *;wait until Download aborted* h) Evaluation 1) i) DLL_GetPortEvents(3) *;minED=3, returns"PortEventList* j) Evaluation 2) k) MTU_DS_CheckParameter(PARSET1) *;returns "PARSET1 active"* l) Evaluation 3) m) MTU_State_CheckPreoperate *;returns "PREOPERATE"* n) Evaluation 4) o) SMI_PortStatus *;returns "PortStatusList"* p) Evaluation 5) q) SMI_PortConfiguration(ABPS_PORTTODI) *;switch Port to DI mode* r) DLL_GetPortEvents(1) *;minED=1, returns "PortEventList* s) Evaluation 6) t) SMI_PortStatus *;returns "PortStatusList"* u) Evaluation 7)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check DS_BREAK 2) Check ArgBlock "PortEventList" 3) Check "PARSET1 active" 4) Check "PREOPERATE" 5) Check ArgBlock "PortStatusList" 6) Check ArgBlock "PortEventList" 7) Check ArgBlock "PortStatusList"
Test passed	DS_BREAK detected, and PortEventList.Entry1 (EventQualifier: Master/Port, Error, Event appears, EventCode = 0x180D, and "PARSET1 active" = TRUE, and "PREOPERATE" = REACHED, and PortStatusList: PortStatusInfo = PORT_DIAG, DiagEntryx = result of 1), and PortEventList.Entry1 (EventQualifier: Master/Port, Error, Event disappears, EventCode = 0x180D), and

TEST CASE RESULTS	CHECK / REACTION
	PortStatusList: PortStatusInfo = DI, DiagEntryx = empty
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

2925

8.14.18 DS Upload fault

Table 312 defines the test conditions for this test case.

Table 312 – DS Upload fault

-CR062-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0370
Name	TCM_ALIC_STOR_DSUPLOADFAULT
Purpose (short)	DS error during DS Upload at Port start-up
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	An error occurs while uploading DS data object during Port start-up, for example a read Index returns negative response. This leads to Upload aborted, MTU on hold in PREOPERATE, PortEvent generated, rollback of DS data object.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;PARSET1 uploaded to Master DS* b) DLL_GetPortEvents *;clear Event entries* c) MTU_State_Deactivate *;disable response to Master request* d) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* e) MTU_DS_SetUpload(<Flag_on>) *;set DS_UPLOAD_FLAG* f) MTU_DS_SetError(R_Index19_err) *;MTU to return negative response* g) MTU_State_Activate *;enable response to Master request* h) MTU_DS_WAIT(DS_BREAK) *;wait until Download aborted* i) Evaluation 1) j) DLL_GetPortEvents(3) *;minED=3, returns"PortEventList* k) Evaluation 2) l) SMI_DSToParServ ;*returns "DS_Data"* m) Evaluation 3) n) MTU_State_CheckPreoperate *;returns "PREOPERATE"* o) Evaluation 4) p) SMI_PortStatus *;returns "PortStatusList"* q) Evaluation 5) r) SMI_PortConfiguration(ABPS_PORTTODI) *;switch Port to DI mode* s) DLL_GetPortEvents(1) *;minED=1 , returns"PortEventList* t) Evaluation 6) u) SMI_PortStatus *;returns "PortStatusList"* v) Evaluation 7)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check DS_BREAK 2) Check ArgBlock "PortEventList" 3) Check ArgBlock "DS_Data" 4) Check "PREOPERATE" 5) Check ArgBlock "PortStatusList" 6) Check ArgBlock "PortEventList" 7) Check ArgBlock "PortStatusList"

TEST CASE RESULTS	CHECK / REACTION
Test passed	DS_BREAK detected, and PortEventList.Entry1 (EventQualifier: Master/Port, Error, Event appears, EventCode = 0x180C), and DataStorageObject = PARSET1, and "PREOPERATE" = REACHED, and PortStatusList: PortStatusInfo = PORT_DIAG, DiagEntryx = result of 1), and PortEventList.Entry1 (EventQualifier: Master/Port, Error,…, Event disappears, EventCode = 0x180C), and PortStatusList: PortStatusInfo = DI, DiagEntryx = empty
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

2932

8.14.19 Incompatible Device and DS

Table 313 defines the test conditions for this test case.

Table 313 – Incompatible Device and DS

-CR062--CR091--CR080-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0371
Name	TCM_ALIC_STOR_INCOMPDEVICE&DS
Purpose (short)	DS in case of identification fault (incompatible Device)
Equipment under test (EUT)	Master + Port
Test case version	1.1
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	At start-up, the Master detects a Device, not type compatible with configured data. This leads to PortEvent generated, no DS Upload/Download, MTU is on hold in PREOPERATE and shows diagnosis information.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD(PARSET1) *;PARSET1 uploaded to Master DS* b) DLL_GetPortEvents *;clear Event entries* c) MTU_State_Deactivate *;disable response to Master request* d) MTU_ DPP1_Set(DeviceID = 0x102BD2) *;MTU gets different DeviceID* e) MTU_DS_SetParameter(PARSET2) *;MTU activates PARSET2* f) MTU_DS_SetUpload(<Flag_on>) *;set DS_UPLOAD_FLAG* g) MTU_State_Activate *;enable response to Master request* h) DLL_GetPortEvents(3) *;minED=3, returns"PortEventList* i) Evaluation 1) j) MTU_DS_CheckDSCommands*;* *;returns "DS_Command performed"* k)Evaluation 2) l) SMI_PortStatus *;returns "PortStatusList"* m) Evaluation 3) n) SMI_DSToParServ *;returns "DS_Data"* o) Evaluation 4) p) SMI_PortConfiguration(ABPS_PORTTODI) *;switch Port to DI mode* q) DLL_GetPortEvents(1) *;minED=1 , returns"PortEventList* r) Evaluation 5) s) TM_AWAIT_PORT_STATUS(DI) *;returns "PortStatusList"* t) Evaluation 6)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortEventList" 2) Check "DS_Command" 3) Check ArgBlock "PortStatusList" 4) Check ArgBlock "DS_Data" 5) Check ArgBlock "PortEventList" 6) Check ArgBlock "PortStatusList"
Test passed	PortEventList.Entry1 EventQualifier: Master/Port, Error, Event appears, EventCode = 0x1803), and "DS_Commands performed" = FALSE, and PortStatusList: PortStatusInfo = PORT_DIAG, DiagEntryx = result of 1), and DataStorageObject = PARSET1, and PortEventList.Entry1 (EventQualifier: : Master/Port, Error, Event disappears,EventCode = 0x1803, and PortStatusList: PortStatusInfo = DI, DiagEntryx = empty
Test not passed (examples)	Evaluation 1,2,3,4,5,6 failed

TEST CASE RESULTS	CHECK / REACTION
Report	All evaluations <ok | nok>

8.14.20 Master power OFF/ON (non-volatile DS data)

Table 314 defines the test conditions for this test case.

-CR030-

Table 314 – Master power OFF/ON (non-volatile DS data)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0372
Name	TCM_ALIC_STOR_POWEROFFONNONVOLATILEDS
Purpose (short)	Non-volatile storage of Port configuration and DS data
Equipment under test (EUT)	Master + Port
Test case version	1.0
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.3, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	After power off/on, the Master starts with the same configuration data and DS data as before the power cycle. These data are saved in non-volatile memory.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) TM_MASTER_UPLOAD (PARSET1) *;PARSET1 uploaded to Master DS* b) Switch off EUT power supply *;Depending on Tester implementation* c) Switch on EUT power supply d) TM_AWAIT_PORT_STATUS (OPERATE) e) SMI_DSToParServ *;returns "DS_Data"* f) Evaluation 1)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "DS_Data"
Test passed	DataStorageObject = PARSET1
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.14.21 DS Upload trial with locked Device Data Storage

Table 315 defines the test conditions for this test case.

Table 315 – DS Upload trial with locked Device Data Storage

-CR057--CR062-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0280
Name	TCM_ALIC_STOR_DSLOCKED
Purpose (short)	DS Upload trial with locked Device Data Storage
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master DS test: test to pass
Specification (clause)	[6], 11.2, 11.4, B.2.3, B.2.4
Configuration / setup	Master-Tester-System with MTU

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master shall detect the locked Data Storage of a Device. In this case the Master shall deny access for all data storage actions from its own data storage handler. NOTE Even it is highly recommended for Devices not to implement locking for DS, Devices in the field can show this feature and Master shall be able to handle it.
Precondition	EUT: EUT_PORT_DI MTU: MTU_STANDARD_STATE
Procedure	a) MTU_DS_Locked b) DLL_GetPortEvents *;clear Event entries* c) SMI_PortConfiguration(ABPS_TYPE_COMP <Validation&Backup = "3">) d) TM_AWAIT_PORT_STATUS (PORT_DIAG) e) MTU_DS_CheckDSCommands *;returns "DS_Command performed"* f) Evaluation 1) g) DLL_GetPortEvents(1) *;minED=1, returns "PortEventList* h) TM_AWAIT_PORT_EVENT i) Evaluation 2) j) SMI_DSToParServ *;returns "DS_Data"* k) Evaluation 3)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check "DS_Command performed" 2) Check ArgBlock "PortEvent" 3) Check "DS_Data"
Test passed	"DS_Command performed" = FALSE, and PortEventList.Entry1 (EventQualifier: INSTANCE = SYS, SOURCE = Master/Port, TYPE = Error, Mode = Event appears, EventCode = 0x180B), and "DS_Data" = no data/invalid (Header = "0") and length = 12 octets
Test not passed (examples)	Any evaluation failed
Report	All evaluations <ok | nok>

8.15 Legacy Device ("V1.0")

8.15.1 General

Since a Master designed according to [6] shall support legacy Devices designed according to [5], it shall pass the following test cases.

8.15.2 Master detects legacy Device and establishes connection

Table 316 defines the test conditions for this test case.

Table 316 – Master detects legacy Device and establishes connection

-CR077-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0285
Name	TCM_LGCY_MANY_DETECTANDCONNECT
Purpose (short)	Master detects legacy Device and establishes connection
Equipment under test (EUT)	Master + Port
Test case version	1.3
Category / type	Master legacy Device test: test to pass
Specification (clause)	[5], 7.2.2.1, 9.3.3; [6], 9.2.3.2, A.2.6, Figures 71, 72
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Compatibility of the startup phase of V1.1 Master and a V1.0 Device. Master shall detect connection to a V1.0 Device and shall adjust its startup behavior.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* d) Evaluation 1) e) MTU_Startup_Check10 *;returns "Startup1.0"* f) Evaluation 2)
Test parameter	–
Post condition /next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortStatusList" 2) Check "Startup1.0"
Test passed	PortStatusList: PortStatusInfo = OPERATE, TransmissionRate = COM2, MasterCycleTime (decoded) ≥ 4 ms, InputDataLength = 1, OutputDataLength = 1, VendorID = 0xFDE8, DeviceID = 0x002BD2, RevisionID = 0x10, and Startup1.0 = TRUE
Test not passed (examples)	Any evaluation failed, or Test_Timeout
Report	Start-up according to [5]: <yes/no> <ok | nok>

8.15.3 Master detects legacy Device and establishes interleave mode

Table 317 defines the test conditions for this test case.

Table 317 – Master detects legacy Device and establishes interleave mode

-CR077-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0286
Name	TCM_LGCY_MANY_DETECTANDINTERLEAVE
Purpose (short)	Master detects legacy Device and establishes interleave mode
Equipment under test (EUT)	Master + Port
Test case version	1.3
Category / type	Master legacy Device test: test to pass
Specification (clause)	[5], 9.3.3; [6], 9.2.3.5, A.2.6, Figures 71, 72
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Compatibility of the startup phase of V1.1 Master and a V1.0 Device using M-sequence TYPE_1 in interleave mode. Master shall detect connection to a V1.0 Device and shall adjust its startup behavior.
Precondition	MTU: MTU_STANDARD_STATE EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) MTU_DPP1_Set(InputLength = 3) c) MTU_DPP1_Set(OutputLength = 3) d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* f) Evaluation 1) g) MTU_Startup_Check10 *;returns "Startup1.0"* h) Evaluation 2)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortStatusList" 2) Check "Startup1.0"
Test passed	PortStatusList: PortStatusInfo = OPERATE, TransmissionRate = COM2, MasterCycleTime (decoded) ≥ 4 ms, InputDataLength = 3, OutputDataLength = 3, VendorID = 0xFDE8, DeviceID = 0x002BD2, RevisionID = 0x10, and Startup1.0 = TRUE
Test not passed (examples)	Any evaluation failed, or Test_Timeout
Report	Start-up according to [5]: <yes/no> <ok | nok> M-sequence TYPE_1 used: <yes/no> <ok | nok>

8.15.4 Master receives an Event without details (Warning)

Table 318 defines the test conditions for this test case.

Table 318 – Master receives an Event without details (Warning)

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0287
Name	TCM_ALIC_EVNT_NODETAILSWARNING
Purpose (short)	Master receives an Event without details
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master legacy Device test: test to pass
Specification (clause)	[5], 7.2.4.4.1 [6], 7.3.8.3, 8.3.3.1, 11.5, Annex A.6, Annex D
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master receives Event without details (StatusCode type 1, notification). Master extracts the EventCode (type 1), maps it accordingly into StatusCode (type 2) and transfers it to the upper-level system. Similarly, the StatusCode indicates with 1 bit the validity of the Process Data. The Master shall read this information as soon as possible to indicate the actual state of the Process Data to the upper system. The Master acknowledges the Event; the Master-Tester-Unit resets the Event flag.
Precondition	MTU: MTU_STANDARD_STATE, No Event in process EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* d) Evaluation 1) e) MTU_Event_SetStatusCode(0x01) f) MTU_Event_TriggerAndWaitForAck *;returns "EventAck"* j) Evaluation 2) g) DLL_GetDeviceEvents(1) *;minED = 1, returns "EventList"* h) Evaluation 3) k) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* l) Evaluation 4)
Test parameter	–
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortStatusList" 2) Check "EventAck" 3) Check "EventList" 4) Check ArgBlock "PortStatusList"
Test passed	PortStatusList: PortStatusInfo = OPERATE, RevisionID = 0x10, NumberOfDiags = 0, and EventAck: StatusCodeRead = TRUE, AllSlotsRead = TRUE, Ack. = TRUE, and EventList.Entry1 = 0xFF80, Application, notification, and PortStatusList: PortStatusInfo = OPERATE, RevisionID = 0x10, NumberOfDiags = 0 *;since notifications will not be placed into list*
Test not passed (examples)	Any evaluation failed, or Test-Timeout
Report	Propagated information to the upper-level system: <code> <ok | nok> Master acknowledgement: <code> <ok | nok> Event flag: <0/1> <ok | nok>

8.15.5 ISDU Write interrupted by an Event leads to a Write error

Table 319 defines the test conditions for this test case.

Table 319 – ISDU Write interrupted by an Event leads to a Write error

-CR081--CR099-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0289
Name	TCM_LGCY_MANY_EVENTINTERRUPTSISDU
Purpose (short)	ISDU Write interrupted by an Event leads to write error
Equipment under test (EUT)	Master + Port
Test case version	1.3
Category / type	Master legacy Device test: test to pass
Specification (clause)	[5], 7.3.6.3, Figure 47; [6], Table C.2, Annex C.3.8
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	According to [5], the legacy Device can cancel an ongoing ISDU transfer via an Event. This Event shall lead to a Read error or Write error on the Master side.
Precondition	MTU: MTU_STANDARD_STATE, no Event pending EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) c) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* d) Evaluation 1) e) MTU_Add_ISDU(Index = <Index>, Subindex = 0, Specialty = ISSUE_EVENT) f) SMI_DeviceWrite(ABPS_DEVICEWRITE(<Index>)) *;returns "JobError"* *;the Master initiates an ISDU Write with too large length of data. The legacy* *;Device sends an Event (DL, Error, Event single shot, EventCode = 0x5200)* *;during the ISDU service to abort it.* g) Evaluation 2) h) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* i) Evaluation 3)
Test parameter	<Index> = 251, data = all "0", data length = 33
Post condition / next test	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortStatusList" 2) Check ArgBlock "JobError" 3) Check ArgBlock "PortStatusList"
Test passed	PortStatusList: PortStatusInfo = OPERATE, RevisionID = 0x10, NumberOfDiags = 0, and JobError: ErrorCode = 0x80, AdditionalCode = 0x23, and PortStatusList: PortStatusInfo = OPERATE, RevisionID = 0x10, NumberOfDiags = 0
Test not passed (examples)	Any evaluation failed, or Test_Timeout, or Receive "JobError" does not terminate within 10 s, or Received positive result
Report	Event received by the upper-level system: <yes/no> <ok | nok> ISDU service aborted: <yes/no> <ok | nok>

8.15.6 Master transforms PD_invalid Event into appropriate propagation

Table 320 defines the test conditions for this test case.

Table 320 – Master transforms PD_invalid Event into appropriate propagation

-CR081-

TEST CASE ATTRIBUTES	IDENTIFICATION / REFERENCE
Identification (ID)	SDCI_TC_0290
Name	TCM_LGCY_MANY_PDINVALIDEVENT
Purpose (short)	Master transforms PD_invalid and PD_valid Event into appropriate propagation
Equipment under test (EUT)	Master + Port
Test case version	1.2
Category / type	Master legacy Device test: test to pass
Specification (clause)	[5], 7.2.4.4; [6], A.6.2, A.6.3
Configuration / setup	Master-Tester-Unit

TEST CASE	CONDITIONS / PERFORMANCE
Purpose (detailed)	Master transforms a "PD invalid" and a "PD_valid" Event from a legacy Device into an appropriate propagation form and passes Process Data correctly.
Precondition	MTU: MTU_STANDARD_STATE, No Event in process EUT: PORT_INACTIVE
Procedure	a) MTU_DPP1_Set(RevisionID = 0x10) b) MTU_DPP1_Set(PDIn length) = 0x83 *;"32" Bit* c) MTU_DPP1_Set(PDOut length) = 0x83 *;"32" Bit* d) SMI_PortConfiguration(ABPS_NOTYPE_CHECK) e) TM_AWAIT_PORT_STATUS(OPERATE) *;returns "PortStatusList"* f) Evaluation 1) g) MTU_State_MirrorPD *;perform PD mirroring* h) MTU_State_SetPDValidity(VALID) i) TM_AWAIT_PD_VALIDITY(VALID) *;return "PDIn"* j) Evaluation 2) k) SMI_PDOut(ABPS_PDOUT<OutputDataLength=4, PDO = [0x12,0x34,0x56,0x78]>) l) SMI_PDIn *;returns "PDIn"* m) Evaluation 3) n) MTU_State_SetPDValidity(INVALID) *;Device issues Event with PD Invalid* o) TM_AWAIT_PD_VALIDITY(INVALID) *;return "PDIn"* p) Evaluation 4) q) MTU_State_SetPDValidity(VALID) *;Device issues Event with PD Valid* r) TM_AWAIT_PD_VALIDITY(VALID) *;return "PDIn"* s) Evaluation 5)
Test parameter	–
Post condition	–

TEST CASE RESULTS	CHECK / REACTION
Evaluation	1) Check ArgBlock "PortStatusList" 2) Check ArgBlock "PDIn" 3) Check ArgBlock "PDIn" 4) Check ArgBlock "PDIn 5) Check ArgBlock "PDIn
Test passed	PortStatusList: PortStatusInfo = OPERATE, RevisionID = 0x10, and PDIn: PQI.PQ = VALID, and PDIn: PDI = [0x12,0x34,0x56,0x78], and PDIn: PQI.PQ = INVALID, and PDIn: PQI.PQ = VALID
Test not passed (examples)	Any evaluation failed, or Test_Timeout
Report	Correct propagation of PD_INVALID: <yes/no> <ok | nok> Correct propagation of PD_VALID: <yes/no> <ok | nok> Event acknowledged: <yes/no> <ok | nok>

8.16 Test report template

The template is defined by the Master-Tester. The test report shall present at least the reports of the test cases.

9 Environmental tests

9.1 General

Annex G in [6] defines the environmental tests (EMC) for the SDCI communication part of a Master/Device system. A passed EMC test is a precondition for a Manufacturer Declaration. It depends on the particular technology of a Device and the countries of deployment, whether additional environmental tests are necessary to achieve for example a CE mark for Europe.

9.2 Product specific standards

Usually, the product standard for a Master is the IEC 61131-2. For Devices, the major product standard is the IEC 60947-1.

9.3 EMC tests

EMC tests in respect to a particular phenomenon are defined in the IEC 61000-4-x series. Details for the execution are described in Annex G.2 in [6] and in 4.4 or 4.5 respectively.

Hint: Length "L" in Figures G.4 and G.8 in [6] shall be as short as possible.

9.4 Test report templates

9.4.1 Overview

Tests are required for the following phenomena:

- Electrostatic discharge (ESD: IEC 61000-4-2)
- Electromagnetic field (HF: IEC 61000-4-3)
- Fast transients (Burst: IEC 61000-4-4)
- Conducted radio frequency (CRF: IEC 61000-4-6)

The SDCI manufacturer declaration of conformity comprises EMC tests according to Annex G.2.4 in [6]. The following forms or any other document may be used as long as it contains the same information.

9.4.2 ESD

Figure 14 shows a proposed template for ESD tests.

Project:	
Test Item:	
Responsible Party:	
Tester:	
Applied Standard/Guideline:	
Type of Device:	
Test Location:	
Time Range:	

Test Requirements/Results:				
Type of discharge	Requirement fulfilled?		Achieved ImmunityTest Voltage kV	PerformanceCriterion
	yes	no		
Contact discharge	□	□		
Air discharge	□	□		
HCP	□	□		
VCP	□	□		

Result:

Test requirements are □ fulfilled

□ not fulfilled

Every single requirement must be met.

Remarks:

____

____

____

____

____

Enclosures:

____

____

______________________________ ______________________________

Date Tester’s Signature

Figure 14 – Proposed template for ESD tests

9.4.3 HF

Figure 15 shows a proposed template for HF tests.

Project:	
Test Item:	
Responsible Party:	
Tester:	
Applied Standard/Guideline:	
Type of Device:	
Test Location:	
Time Range:	

Test Requirements/Results:				
Type of HF Field	Requirement fulfilled?		Achieved Immunity Test Field V/m	Performance Criterion
	yes	no		
	□	□		
	□	□		
	□	□		
	□	□		

Result:

Test requirements are □ fulfilled

□ not fulfilled

Every single requirement must be met.

Remarks:

____

____

____

____

____

Enclosures:

____

____

____________________ ____________________________

Date Tester’s Signature

Figure 15 – Proposed template for HF tests

9.4.4 Burst

Figure 16 shows a proposed template for Burst tests.

Project:	
Test Item:	
Responsible Party:	
Tester:	
Applied Standard/Guideline:	
Type of Device:	
Test Location:	
Time Range:	

Test Requirements/Results:				
Type of burst	Requirement fulfilled?		Achieved Immunity Test Voltage kV	Performance Criterion
	yes	no		
power supply lines	□	□		
data lines	□	□		
	□	□		
	□	□		

Result:

Test requirements are □ fulfilled

□ not fulfilled

Every single requirement must be met.

Remarks:

____

____

____

____

____

Enclosures:

____

____

Date

Tester’s Signature

Figure 16 – Proposed template for Burst tests

9.4.5 Conducted RF

Figure 17 shows a proposed template for conducted RF tests.

Project:	
Test Item:	
Responsible Party:	
Tester:	
Applied Standard/Guideline:	
Type of Device:	
Test Location:	
Time Range:	

Test Requirements/Results:				
Type of Frequency MHz	Requirement fulfilled?		Achieved Immunity Test Voltage V	Performance Criterion
	yes	no		
0,15 to 80	□	□		

Result:

Test requirements are □ fulfilled

□ not fulfilled

Every single requirement must be met.

Remarks:

____

____

____

____

____

Enclosures:

____

____

______________________________ ______________________________

Date Tester's Signature

Figure 17 – Proposed template for conducted RF tests

9.4.6 Explanation of template terms

The terms in the templates are defined as follows:

- "Project" means for example the name of an SDCI Device.
- "Test Item" means the name and order number of the particular Device under test.
- "Responsible Party" means the manufacturer or a third-party company who takes responsibility for the Device.
- "Tester" means the full name of the test person in charge.
- "Applied standards or guidelines" shall comprise at least [6] and a product standard such as IEC 60947-1
- "Type of Device" identifies the type of the device thus indicating the appropriate level of EMC test. Possible types are "open type", "cabinet" or "enclosed type".
- "Test Location" indicates the name and address of the EMC test laboratory.
- "Time Range" indicates the date and the duration of the test.

Annex A
(normative)
Test configurations and test tools

A.1 Test configurations

A.1.1 Overview

The test cases for the physical layer tests and data link layer tests can be performed with the help of

- A variable power supply between 20 V and 30 V
- Discrete components such as capacitors and resistors according to the test case
- A voltmeter (accuracy of 2 %) and a current meter (accuracy of 10 %)
- An oscilloscope for Wake-up pulses and eye-diagrams
- A logic analyzer for message timings
- A line-monitor to record protocol sequences
- A Reference-Master and a Reference-Device

A.1.2 Measurement circuits for electrical isolation

A.1.2.1 Measurement of Master Port class B isolation

Figure A.1 shows the measurement circuit for electrical isolation of Master Port class B.

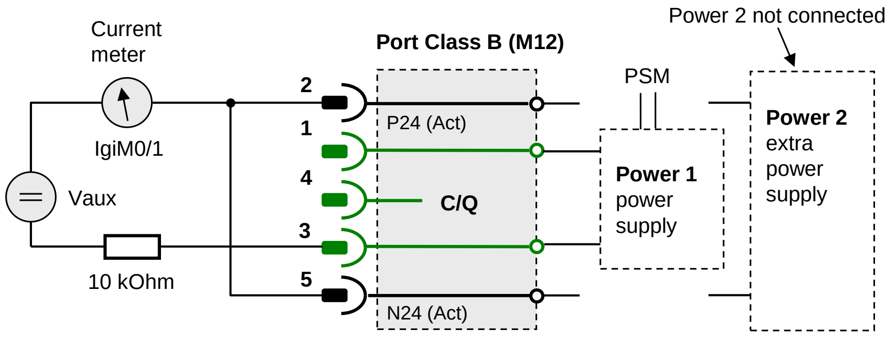


Figure A.1 – Measurement of Master Port class B isolation

A.1.2.2 Measurement of Device isolation

Figure A.2 shows the measurement circuit for electrical isolation of Device power supplies.

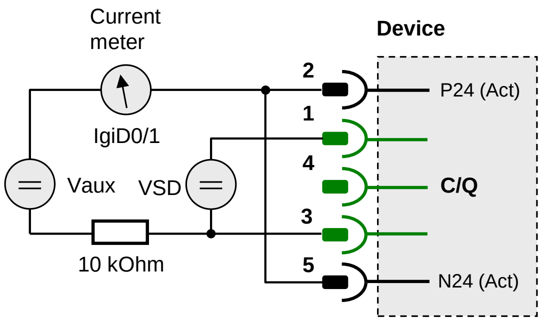


Figure A.2 – Measurement of Device isolation

A.1.3 Measurement circuits for the physical layer tests

A.1.3.1 Measurement of static parameters

-CR041-

Figure A.3 shows the measurement circuit diagram for static parameters with the help of a voltmeter.

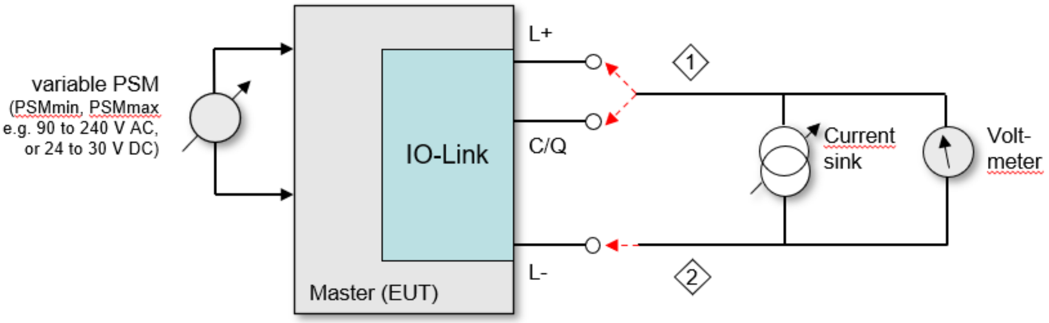


Figure A.3 – Measurement circuit diagram for static parameters

A.1.3.2 Measurement of power supply behavior

Figure A.4 shows the circuit diagram for the measurement of the power-on behavior of Devices.

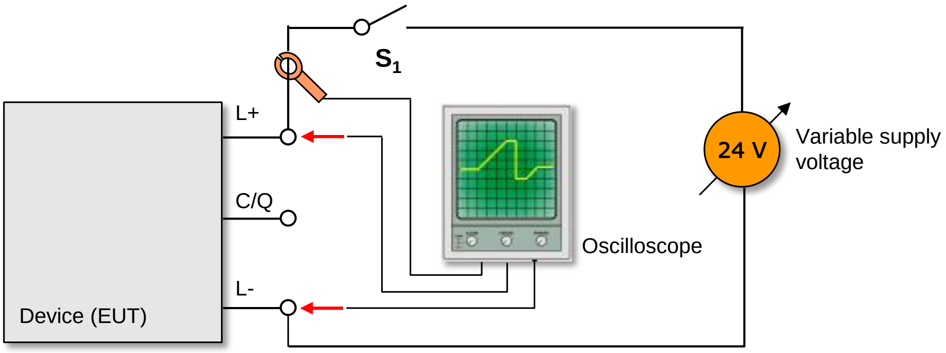


Figure A.4 – Measurement circuit diagram for power supply behavior

A.1.3.3 Measurement of dynamic parameters

Figure A.5 shows the measurement circuit diagram for dynamic parameters with the help of an oscilloscope.

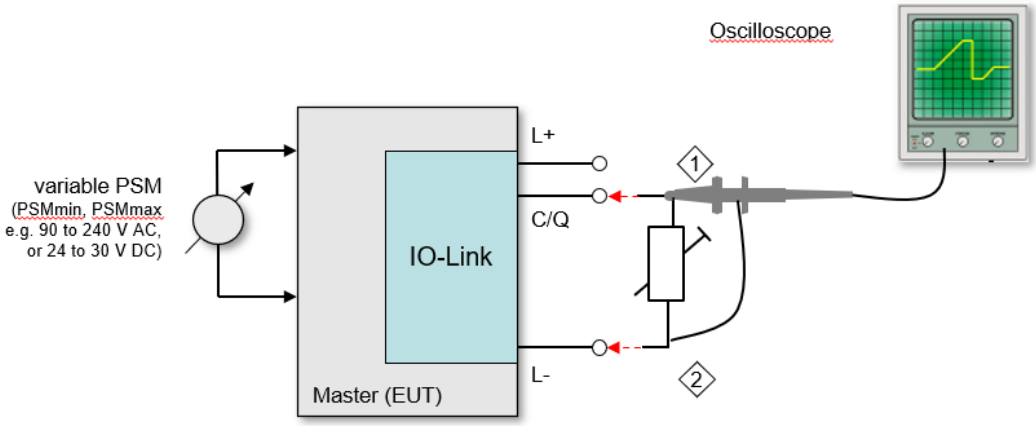


Figure A.5 – Measurement circuit diagram for dynamic parameters

A.1.3.4 Measurement of Master input thresholds

Figure A.6 shows the measurement circuit diagram for Master input thresholds with the help of an auxiliary variable voltage and a voltmeter.

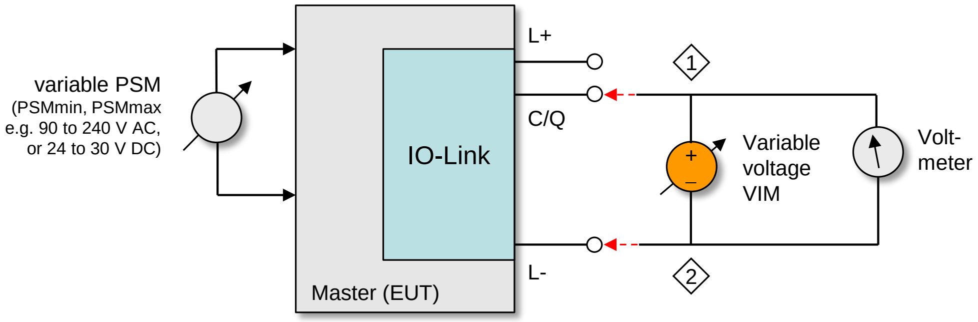
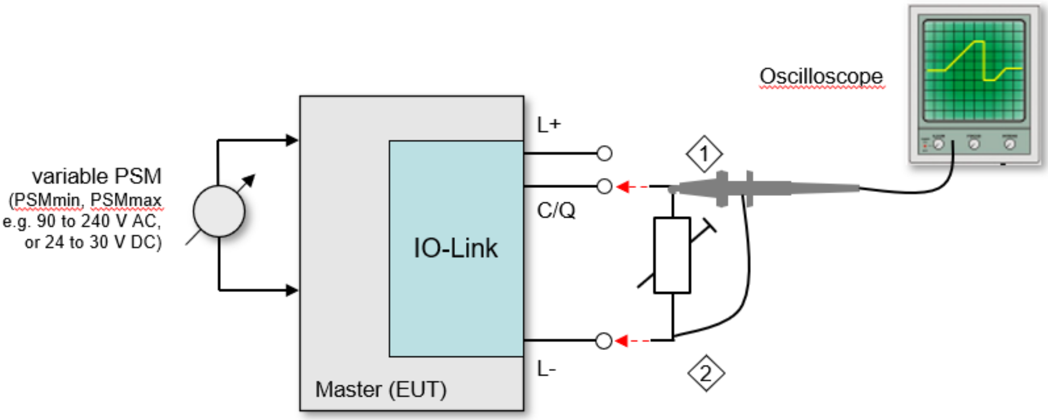


Figure A.6 – Measurement circuit diagram for input thresholds

A.1.3.5 Measurement of Wake-up requests (high)

Figure A.7 shows the measurement circuit diagram for Wake-up requests with the help of an oscilloscope if the steady state level (of a Device) is high.

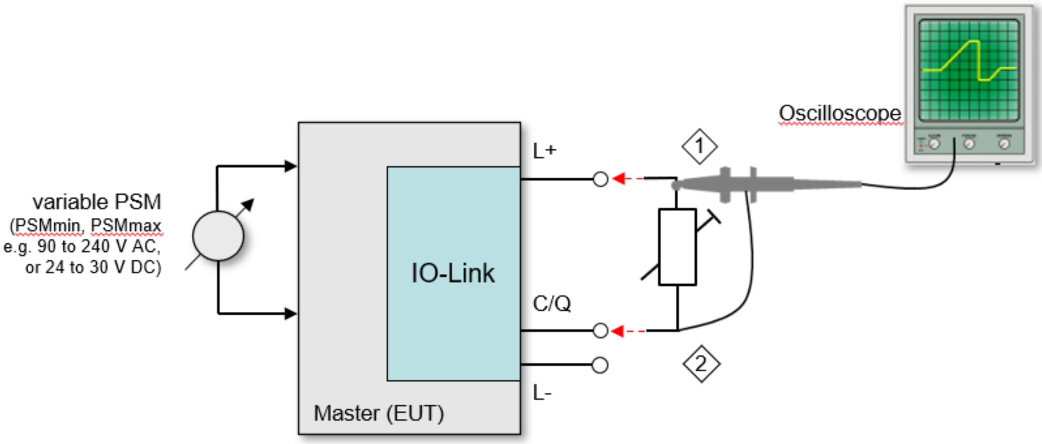


-CR041-

Figure A.7 – Measurement circuit diagram for Wake-up requests (high)

A.1.3.6 Measurement of Wake-up requests (low)

Figure A.8 shows the measurement circuit diagram for Wake-up requests with the help of an oscilloscope if the steady state level (of a Device) is low.



-CR041-

Figure A.8 – Measurement circuit diagram for Wake-up requests (low)

A.1.3.7 Measurement of return time delay to SIO mode

Figure A.9 shows the circuit diagram for measurements of the delay time of a Device to return to SIO-mode with the help of an oscilloscope.

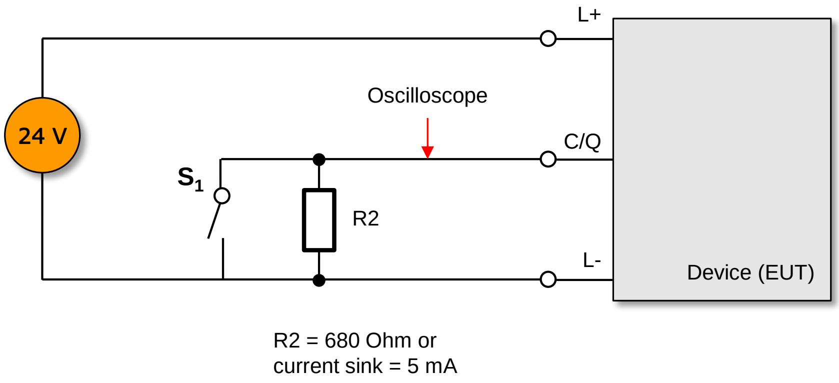


Figure A.9 – Measurement of return time delay to SIO mode

A.1.3.8 Measurement of dynamic parameters (Device input threshold)

Figure A.10 shows the measurement circuit diagram for Device input thresholds with the help of an auxiliary variable voltage and a voltmeter.

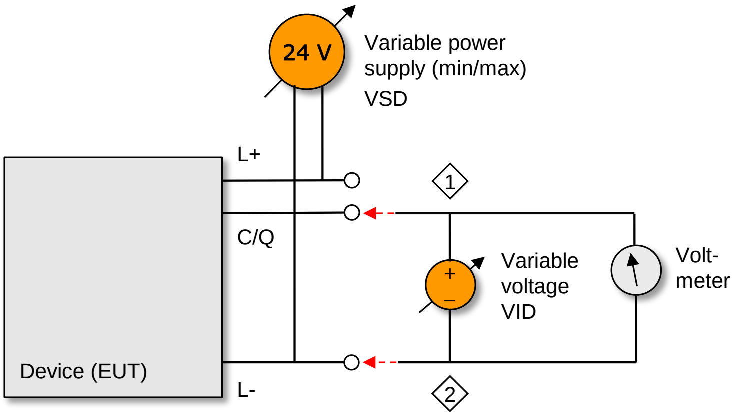


Figure A.10 – Measurement circuit diagram for input thresholds

A.1.3.9 Measurement of Wake-up requests (timing)

Figure A.11 shows the measurement circuit diagram for the timing of Wake-up requests with the help of an oscilloscope.

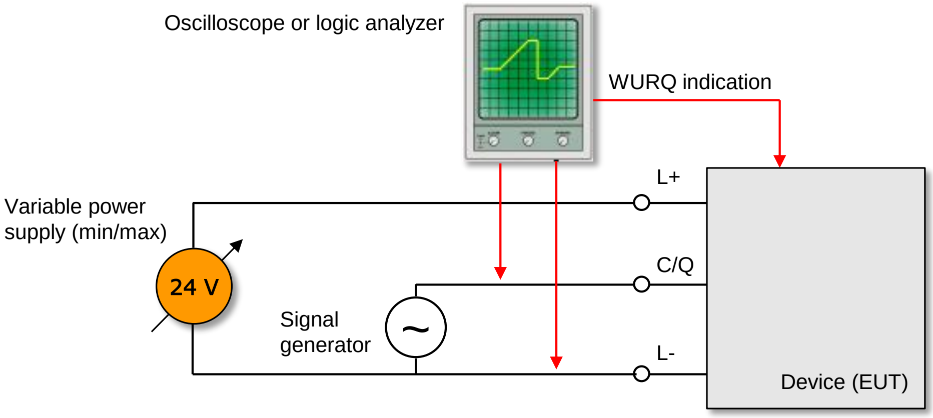


Figure A.11 – Measurement circuit diagram for Wake-up request timings

A.1.3.10 Measurement of Receive Enable after Wake-up (C/Q high)

Figure A.12 shows the circuit diagram for the measurement timing of t_{REN} (receive enable delay) with the help of an oscilloscope in case of C/Q = high.

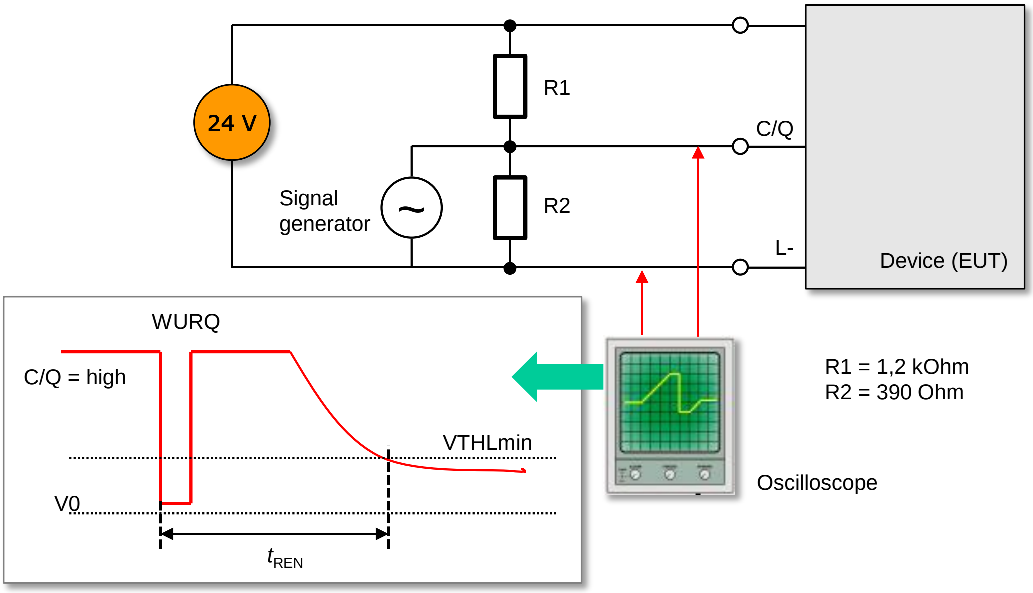


Figure A.12 – Measurement circuit diagram for timing t_{REN} (C/Q high)

A.1.3.11 Measurement of Receive Enable after Wake-up (C/Q low)

Figure A.13 shows the circuit diagram for the timing measurement of t_{REN} (receive enable delay) with the help of an oscilloscope in case of C/Q =low.

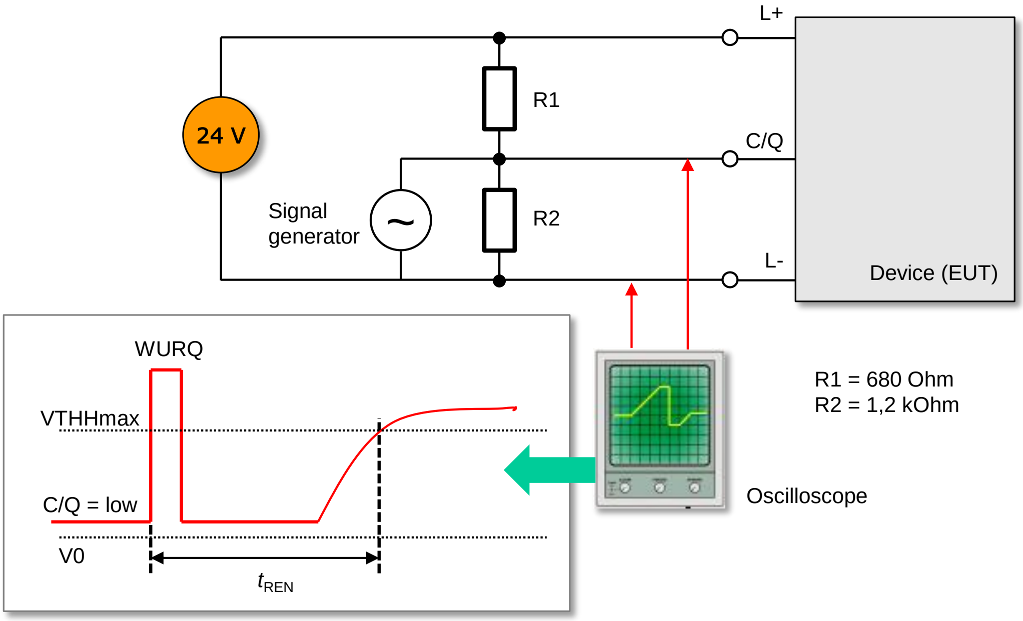


Figure A.13 – Measurement circuit diagram for timing t_{REN} (C/Q low)

A.1.3.12 Measurement of start-up and readiness timings

Figure A.14 shows the measurement circuit diagram for start-up and readiness timings with the help of an oscilloscope or a logic analyzer.

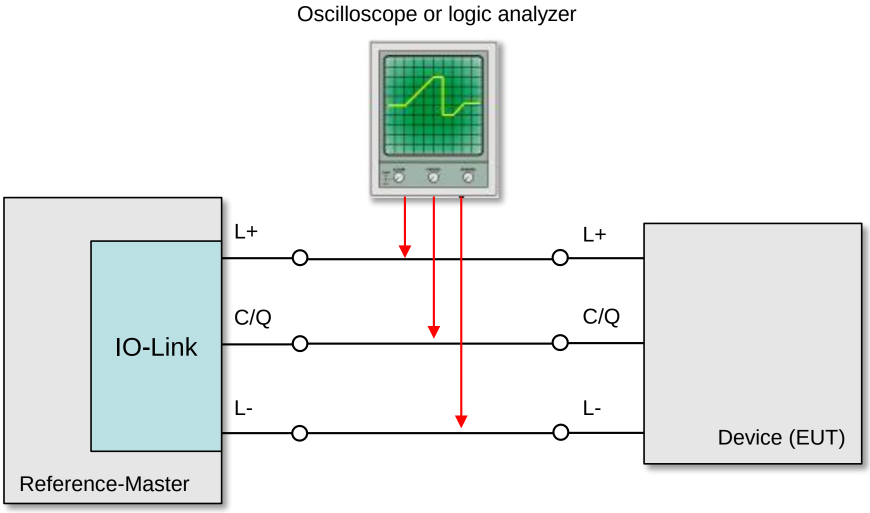


Figure A.14 – Measurement of start-up and readiness timings

A.1.3.13 Eye diagram measurement using a line simulation

Figure A.15 shows the circuit diagram for the eye diagram measurements using a line simulation as required load.

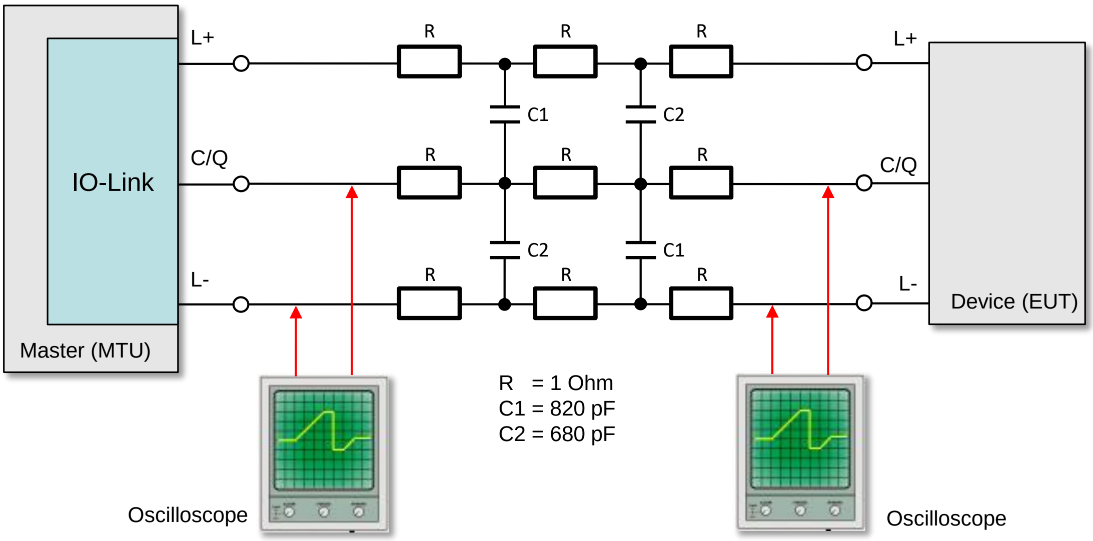


Figure A.15 – Eye diagram measurement using a line simulation

A.1.4 Protocol recording via a Line-Monitor

Usually, the test cases assume that a test passed if data are written or read in the expected manner. Sometimes it is easier to observe the protocol steps with the help of a Line-Monitor that lists the Master request messages and the Device response messages in a convenient manner on the screen of a laptop. Figure A.16 shows the principle.

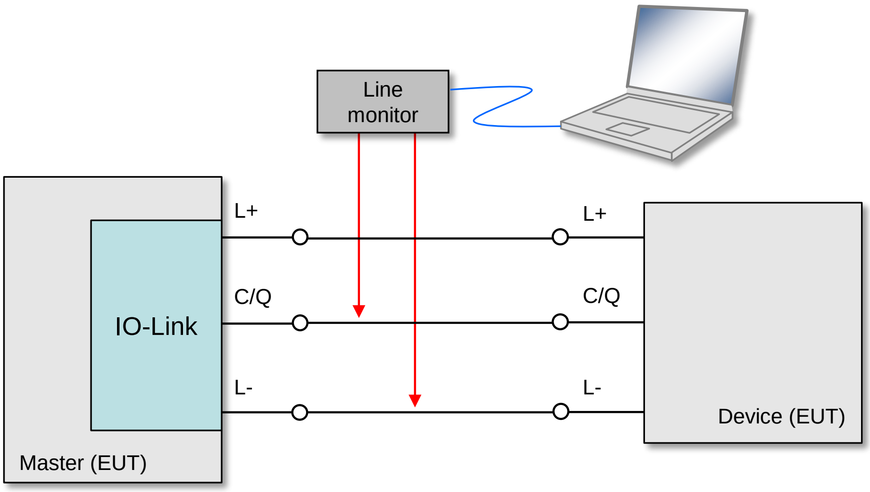


Figure A.16 – Message recording via Line-Monitor

A.1.5 Requirements for a Reference-Master and a Reference-Device

The features in Table A.1 are mandatory for a Reference-Master as referred to in the physical layer tests of Devices.

Table A.1 – Mandatory features of a Reference-Master

Number	Feature	Reference
FRM1	Full compliance with respect to the signal on C/Q	[6]
FRM2	VSM voltage adjustable in the range from 20V to 30V	Master with nominal 24 V
FRM3	CQM selectable between 500 pF and 1 nF	[6], Table 6
FRM4	Functionality of the following SMI services: a) SMI_PortPowerOffOn(ABPS_PORTPOWERON), b) SMI_PortConfiguration(ABPS_NO_TYPE_CHECK), c) SMI_PortConfiguratoin(ABPS_PORT_INACTIVE) d) SMI_PortConfiguration(ABPS_TYPE_COMP)	[6], 11.2 and Annex E For parameter sets see Table A.10

The features in Table A.2 are mandatory for a Reference-Device as referred to in the physical layer tests of Masters.

Table A.2 – Mandatory features of a Reference-Device

Number	Feature	Reference
FRD1	Full compliance with respect to the signal on C/Q	[6]
FRD2	Transmission rates selectable between COM2 and COM3	[6], Table 9
FRD3	CQD selectable between 500 pF, 1 nF, and 10 nF	[6], Table 7

A.1.6 Diagrams for evaluations

Figure A.17 demonstrates how to determine maximum and minimum durations of Wake-up pulse high.

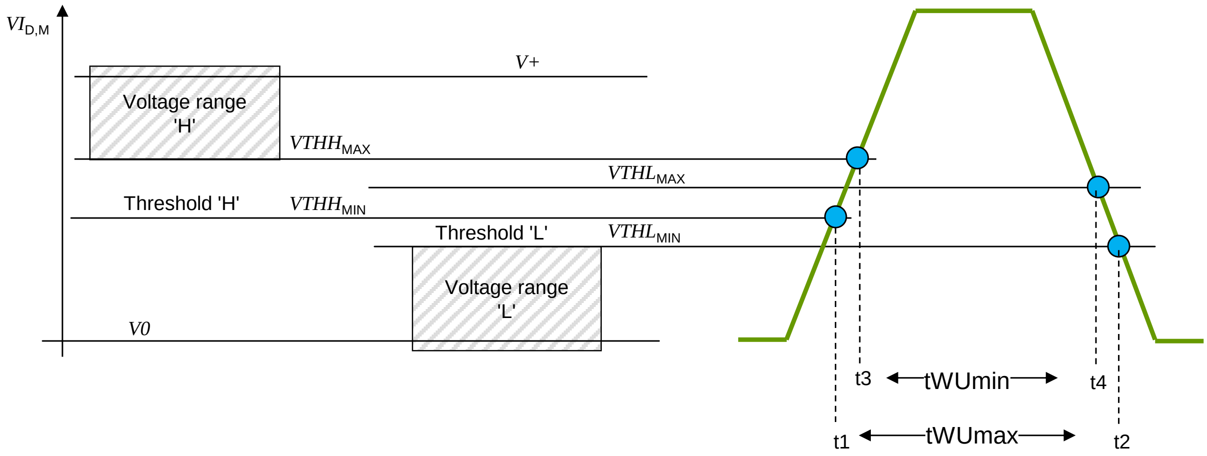


Figure A.17 – Evaluation of Wake-up pulse duration high

Figure A.18 demonstrates how to determine maximum and minimum durations of Wake-up pulse low.

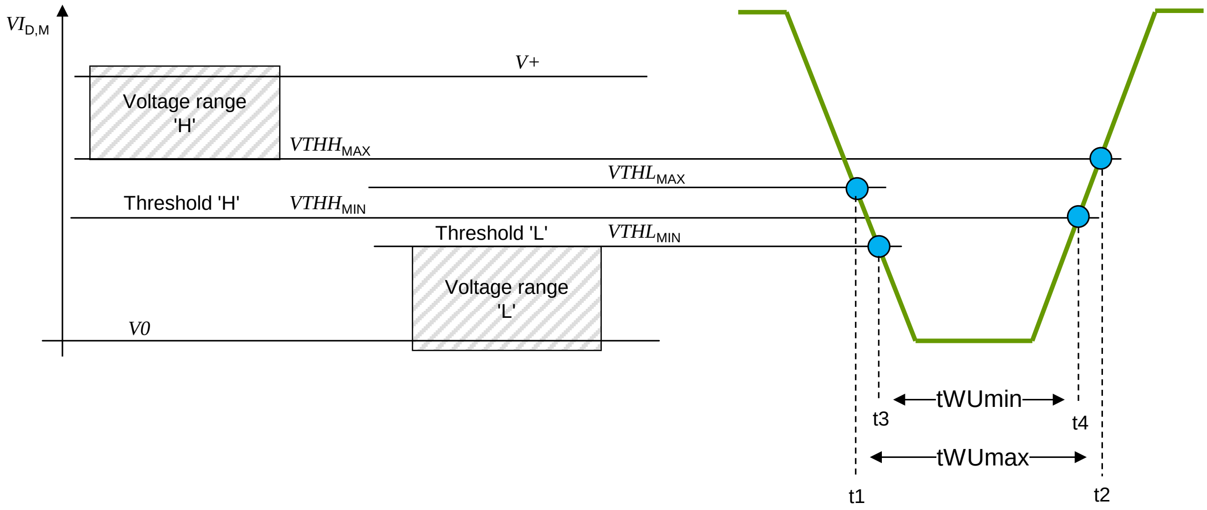


Figure A.18 – Evaluation of Wake-up pulse duration low

A.2 Device-Tester-System

A.2.1 Overview

To facilitate the tests of Devices and to ensure highest levels of conformity, several tools and the associated requirements (see Table A.3) are defined. These tools shall be type-approved by the organization mentioned in Annex D prior to any conformity testing for a manufacturer declaration.

A.2.2 Test principle and requirements

Figure A.19 shows the principle of a Device-Tester-System comprising

- A Device-Tester-Unit hardware with at least one SDCI port, which can be a modified standard Master with an adequate communication interface to a personal computer,
- A personal computer supporting the communication interface of the Device-Tester-Unit hardware,
- A Device-Tester-Program running on that personal computer serving as a control and monitorring program for the Device-Tester-Unit hardware,

- A Device – the EUT – that shall be tested for conformity.

Table A.3 – Requirements for the Device-Tester-System

Requirement	Description
SR1	The Device-Tester system shall execute and evaluate the test cases defined in this specification. This can include some functions or behavior not defined in the SDCI specification but is necessary to run the EUT into a specific state, e.g. generation of checksum errors.
SR2	The result of each test case and additional information about the test execution shall be reported to the user (test report, log file). The user shall be able to store and print this information.
SR3	The conformity test cases shall be secured against manipulation.
SR4	Optional requirement: The Device-Tester can interpret a valid IODD and generate different settings which are required for the conformity test. In case of absence of the IODD file there shall be a possibility to edit the settings manually.

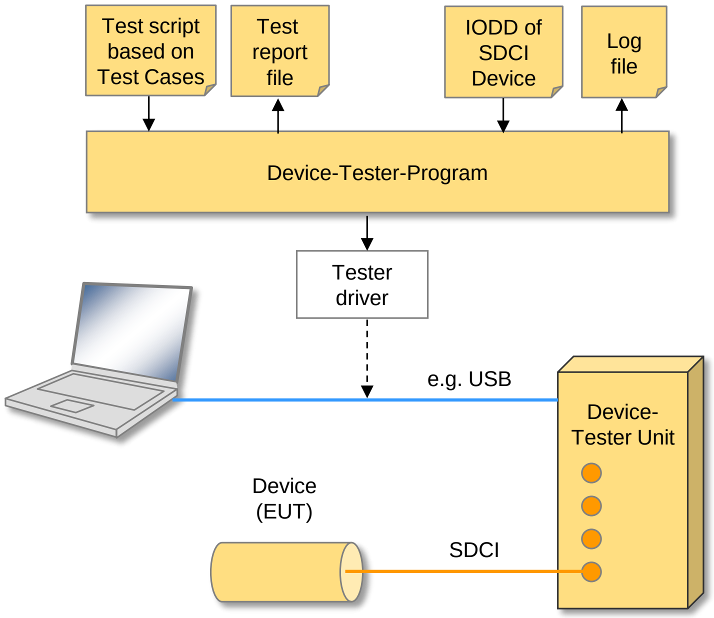


Figure A.19 – Principle of a Device-Tester-System

A.3 Master-Tester-System and approach to Master testing

A.3.1 Master-Tester using common communication interfaces

Figure A.20 shows the principle of a Master-Tester-System comprising

- A Master-Tester-Unit (MTU), which is a configurable and observable Device emulator with an IO-Link Port and with any communication interface to a personal computer, e.g. USB (Universal Serial Bus),
- A personal computer supporting the communication interface of the MTU and a communication interface to the SMI services (SMI Test Communication Server) of the Master to be tested (EUT = Equipment Under Test),
- A Master-Tester-Program running on that personal computer serving as a control and monitoring program for the MTU as well as for the EUT,
- A Master – the EUT – that shall be tested for conformity and that provides at least one Port and usually a communication interface based on Ethernet to an upper level system or in this case to the personal computer.

To perform a test case, the Master-Tester-Program uses SMI service instances with appropriate ArgBlock parameters (ABPS) to establish preconditions for the EUT (Master) and to step through the procedure.

In addition, the Master-Tester-Program also communicates with the Master-Tester-Unit using "Control/Status" instructions via a common interface, e.g. USB. These MTU-Instructions are identified by a characteristic and intuitive name.

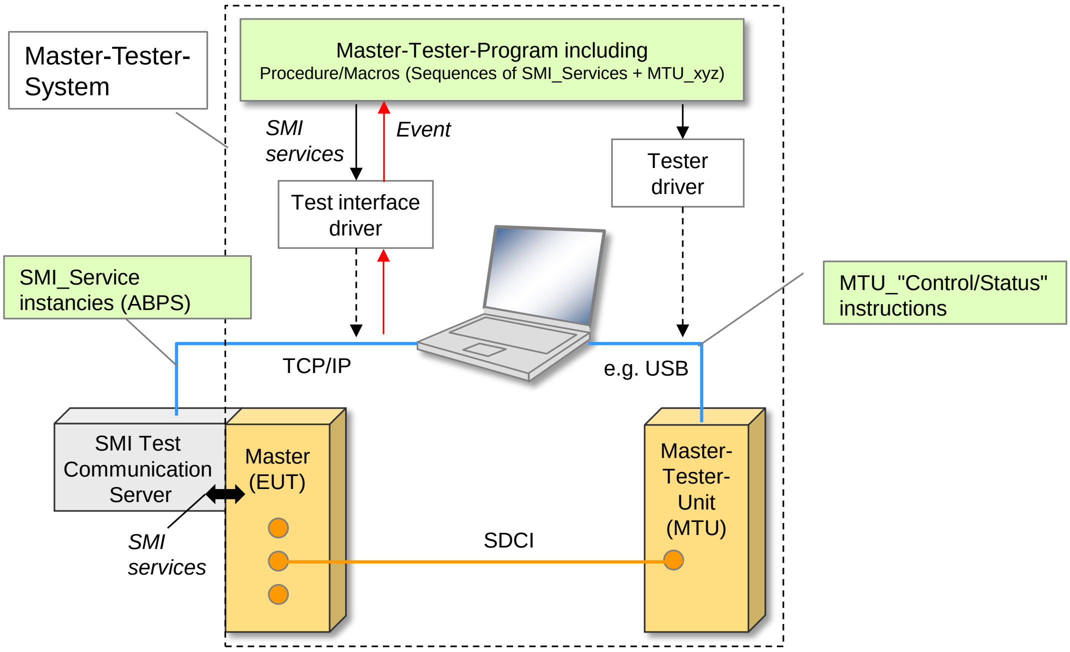

Master-Tester-System
Master-Tester-Program including
Procedure/Macros (Sequences of SMI_Services + MTU_xyz)
SMI services
Event
Test interface driver
Tester driver
SMI_Service instancies (ABPS)
MTU_"Control/Status" instructions
TCP/IP
e.g. USB
SMI Test Communication Server
Master (EUT)
Master-Tester-Unit (MTU)
SMI services
SDCI

Figure A.20 – Principle of a Master-Tester-System using SMI

A.3.2 System requirements for Master-Tester

The tools of a Master-Test-System shall be type-approved by the organization mentioned in Annex D prior to any conformity testing for a manufacturer declaration.

Table A.4 lists the system requirements for the approval of a Master-Tester-System.

Table A.4 – System requirements for the Master-Tester

-CR0111- -CR112-

Requirement	Description
SR1	The test system shall execute and evaluate the test cases defined in this specification
SR2	It should be possible to define, execute and evaluate additional customer specific test cases.
SR3	The result of every test case and also additional information about test execution shall be reported to the user (test report, log file). The user shall be able to store and print this information.
SR4	The conformity test cases defined in this specification and the user defined test cases should be coded in a script file (XML or TCL).
SR5	The conformity test script file shall be provided by test system supplier.
SR6	The conformity test script file shall be secured against manipulation.
SR7	For the sake of an approval of the test system, it shall have a TCP/IPcommunication path between EUT (SDCI Master) and the personal computer.
SR10	The test system manual shall be provided as PDF document. The user shall be able to read this document via freely available Adobe Reader software.

Requirement	Description
SR11	The Master-Tester software can be used to download new firmware updates to the Master-Tester. The download process can not be interrupted by the user. The software can not verify the content of the downloaded file. The user is responsible to use a valid and correct Master-Tester firmware update file.
SR13	The timeout for the time between entering the PREOPERATE state and leaving this state shall be adjustable in the Master tester

Table A.5 lists the functional requirements for the approval of a Master-Tester-System.

Table A.5 – Functional requirements for the performance of test cases

Requirement	Description
FR1	Usecase 1: Simulation of an SDCI Device
FR2	Usecase 2: Error behavior (stack-Errors like checksum errors, invalid timing and application errors such as creation of ErrorCodes)
FR3	Usecase 3: Creation of status information (number of transmitted messages by the master, number erroneous messages)
FR4	Usecase 4: Stand-alone device for EMC tests
FR5	Hardware EMC Requirements: - IEC61000-4-4 (Burst) +/- 2 kV Crit. A; +/- 4 kV Crit. B - IEC61000-4-6 (RF) 13 V Crit. A
FR6	Hardware Requirements: - SDCI interface - Slew Rate > 200 ns - Signaling LED or display: error counter (with active reset) - Power LED - SDCI communication LED - USB interface V2.0 (API) - Power supply via SDCI (optional USB or battery or external supply) - Non-volatile storage of configuration (maximum 1024 octets) - Monitoring (optional as independent tool) - Trigger output (24 V/10 mA)
FR7	Configuration areas: - Device configuration (MinCycleTime, M-sequence Capability, RevisionID, ProcessDataIn, ProcessDataOut, VendorID, DeviceID, FunctionID, transmission rate) - IO data configuration (Input data adjustable, mirror output data onto input data, increment input data) - Event configuration (maximum 6 events, unique, cyclic, depending on output) - ISDU configuration (all Indices) - Stack configuration
FR8	SDCI functionality: - all transmission rates (4,8; 38,4; 230,4 kbit/s) - SIO-Mode - All valid M-sequence types (TYPE_0, TYPE_1_1, TYPE_1_2, TYPE_1_V, TYPE_2_1, TYPE_2_2, TYPE_2_3, TYPE_2_4, TYPE_2_5, TYPE_2_6, TYPE_2_V) - All specified IO configurations - All specified ISDU Indices - Direct Parameter page (Index 0 and 1)
FR9	Trigger incidences: - Begin of start-up sequence (Wake-up is detected) - New Process Data cycle started - New SDCI M-sequence detected - Start of a new ISDU request detected - New ISDU response is generated - An Event is generated - Errors (checksum, parity, frame, protocol)

A.3.3 SMI Test Communication Server (STCS)

In order to perform an automated type testing, it is recommended for a Master manufacturer to provide an SMI Test Communication Server allowing the Master-Tester-Program to send and receive SMI service messages that carry the corresponding ArgBlocks. Send and receive is performed using the Transmission Control Protocol (TCP/IP) on Ethernet communication to the TCP/IP client of the Master-Tester-Program (see Figure A.20).

TCP/IP eliminates possible transmission errors and ensures that the transferred packages remain in the correct order. The Master-Tester-Program sends and receives the SMI service messages as described in [6] in serialized form and with Big-Endian encoding for multiple octet elements. The SMI service messages are self-contained as the receiving side can extract all the information necessary for decoding (length, Port, ClientID, etc.) from the octet stream. Thus, there is no need to add any overhead.

The STCS consists of a socket listener and a mapping logic converting SMI service messages into a form that can be understood by the Master (EUT). To establish a TCP/IP connection between Master-Tester-Program and STCS, the following sequence of steps is performed:

a) STCS listens on a local port (preferred: 49850) to connection requests issued by the Master-Tester-Program,

b) The Master-Tester-Program requests a connection from the STCS, which it shall accept,

c) A port is created by the Master-Tester-Program and is connected to the corresponding STCS port,

d) A socket is created on both ends of the connection, and the details of the connection are encapsulated by the socket,

e) The Master-Tester-Program sends SMI request messages and receives SMI response messages or event messages,

f) The STCS continues to listen for further connection requests.

The port number 49850 shall be used as default port number. If this port is not available or already in use, the next available port shall be used. Means for port number reassignment shall be provided both on the STCS and the Master-Tester-Program client. The following rules apply to the described TCP/IP connection:

- One STCS shall only correspond with one Master (EUT) instance,
- The STCS shall not store any status information including event queues. The Master-Tester-Program shall be responsible for queuing and processing events,
- The selected TCP/IP port number shall be within the dynamic, private port range (49152 to 65535). This range is used for private or customized services, for temporary purposes, and for automatic allocation of ephemeral ports.

The following hints should be considered for the implementation of the STCS and the Master-Tester-Program client:

- It is not guaranteed over TCP/IP that the sent SMI service messages are transmitted within one TCP/IP package. Therefore, means for splitting and merging SMI service messages shall be part of the implementation of the STCS and the Master-Tester-Program.
- The communication interface is full duplex.
- This concept allows for parallel testing of several Master Ports using different STCS connection ports (> 49850).

A.4 Components of Master test cases

A.4.1 Overview

SMI service instances and MTU-Instructions can be directly used to define test cases. In certain cases, it is advantageous to use macros combining basic SMI service instances and/or MTU-Instructions. Flow control expressions such as "wait until", "repeat from", etc. can supplement these macros. Macros shall be named intuitively and shall be preceded by the prefix "TM_" for test macro. Figure A.21 shows the components of Master test cases.

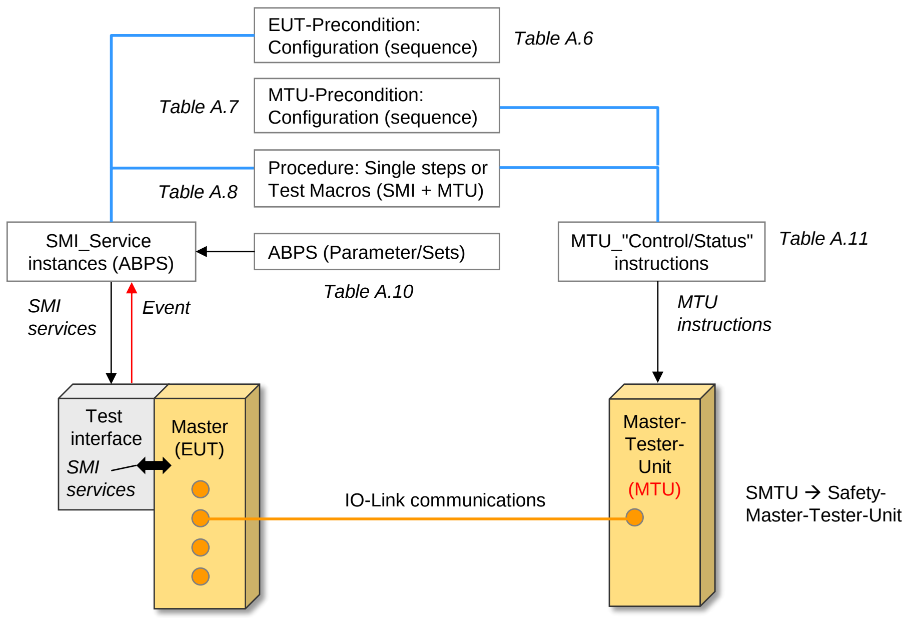

EUT-Precondition: Configuration (sequence)
Table A.6
Table A.7
MTU-Precondition: Configuration (sequence)
Procedure: Single steps or Test Macros (SMI + MTU)
Table A.8
SMI_Service instances (ABPS)
ABPS (Parameter/Sets)
Table A.10
MTU_"Control/Status" instructions
Table A.11
SMI services
Event
MTU instructions
Test interface
SMI services
Master (EUT)
IO-Link communications
Master-Tester-Unit (MTU)
SMTU → Safety-Master-Tester-Unit

Figure A.21 – Components of Master test cases

A.4.2 EUT preconditions (Master Port)

Table A.6 shows preconditions of the EUT. They can represent a description of a state or a sequence of activities to reach a certain state of the EUT.

Table A.6 – Preconditions of the EUT

-CR090-

Identifier	Description of state or activities to reach state
PORT_DI	Port is in DI Mode *;SMI_PortConfiguration*
PORT_DO	Port is in DO Mode
PORT_INACTIVE	Port is in Inactive Mode
PORT_AUTOSTART	Port is in Autostart Mode
NOTE EUT that supports PortPowerOffOn (SMI_Masteridentification; Features_1, Bit 2 = 1), the port power shall be switched on (SMI_PortPowerOffOn; PortPowermode = 2) before the SMI_Portconfiguration command is sent.	

A.4.3 MTU preconditions

Table A.7 shows macros of preconditions of the MTU playing the role of a controllable and observable Device. They describe values (instances) of parameters of a state or a sequence of activities to reach a certain state of the MTU.

Table A.7 – Preconditions of the MTU

-CR077-

Identifier	Description of state or activities to reach state
MTU_STANDARD_STATE	Transmission rate = 38,4 kbit/s *;COM2* DS_UPLOAD_FLAG = "0" *;no DS_UPLOAD_REQ* DPP1(MinCycleTime) = 0x28 *;4ms* DPP1(M-sequenceCapability) = 0x11 *;PREOPERATE = TYPE1_2,* *;OPERATE = TYPE_2_5* *;ISDU supported* DPP1(RevisionID) = 0x11 (NOTE) *;Revision = V1.1.x* DPP1(ProcessDataIn) = 0x08 *;PDIn = 8 bit* DPP1(ProcessDataOut) = 0x08 *;PDOut = 8 bit* DPP1(VendorID) = 0xFDE8 *;ID of IO-Link Community (Tester)*

Identifier	Description of state or activities to reach state
	DPP1(DeviceID) = 002BD2 (NOTE) *;DID = 11218* DPP1(FunctionID) = 0x0000 *;FID is reserved* Device is activated *;Device responds to Master* Event flag = 0 *;no Events pending* PD status flag = 1 *;Process Data valid* Mandatory Indices: Index 0x0010 (VendorName) = "IO-Link Community" *;UTF8 coding* Index 0x0012 (ProductName) = "MTU" *;UTF8 coding* Index 0x0003 (DataStorageIndex) = PARSET1 *;see Table A.13* All states and counters reset
NOTE Only these IDs can be overwritten by Master for compatibility tests (see 8.5)	
Permitted values for RevisionID are: 0x11 and 0x10.	

A.4.4 TestMacros (TM) of the Master-Tester-Program

Table A.8 shows TestMacros of the Master-Tester-Program for both EUT (Master) and MTU. All TestMacros shall return after ≤ 30 s (default Test_Timeout).

Table A.8 – TestMacros of the Master-Tester-Program

-CR055--CR053-

TestMacro identifier	Variable	Test Service Action to enter mode	Comment
TM_AWAIT	Time	Pause "Time" (e.g. 2000) before next step	Milliseconds
TM_AWAIT_PORT_STATUS	NO_DEVICE, PORT_DIAG, OPERATE, PREOPERATE	a) Repeat SMI_PortStatus service until PortStatusList.PortStatusInfo = NO_DEVICE, or PORT_DIAG, or or OPERATE	Monitored by Test_Timeout
TM_AWAIT_DI_HIGH	–	Repeat SMI_PDIn until PDIn.PDI0 = 1 and PDIn.PQI = 0x00	Monitored by Test _Timeout
TM_AWAIT_PD_VALIDITY	VALID/ INVALID	Repeat SMI_PDIn until PDIn.PQI.PQ = "1" (VALID) or "0" (INVALID)	–
TM_MASTER_UPLOAD	PARSET1, PARSET2 (see Table A.13)	a) MTU_DS_SetParameter(<variable>) b) SMI_PortConfiguration(ABPS_TYPE_COMP, <Validation&Backup = 3>) c) TM_AWAIT_PORT_STATUS(OPERATE) d) SMI_DSToParServ e) MTU_DS_CheckUpload *;delete monitoring states*	Provides EUT and MTU with DS data object

A.4.5 SMI Event Handling

Since the SMI services SMI_DeviceEvent and SMI_PortEvent are initiated by the EUT, the "Test interface driver" implementation shall provide a synchronous API to the Master-Tester-Program (see Figure A.18). Events sent by the EUT shall be enqueued into a separate internal queue data structure (one queue for Device Events, one queue for Port Events). The Master-Tester-Program can access these queues by means of the functions in Table A.9.

Table A.9 – Event functions for Event tester handling

Event function name	Parameter	Return value	Description
DLL_ClearAllEvents	–	–	Clear both queues
DLL_GetPortEvents	minEP	List of PortEvents (Qualifier, Code)	Buffer up to minEP Port Events in queue or until timeout before returning the list. Ignores optional 0xFF26, 0xFF27, or vendor-specific Events.
DLL_GetDeviceEvents	minED	List of DeviceEvents (Qualifier, Code)	Buffer up to minED Device Events in queue or until timeout before returning the list

A.4.6 SMI ArgBlock parameter sets (ABPS)

Table A.10 contains a list of ArgBlock parameter sets (ABPS). They contain predefined values for ArgBlock elements. The data types of the elements are specified in [6]. ABPS shall use the prefix "ABPS_".

ABPS names can be used in the description of SMI service instances. Predefined elements of an ABPS can be overwritten via the following syntax:

ABPS_xxxx<element1 = value, element2 = value, …>

For example, ABPS_PORTTODI<VendorID = 286> uses all element definitions of the ABPS_-PORTTODI but replaces the default value "0" the element "VendorID" by the value "286".

Table A.10 – ArgBlock Parameter Sets (ABPS)

-CR108- -CR077- CR054- -CR057- -CR092- -CR045-

ABPS	ArgBlock	Element	Type	Value
ABPS_VOIDBLOCK	VoidBlock	ArgBlockID		0xFFF0
ABPS_PORTTODI	PortConfigList	ArgBlockID	Unsigned16	0x8000
		PortMode	PortMode	PM_DI_CQ
		Validation&Backup	DSType	No_Device_Check
		I/Q behavior	IQ_Behavior	IQ_not_supported
		PortCycleTime	Unsigned8	0
		VendorID	Unsigned8	0
		DeviceID	Unsigned8	0
ABPS_PORTTODO	PortConfigList	ArgBlockID	Unsigned16	0x8000
		PortMode	PortMode	PM_DO_CQ
		Validation&Backup	DSType	No_Device_Check
		I/Q behavior	IQ_Behavior	IQ_not_supported
		PortCycleTime	Unsigned8	0
		VendorID	Unsigned8	0
		DeviceID	Unsigned8	0
ABPS_PORTINACTIVE	PortConfigList	ArgBlockID	Unsigned16	0x8000
		PortMode	PortMode	PM_DEACTIVATED
		Validation&Backup	DSType	No_Device_Check
		I/Q behavior	IQ_Behavior	IQ_not_supported
		PortCycleTime	Unsigned8	0
		VendorID	Unsigned8	0
		DeviceID	Unsigned8	0
ABPS_PORTPOWERON	PortPowerOffOn	ArgBlockID	Unsigned16	0x7003
		PortPowerMode	PortPowerMode	PP_Port_Power_On
		PortPowerOffTime	Unsigned16	0
ABPS_PORTPOWEROFF	PortPowerOffOn	ArgBlockID	Unsigned16	0x7003
		PortPowerMode	PortPowerMode	PP_Port_Power_Off
		PortPowerOffTime	Unsigned16	0
ABPS_DO_HIGH	PDOut	ArgBlockID	Unsigned16	0x1002
		OE	PDOutQualityInfo	PQ_PDOUTVALID
		DO	Q_Value	Q_High
ABPS_DO_LOW	PDOut	ArgBlockID	Unsigned16	0x1002

ABPS	ArgBlock	Element	Type	Value
		OE	PDOutQualityInfo	PQ_PDOUTVALID
		DO	Q_Value	Q_Low
ABPS_TYPE_COMP	PortCycleTime	ArgBlockID	Unsigned16	0x8000
		PortMode	PortMode	PM_IOL_MANUAL
		Validation&Backup	DSType	TYPE_compatible_Device_V1.1
		I/Q behavior	IQ_Behavior	IQ_not_supported
		PortCycleTime	Unsigned8	0x28
		VendorID	Unsigned16	0xFDE8
		DeviceID	Unsigned8	0x002BD2
ABPS_NOTYPE_CHECK	PortCycleTime	ArgBlockID	Unsigned16	0x8000
		PortMode	PortMode	PM_IOL_AUTOSTART
		Validation&Backup	DSType	No_Device_Check
		I/Q behavior	IQ_Behavior	IQ_not_supported
		PortCycleTime	Unsigned8	0x28
		VendorID	Unsigned16	0xFDE8
		DeviceID	Unsigned8	0x002BD2
ABPS_DEVICEWRITE	On-request_Data	ArgBlockID	Unsigned16	0x3000
		Index	Unsigned16	0
		SubIndex	Unsigned8	0
		On-request-Data	Octet string	0
ABPS_DEVICEREAD	On-request_Data	ArgBlockID	Unsigned16	0x3001
		Index	Unsigned16	0
		SubIndex	Unsigned8	0
ABPS_PDOUT	PDOut	ArgBlockID	Unsigned16	0x1002
		OE	Unsigned8	1
		OutputDataLength	Unsigned8	1
		PDO0	Unsigned8	0x00

A.4.7 MTU instructions

Table A.11 shows (fixed) instructions of the Master-Tester-Program for the MTU. Every MTU-instruction returns the specified parameters defined in "Return value".

Table A.11 – MTU instructions

Name	Parameter	Return value	Definition
MTU_Startup_Check10	–	TRUE/FALSE	Monitor whether Master connects to Device via standard start-up: a) Initiate wake-up b) Read DPP1 (Address 0x02 to 0x06) c) Write MasterCycleTime d) Write MasterCommand OPERATE e) Change to the appropriate M-sequence type (see TestCase) Reset monitoring states after invocation
MTU_Startup_GetLog	–	MessageLog	Returns a log of all Master messages at STARTUP

Name	Parameter	Return value	Definition
MTU_DPP1_Get	ParameterName	Value	Get parameter in full length from DPP1 (Direct Parameter Page 1)
MTU_DPP1_Set	ParameterName, Value	–	Set parameter in DPP1, e.g. MTU_DPP1_Set(VendorID = 0xFDE8) Values for PDIn and PDOut lengths presented in bits
MTU_DS_CheckDownload	–	TRUE/FALSE	Monitoring Device activity a) Read Index 3 b) Master sends DS_DownloadStart c) Master writes Parameter d) Master sends DS_DownloadEnd Returns TRUE if sequence above was performed at least once since start-up Reset monitoring states after invocation and start-up
MTU_DS_CheckDSCommands	–	TRUE/FALSE	Monitoring Device activity whether DS_Download commands or DS_Upload commands have been performed. Reset monitoring states after invocation and start-up
MTU_DS_CheckParameter	PARSET1, or PARSET2	TRUE/FALSE	Returns TRUE if requested PARSET is active
MTU_DS_CheckUpload	–	TRUE/FALSE	Checks Device activity a) Read Index 3 b) Master sends DS_UploadStart c) Master reads Parameter d) Master sends DS_UploadEnd Returns TRUE if sequence above was performed at least once since start-up Reset monitoring states after invocation and start-up
MTU_DS_Locked	–	–	Parameter DSIndex.StateProperty = “Data Storage locked”
MTU_DS_SetMaxDataStorage	MAXDATA, or MAXINDEXLIST	–	Activates Index_List MAXINDEXLIST or MAXDATA as shown in Table A.11
MTU_DS_SetParameter	PARSET1, or PARSET2	–	MTU activates PARSET1 or PARSET2 (see Table A.13)
MTU_DS_SetUpload	Flag_off, or Flag_on, or Event	TRUE/FALSE	Set upload behavior on MTU side: Sets/resets DS_UPLOAD_FLAG or raises a DS_UPLOAD_REQ Event and sets DS_UPLOAD_FLAG.
MTU_DS_SetError	IndexList_err, or R_Index19_err, or W_Index19_err	–	MTU to generate ErrorType = 0x8023 "Access denied". Either via - IndexList_err: Read on Index 3, Subindex 05 - R_Index19_err: Read on Index 19 - W_Index19_err: Write on Index 19
MTU_DS_Wait	DS_BREAK, or DS_DOWNLOAD END, or DS_UPLOAD END	–	MTU delays processing until it detects the chosen parameter. Monitored by Test_Timeout.
MTU_Event_CheckNoDetail	–	TRUE/FALSE	Check if Master reads Event without details and acknowledges within Test_Timeout a) Read Event StatusCode (address "0") in next cycle b) Write Event StatusCode (address "0") within Test_Timeout c) Master does not read other Event memory addresses
MTU_Event_Clear	–	–	Clear all values in Event memory
MTU_Event_SetStatusCode	code	–	Set StatusCode of the Event memory

Name	Parameter	Return value	Definition
			(address "0") to "code"
MTU_Event_SetSlot	Slot number, Qualifier, Code	–	Set Event Qualifier and Code of slot number in the Event memory
MTU_Event_Trigger-AndWaitForAck	–	StatusCode-Read, AllSlotsRead, Acknowledge	a) Set Event flag in Msequence.CKS b) Wait until Master Write to StatusCode (address "0" in Event memory, acknowledge) or until Test-Timeout. c) Return StatusCodeRead = TRUE if Master read the StatusCode d) Return AllSlotsRead = TRUE if Master read all Qualifier and Code values of the activated Events (indicated by StatusCode, always TRUE for StatusCode type 1) e) Return Acknowledge = TRUE if Master wrote to StatusCode and = FALSE upon Test-Timeout
MTU_Event_ISDUInterrupted	–	TRUE/FALSE	Return TRUE if the master switched to communication channel Diagnosis while last ISDU transfer was running
MTU_Event_SetFlag	–	–	In PREOPERATE, OPERATE: Set Event flag in MSequence.CKS immediately. Else: Set Event flag when PREOPERATE is reached.
MTU_ISDU_Add	Index, Subindex, Value (Octet-String), ErrorType_W, ErrorType_R, Specialty	–	Adds virtual ISDU to parameter-space. Length of OctetString reflects size ([0x01, 0x02…]). Master Read or Write instruction returns ErrorType_R/W as defined in Table C.1. [6] respectively, for example 0x8033. Specialties see Table A.14.
MTU_ISDU_Read	Index, Subindex,	Value (Octet-String)	Returns ISDU Parameter content as OctetString value
MTU_ISDU_TrailingIdles	–	NumIdles	Return minimum number of "IDLE 1" requests between ISDU services. Reset monitoring states after invocation.
MTU_ISDU_Write	Index, Subindex, Value (Octet-String)	–	Writes Value defined as OctetString to virtual or real ISDU.
MTU_PD_Get	–	OctetString	Return PDout values
MTU_PD_Set	OctetString	–	Set PDin values
MTU_State_Activate	–	–	Enable response to Master request
MTU_State_Deactivate	–	–	Disable response to Master request
MTU_State_BlockFallback	*n*	–	MTU does not respond to the next *n* Fallback commands
MTU_State_CheckFallback	–	TRUE/FALSE	Observe whether MasterCommand "Fallback" has been sent
MTU_State_CheckOperate	–	TRUE/FALSE	TRUE if MTU is in OPERATE
MTU_State_CheckPreoperate	–	NONE/ REACHED/ PASSED	MTU returns information on occurrence of state PREOPERATE. Reset when the communication is restarted (STARTUP state is entered)
MTU_State_GetMaster-RetryCTviolCount	–	RETRIES, CTVIOLS	Provides Retries and CycleTime violations. Values shall be cleared after performance.
MTU_State_CountRestarts	–	Restarts	MTU counts number of MTU restarts. Reset after invocation.
MTU_State_IncorrectChecksums	*n*	–	*n* reply messages to Master with incorrect checksums
MTU_State_MirrorPD	–	–	Applies only for PDInLength = PDOutLength. Mirrors PDOut to PDIn

Name	Parameter	Return value	Definition
			within the same M-sequence.
MTU_State_SkipResponse	*n* *WURQ*	–	Skip *n* reply messages to Master (at the expected COM speed). If WURQ = TRUE responses are skipped at wake up sequence, default WURQ = FALSE
MTU_State_SetPDValidity	VALID, or INVALID	–	Set PD validity to a given value. The MTU decides if it will propagate the information via flag (V1.1) or via Event (V1.0) whenever validity changed
MTU_Timing_Startup	–	TDWU12 TDWU23 WURQ WURQ101-500 TSD TDMT(COM1) TDMT(COM2) TDMT(COM3)	Returns timing measurement values in ms during start-up
MTU_Timing_GetTinicycInStartup	–	Min	Measure cycle times in STARTUP (minimum recovery time)
MTU_Timing_GetTinitcyc-InPreoperate	–	Min	Measure cycle times in PREOPERATE
MTU_Timing_GetTcyc-InOperate	–	Min	Measure cycle times in OPERATE during 20 Master cycles
MTU_Timing_SetReplyMessage Delay	TA, or T2	–	in TBIT; default values: TA = 5 TBIT, T2 = 0 TBIT

MTU_Timing_SetCommunication Mode	COM1, COM2, COM3	–	Transmission rates:
			COM1 = 4,8 kbit/s
			COM2 = 38,4 kbit/s
			COM3 = 230,4 kbit/s

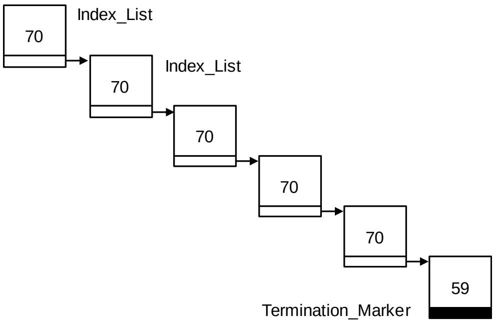
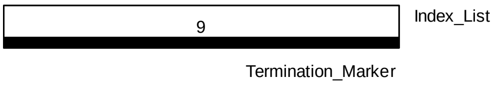
MTU_Timing_GetT1	–	Max	Measure the delays between the end of the stop bit and the beginning of the start bit of the next octet

A.4.8 MTU data sets

A.4.8.1 MTU Index lists for Data Storage tests

Table A.12 shows the two data sets used for Data Storage testing of Masters (see Table B.10 and G.1 in [6]).

Table A.12 – MTU Index lists for Data Storage tests

Item	MAXINDEXLIST (Concatenated)	MAXDATA per object
DataStorage Index	Index 3, Subindex 03 (Data_Storage_Size) = maximum size (2048 octets)	Index 3, Subindex 03 (Data_Storage_Size) = size (2048 octets)
Index_List	Six concatenated Index_Lists. Five of them with 70 entries, the sixth with 59 entries plus Termination_Marker  Index_List 70 Index_List 70 70 70 70 59 Termination_Marker	Single Index_List contains 9 entries (X1 to X9) plus Termination_Marker. 

Item	MAXINDEXLIST (Concatenated)	MAXDATA per object
Entries	408 Objects (2 octets Index, 1 octet Subindex, 1 octet length) with 1 octet data length > 2040 octets.	8 objects with 232 octets data length plus 1 object with 156 octets data length. This leads to a total size of 8 x (4+232) + 1 x (4+156) = 2048 octets for the structure defined in Annex G.1 in [6].

A.4.8.2 MTU parameter sets

Table A.13 shows the parameter sets used for MTU instructions "MTU_DS_SetParameter" and "MTU_DS_CheckParameter" (see Table A.11).

Table A.13 – MTU parameter sets

-CR098-

Name	Content 1	Content 2
PARSET1	Index: 64 Subindex 1: [0x12, 0x34] Subindex 2: [0x35, 0x69]	Index: 256 Subindex1: [0x79, 0x85] Subindex2: [0x92, 0x23]
PARSET2	Index: 64 Subindex 1: [0x47,0x11] Subindex 2: [0x98,0x76]	Index: 256 Subindex 1: [0x97,0x85] Subindex 2: [0x40,0x40]

A.4.8.3 MTU specialties

-CR052-

Table A.14 shows specialties used for MTU instructions "MTU_ISDU_Add" (see Table A.11).

Table A.14 – MTU specialties

Name	Definition
NO_DEVICE_BUSY	MTU generates an immediate response to the ISDU request without indicating "Device busy" (no I-Service/Length = 0x01 responses, see Table A.14 in [6]).
DEVICE_BUSY	MTU responds to the ISDU request indicating "Device busy" (at least one I-Service/Length = 0x01 response, see Table A.14 in [6])
ISSUE_EVENT	Event (DL, Error, Event single shot, EventCode = "0x5200")
TIMEOUT	Device does not respond (protocol error)
INCORRECT_SERVICE_CODE	Device responds with incorrect service code (I-Service/Length = 0x02, see Table A.14 in [6]) (sequence) (protocol error)
INCORRECT_CHKPDU	Device creates incorrect CRC signature within response (protocol error)
RESERVED_DATA_LENGTH	Device uses reserved combinations of iService and length (I-Service/Length = 0x10, see Table A.14 in [6])
EVENT	Set the Event flag bit in M-sequence.CKS on the next M-sequence after flowCTRL = 2 for the ISDU request/response on ISDU Write/Read access
NO_SERVICE	MTU responds directly (no busy responses) with “No service” (I-Service/Length = 0x00 response, see Table A.14 in [6])

Annex B
(normative)
Supplement to the legacy specification V1.0

B.1 General

The definitions in [6] are more comprehensive than the definitions in the predecessor [5]. In order to establish a reliable interoperation of legacy Master and Devices with their SDCI counterparts it is necessary to supplement the predecessor specification [5] by a few clarifications.

B.2 Legacy-Master power-on driver capability

If the actual power-on driver capability does not meet the requirements defined in [6], the measured value(s) of TC_0002 (5.2.2) shall be documented in the user manual of the Legacy-Master.

B.3 Legacy-Device power-on current consumption

If the actual power-on current consumption does not meet the requirements defined in [6], the measured value(s) of TC_0012 (5.3.2) shall be documented in the user manual of the Legacy-Device.

B.4 ISDU request and response abort

The "abort" feature is not specifically defined in [5]. All Legacy-Devices shall have implemented this behavior, which is tested in TC_0067 (6.5.17) and TC_0068 (6.5.18).

B.5 "Device 1.1" connected to a "Master 1.0"

A manufacturer or vendor of a Device without backward compatibility (V1.0 not supported) shall document in product sheet or user manual that the Device supports IO-Link V1.1 only (6.8.2.2).

B.6 Maximum MasterCycleTime

The maximum MasterCycleTime for both Master and Legacy-Master is 134 ms. This limit is checked in TC_0089 (6.9.1).

B.7 Maximum MinCycleTime

The maximum MinCycleTime for both Device and Legacy-Device is 134 ms. This limit is checked in TC_0090 (6.9.2).

B.8 Write access to reserved system commands

The following System commands shall not be tested in Legacy-Devices: 0x5A; 0x8D to 0x8F; 0x97 to 0x99.

A Write access to reserved system commands within a Legacy-Device returns a negative response: PAR_VALOUTOFRNG (0x8030). TC_0104 (6.10.2) is affected.

B.9 Time-out for Write access to system commands

Legacy-Devices shall respond within 5 s.

B.10 Text string length for Application Specific Tag

Existing Legacy-Devices are permitted to have text string length <16 octets. In this case, the manufacturer or vendor shall document the text string length in the user manual. It is highly recommended to provide a minimum of 16 octets. TC_0122 (6.10.19) and TC_0123 (6.10.20) are affected.

B.11 Write access with invalid length

A Write access to reserved system commands within a Legacy-Device returns a negative response: PAR_VALOUTOFRNG (0x8030). TC_0141 (6.10.32) and TC_0142 (6.10.33) are affected.

B.12 IODD "reset to factory settings" verification

It is highly recommended for Legacy-Devices to show the behavior defined in [6]. Deviations shall be documented in the user manual. TC_0155 (0) is affected.

B.13 Fallback in PREOPERATE

If the Master does not support the Fallback through a command from the upper-level system such as a fieldbus, the manufacturer or vendor of the Device or Legacy-Device respectively shall document the restriction or behavior in the user manual. TC_0213 (8.8.1) and TC_0214 (8.8.2) are affected.

Annex C
(normative)
Listing of test cases

C.1 Listing of test cases sorted by IDs

Table C.1 shows the Test cases and its references.

Table C.1 – Test cases sorted by IDs

SDCI TC ID	TC Name	Reference
TC_0001	TCM_PHYL_INTF_ISM	Table 7
TC_0002	TCM_PHYL_INTF_ISIRM	Table 8
TC_0003	TCM_PHYL_INTF_ILLM	Table 9
TC_0004	TCM_PHYL_INTF_VRESHIGH	Table 10
TC_0005	TCM_PHYL_INTF_VRESLOW	Table 11
TC_0006	TCM_PHYL_INTF_VTHHM	Table 12
TC_0007	TCM_PHYL_INTF_VTHLM	Table 13
TC_0008	TCM_PHYL_INTF_VHYSM	Table 14
TC_0011	TCD_PHYL_INTF_ISD	Table 17
TC_0012	TCD_PHYL_INTF_ISIRD	Table 18
TC_0013	TCD_PHYL_INTF_VRESHIGH	Table 20
TC_0014	TCD_PHYL_INTF_VRESLOW	Table 21
TC_0015	TCD_PHYL_INTF_IQQD	Table 22
TC_0016	TCD_PHYL_INTF_VTHHD	Table 23
TC_0017	TCD_PHYL_INTF_VTHLD	Table 24
TC_0018	TCD_PHYL_INTF_VHYSD	Table 25
TC_0021	TCM_PHYL_INTF_IQWUH	Table 28
TC_0022	TCM_PHYL_INTF_TWUH	Table 29
TC_0023	TCM_PHYL_INTF_IQWUL	Table 30
TC_0024	TCM_PHYL_INTF_TWUL	Table 31
TC_0025	TCD_PHYL_INTF_TWUH	Table 32
TC_0026	TCD_PHYL_INTF_TWUL	Table 33
TC_0027	TCD_PHYL_INTF_TRENHIGH	Table 34
TC_0028	TCD_PHYL_INTF_TRENLOW	Table 35
TC_0029	TCD_PHYL_INTF_TRDL	Table 36
TC_0030	TCM_PHYL_INTF_BITEYEMAXLOAD	Table 39
TC_0031	TCM_PHYL_INTF_BITEYEMINLOAD	Table 41
TC_0032	TCM_PHYL_INTF_UARTEYEMAXLOAD	Table 43
TC_0033	TCM_PHYL_INTF_UARTEYEMINLOAD	Table 45
TC_0034	TCD_DLPC_STUP_CYCTIME	Table 52
TC_0035	TCD_DLPC_STUP_STUPOPER1	Table 53
TC_0036	TCD_DLPC_STUP_STUPOPER2	Table 55
TC_0037	TCD_DLPC_OPER_OPERSTUP1	Table 56
TC_0038	TCD_DLPC_OPER_OPERSTAR2	Table 57
TC_0039	TCD_DLPC_PROP_READDPP1	Table 58
TC_0040	TCD_DLPC_PROP_WRITEDPP1	Table 59

TC_0041	TCD_DLPC_PROP_SHORTMESSAGE	Table 60
TC_0043	TCD_DLPC_PROP_SIMRESET	Table 61
TC_0044	TCD_DLPC_PROP_MSEQFAULT	Table 62
TC_0045	TCD_DLPC_OPER_READ	Table 64
TC_0046	TCD_DLPC_OPER_WRITE	Table 65
TC_0047	TCD_DLPC_OPER_NEGWRITE	Table 66
TC_0049	TCD_DLPC_OPER_SIMRESET	Table 67
TC_0052	TCD_DLPC_ISDU_AVAILMSEQCAP	Table 71
TC_0053	TCD_DLIC_ISDU_IDLEBUSYCHECK	Table 72
TC_0054	TCD_DLIC_ISDU_READINDEX8	Table 73
TC_0055	TCD_DLIC_ISDU_READ8EXTLENGTH	Table 74
TC_0056	TCD_DLIC_ISDU_WRITE8	Table 75
TC_0057	TCD_DLIC_ISDU_READ8RESERVED	Table 76
TC_0058	TCD_DLIC_ISDU_READ8NOSUBINDEX	Table 77
TC_0059	TCD_DLIC_ISDU_READ16	Table 78
TC_0060	TCD_DLIC_ISDU_WRITE16	Table 79
TC_0061	TCD_DLIC_ISDU_READ16RESERVED	Table 80
TC_0062	TCD_DLIC_ISDU_READ16NOSUBINDEX	Table 81
TC_0063	TCD_DLIC_ISDU_WRITE8LENOVERRUN	Table 82
TC_0064	TCD_DLIC_ISDU_WRITE8WRONGLEN	Table 83
TC_0065	TCD_DLIC_ISDU_WRITE8WRONGCHECKSUM	Table 84
TC_0066	TCD_DLIC_ISDU_WRITE8ROINDEX	Table 85
TC_0067	TCD_DLIC_ISDU_ABORTREADREQ	Table 86
TC_0068	TCD_DLIC_ISDU_ABORTREADRESP	Table 87
TC_0069	TCD_DLIC_EVNT_OPERSINGLEEVENT	Table 89
TC_0070	TCD_DLIC_EVNT_PROPSINGLEEVENT	Table 90
TC_0071	TCD_DLIC_EVNT_OPEREVENTCLEAR	Table 91
TC_0072	TCD_DLIC_EVNT_OPERCOMMINTERRUPT	Table 92
TC_0073	TCD_DLIC_EVNT_OPERPOWERINTERRUPT	Table 93
TC_0074	TCD_DLIC_EVNT_OPERAPPEARDISAPPEAR	Table 94
TC_0075	TCD_DLIC_EVNT_OPERMULTEVENT	Table 95
TC_0076	TCD_DLIC_EVNT_OPERSHORTEVENT	Table 96
TC_0077	TCD_APPS_DSUP_NOFLAG	Table 98
TC_0078	TCD_APPS_DSUP_VIADOWNLOADSTORE	Table 99
TC_0079	TCD_APPS_DSUP_VIADOWNLOADSTORENOWRITE	Table 100
TC_0080	TCD_APPS_DSUP_VIALOCALCHANGE	Table 101
TC_0081	TCD_APPS_DSUP_PARABREAKABORT	Table 102
TC_0082	TCD_APPS_DSDN_PARAMODIFICATION	Table 104
TC_0083	TCD_APPS_DSDN_FACTORYRESET	Table 105
TC_0084	TCD_APPS_DSDN_PARABREAKABORT	Table 106
TC_0085	TCD_DLIC_COMP_STARTUP	Table 110
TC_0086	TCD_DLIC_COMP_TYPE1INTERLEAVE	Table 111
TC_0087	TCD_DLIC_COMP_PDINVALIDEVENT	Table 112
TC_0089	TCD_DLPC_STDP_MASTERCYCLETIME	Table 113
TC_0090	TCD_DLPC_STDP_MINCYCLETIME	Table 114

TC_0091	TCD_DLPC_STDP_MSEQCAPABILITY	Table 115
TC_0092	TCD_DLPC_STDP_REVISIONID	Table 116
TC_0093	TCD_DLPC_STDP_PDIN	Table 117
TC_0094	TCD_DLPC_STDP_PDOUT	Table 118
TC_0095	TCD_DLPC_STDP_VENDORID	Table 119
TC_0096	TCD_DLPC_STDP_DEVICEID	Table 120
TC_0097	TCD_DLPC_STDP_FUNCTIONID	Table 121
TC_0101	TCD_DLPC_STDP_WRITERESPAR	Table 122
TC_0104	TCD_DLIC_DEFP_SYSCMDRES	Table 124
TC_0107	TCD_DLIC_DEFP_DSINDEX	Table 125
TC_0108	TCD_DLIC_DEFP_DSRECORD	Table 126
TC_0109	TCD_DLIC_DEFP_ACCESSLOCKSVAL	Table 128
TC_0110	TCD_DLIC_DEFP_ACCESSLOCKSINVAL	Table 129
TC_0111	TCD_DLIC_DEFP_PROFILCHARAC	Table 130
TC_0112	TCD_DLIC_DEFP_PDINDESC	Table 131
TC_0113	TCD_DLIC_DEFP_PDOUTDESC	Table 132
TC_0114	TCD_DLIC_DEFP_VENDORNAM	Table 133
TC_0115	TCD_DLIC_DEFP_VENDORTEXT	Table 134
TC_0116	TCD_DLIC_DEFP_PRODUCTNAM	Table 135
TC_0117	TCD_DLIC_DEFP_PRODUCTID	Table 136
TC_0118	TCD_DLIC_DEFP_PRODUCTTEXT	Table 137
TC_0119	TCD_DLIC_DEFP_SERNUM	Table 138
TC_0120	TCD_DLIC_DEFP_HARDREV	Table 139
TC_0121	TCD_DLIC_DEFP_FIRMREV	Table 140
TC_0122	TCD_DLIC_DEFP_TAGVALID	Table 141
TC_0123	TCD_DLIC_DEFP_TAGINVALID	Table 142
TC_0124	TCD_DLIC_DEFP_ERRCOUNT	Table 143
TC_0128	TCD_DLIC_DEFP_DEVSTAT	Table 144
TC_0129	TCD_DLIC_DEFP_DETAILDEVSTAT	Table 145 – Detailed Device Status – complete object
TC_0131	TCD_DLIC_DEFP_DETAILDEVSTATACTIVE	Table 146
TC_0132	TCD_DLIC_DEFP_PDIN	Table 147
TC_0133	TCD_DLIC_DEFP_PDOUT	Table 148
TC_0134	TCD_DLIC_DEFP_OFFTIMEVALID	Table 149
TC_0136	TCD_DLIC_DEFP_PROFILEPARREAD	Table 150
TC_0137	TCD_DLIC_DEFP_PROFILEPARWRITE	Table 151
TC_0140	TCD_DLIC_DEFP_WRITETOREADONLY	Table 152
TC_0141	TCD_DLIC_DEFP_WRITETOOSHORT	Table 153
TC_0142	TCD_DLIC_DEFP_WRITETOOLONG	Table 154
TC_0143	TCD_DSBP_APPL_BPDOWNLOAD	Table 155
TC_0144	TCD_DSBP_APPL_BPBREAKCMD	Table 156
TC_0145	TCD_DSBP_APPL_BPBREAKRESET	Table 157
TC_0146	TCD_DSBP_APPL_BPBREAKILLPARAM	Table 158
TC_0147	TCD_DSBP_APPL_BPBREAK2DOWNLOADS	Table 159

TC_0148	TCD_DSBP_APPL_BPBREAKLOCALLOCK	Table 160
TC_0149	TCD_IODD_PARV_IDENT	Table 168
TC_0150	TCD_IODD_PARV_COMPROFILE	Table 169
TC_0151	TCD_IODD_PARV_READVERIFY	Table 170
TC_0152	TCD_IODD_PARV_WRITEVERIFY	Table 171
TC_0155	TCD_IODD_PARV_FACTORYSETTINGS	Table 176
TC_0156	TCD_IODD_PARV_ACCESSLOCK	Table 173
TC_0157	TCD_IODD_PARV_INDEXCONSISTENT	Table 172
TC_0158	TCM_ PHYL_TIME_TDMT	Table 180
TC_0159	TCM_ PHYL_TIME_TDWU	Table 181
TC_0160	TCM_ PHYL_TIME_NUMOFWURQS	Table 182
TC_0161	TCM_ PHYL_TIME_TSD	Table 183
TC_0162	TCM_ PHYL_TIME_TINITCYC	Table 184
TC_0163	TCM_ PHYL_TIME_MASTERCYCLETIME	Table 188
TC_0164	TCM_ PHYL_TIME_MASTERCYCLETIMEREAL	Table 189
TC_0165	TCM_ PHYL_TIME_DEVRESPTIMES	Table 190
TC_0166	TCM_ PHYL_TIME_UARTT2	Table 191
TC_0167	TCM_ PHYL_TIME_UARTT1	Table 192
TC_0168	TCM_ DLPD_CYCC_TYPE21BIT8IN	Table 193
TC_0169	TCM_ DLPD_CYCC_TYPE22BIT16IN	Table 194
TC_0170	TCM_ DLPD_CYCC_TYPE23BIT8OUT	Table 195
TC_0171	TCM_ DLPD_CYCC_TYPE24BIT16OUT	Table 196
TC_0172	TCM_ DLPD_CYCC_TYPE25BIT8INBIT8OUT	Table 197
TC_0173	TCM_ DLPD_CYCC_TYPE1OCTET32IN	Table 199
TC_0176	TCM_ DLPD_CYCC_MIRROREDPD	Table 202
TC_0177	TCM_ DLPD_CYCC_PDINVALID	Table 203
TC_0178	TCM_ DLPD_CYCC_PDVALID	Table 204
TC_0179	TCM_DLOD_CYCC_TYPE2VPDXOD1	Table 205
TC_0180	TCM_DLOD_CYCC_TYPE2VPDXOD2	Table 206
TC_0181	TCM_DLOD_CYCC_TYPE2VPDXOD8	Table 207
TC_0182	TCM_DLOD_CYCC_TYPE2VPDXOD32	Table 208
TC_0183	TCM_DLST_CHCK_COMPARAM	Table 210
TC_0184	TCM_DLST_CHCK_VIDDID	Table 211
TC_0185	TCM_DLST_CHCK_V10VIDDID	Table 212
TC_0186	TCM_DLST_CHCK_NONCONFVIDDID	Table 213
TC_0187	TCM_DLST_CHCK_CONFVIDDID	Table 214
TC_0188	TCM_DLST_CHCK_OVERDIDOK	Table 215
TC_0189	TCM_DLST_CHCK_OVERDIDNOK	Table 216
TC_0190	TCM_DLST_CHCK_OVERRIDNOK	Table 217
TC_0192	TCM_DLST_CHCK_VIDDIDNONCONFIG	Table 218
TC_0193	TCM_DLST_CHCK_VIDDIDCONFIG	Table 219
TC_0194	TCM_DLST_CHCK_DIDWRONG	Table 220
TC_0202	TCM_DLOD_PREP_TYPE0READOD1	Table 222
TC_0203	TCM_DLOD_PREP_TYPE12READOD2	Table 223
TC_0204	TCM_DLOD_PREP_TYPE1VREADOD8	Table 224

TC_0205	TCM_DLOD_PREP_TYPE1VREADOD32	Table 225
TC_0206	TCM_DLOD_PREP_TYPE0WRITEOD1	Table 226
TC_0207	TCM_DLOD_PREP_TYPE12WRITEOD2	Table 227
TC_0208	TCM_DLOD_PREP_TYPE1VWRITEOD8	Table 228
TC_0209	TCM_DLOD_PREP_TYPE1VWRITEOD32	Table 229
TC_0210	TCM_DLOD_OPER_TYPE0READOD1	Table 232
TC_0211	TCM_DLOD_OPER_TYPE0WRITEOD1	Table 233
TC_0212	TCM_DLOD_OPER_TYPE12WRITEOD2	Table 234
TC_0213	TCM_DLFB_PROP_OK	Table 235
TC_0214	TCM_DLFB_PROP_FAILS	Table 236
TC_0215	TCM_DLFB_OPER_OK	Table 237
TC_0216	TCM_DLFB_OPER_FAILS	Table 238
TC_0222	TCM_DLCC_RTRY_MAXWURQNOSUCCESS	Table 251
TC_0223	TCM_ALIC_AERR_WRITEREJECT	Table 252
TC_0224	TCM_ALIC_AERR_WRITEINDEXUNSUPPORTED	Table 253
TC_0225	TCM_ALIC_AERR_WRITESUBINDEXNOTSUPPORTED	Table 254
TC_0226	TCM_ALIC_AERR_WRITETEMPUNAV	Table 255
TC_0227	TCM_ALIC_AERR_WRITEINDEXTEMPANAVLC	Table 256
TC_0228	TCM_ALIC_AERR_WRITEINDEXTEMPANAVDC	Table 257
TC_0229	TCM_ALIC_AERR_WRITEINDEXRO	Table 258
TC_0230	TCM_ALIC_AERR_WRITEINVALIDLEN	Table 259
TC_0231	TCM_ALIC_AERR_WRITEPARAMOUTOFRNG	Table 260
TC_0232	TCM_ALIC_AERR_WRITEPARAMABOVELIMIT	Table 261
TC_0233	TCM_ALIC_AERR_WRITEPARAMBELOWLIMIT	Table 262
TC_0234	TCM_ALIC_AERR_WRITEPARAMINVALID	Table 263
TC_0235	TCM_ALIC_AERR_WRITEDEVICEAPPFAULT	Table 264
TC_0237	TCM_ALIC_AERR_WRITERESERVEDINDEX	Table 265
TC_0238	TCM_ALIC_AERR_WRITERESERVEDINDEXNOISDU	Table 266
TC_0239	TCM_ALIC_DERR_WRITENOBUSY	Table 267
TC_0240	TCM_ALIC_DERR_WRITEAFTERBUSYTIMEOUT	Table 268
TC_0241	TCM_ALIC_DERR_ILLSERVICECODE	Table 269
TC_0242	TCM_ALIC_DERR_WRONGCHECKSUM	Table 270
TC_0243	TCM_ALIC_DERR_READNODATA	Table 272
TC_0244	TCM_ALIC_DERR_WRITERESERVEDDL	Table 271
TC_0245	TCM_ALIC_LIMT_WRITEMINDATALENGTH	Table 273
TC_0246	TCM_ALIC_LIMT_WRITEMAXDATALENGTH	Table 274
TC_0248	TCM_ALIC_LIMT_READMAXDATALENGTH	Table 275
TC_0249	TCM_ALIC_LIMT_WRITEINDEX8NOSUBINDEX	Table 276
TC_0250	TCM_ALIC_LIMT_WRITEINDEX8SUBINDEX8	Table 277
TC_0251	TCM_ALIC_LIMT_WRITEINDEX16SUBINDEX8	Table 278
TC_0252	TCM_ALIC_LIMT_IMMEDIATERESPNOBUSY	Table 279
TC_0253	TCM_ALIC_LIMT_IMMEDIATERESPWITHBUSY	Table 280
TC_0254	TCM_ALIC_LIMT_WRITEMAXSERVICELEN15	Table 281
TC_0255	TCM_ALIC_LIMT_WRITEMINSERVICEEXTLEN17	Table 282
TC_0256	TCM_ALIC_EVNT_NODETAILSNOTIFY	Table 283

TC_0261	TCM_ALIC_EVNT_WITHDETAILSSINGLEEVENT	Table 284
TC_0262	TCM_ALIC_EVNT_WITHDETAILSDOUBLEEVENT	Table 285
TC_0263	TCM_ALIC_EVNT_WITHDETAILSSIXEVENTS	Table 287
TC_0264	TCM_ALIC_EVNT_WRITEISDUWITH EVENT	Table 288
TC_0265	TCM_ALIC_EVNT_READISDUWITH EVENT	Table 289
TC_0266	TCM_ALIC_EVNT_WRITEISDUWITH EVENTDETAILS	Table 290
TC_0267	TCM_ALIC_EVNT_READISDUWITH EVENTDETAILS	Table 291
TC_0280	TCM_ALIC_STOR_DSLOCKED	Table 315
TC_0285	TCM_LGCY_MANY_DETECTANDCONNECT	Table 316
TC_0286	TCM_LGCY_MANY_DETECTANDINTERLEAVE	Table 317
TC_0287	TCM_LGCY_MANY_EVENTACK	Table 318
TC_0288	TCM_LGCY_MANY_IDLEAFTERISDU	Table 209
TC_0289	TCM_LGCY_MANY_EVENTINTERRUPTSISDU	Table 319
TC_0290	TCM_LGCY_MANY_PDINVALIDEVENT	Table 320
TC_0294	TCD_PHYL_INTF_BITEYEMAXLOAD	Table 40
TC_0295	TCD_PHYL_INTF_BITEYEMINLOAD	Table 42
TC_0296	TCD_PHYL_INTF_UARTEYEMAXLOAD	Table 44
TC_0297	TCD_PHYL_INTF_UARTEYEMINLOAD	Table 46
TC_0298	TCM_ DLPD_CYCC_TYPE26BIT16INBIT16OUT	Table 198
TC_0299	TCM_PHYL_INTF_VOLTRANGECQ	Table 15
TC_0300	TCD_PHYL_INTF_VOLTRANGECQ	Table 26
TC_0301	TCD_PHYL_INTF_TDELAYTOSIO	Table 37
TC_0302	TCD_PHYL_INTF_TTOFALLBACK	Table 38
TC_0303	TCM_PHYL_INTF_UARTTRANSDELAY	Table 47
TC_0304	TCD_PHYL_INTF_UARTTRANSDELAY	Table 48
TC_0305	TCD_PHYL_INTF_RESPONSETIME	Table 49
TC_0306	TCD_DLPC_CHCK_OVERRIDOK	Table 54
TC_0307	TCM_DLST_CHCK_OVERRIDOK	Table 221
TC_0308	TCD_DLIC_EVNTOPER_SINGLENOTIFICATION	Table 286
TC_0309	TCD_DLIC_ISDU_ELEMENT_RETRY	Table 88
TC_0310	TCM_PHYL_INTF_MRGI	Table 16
TC_0311	TCD_PHYL_INTF_DRGI	Table 27
TC_0312	TCD_DLPC_OPER_OUTINVALID	Table 68
TC_0313	TCD_DLPC_OPER_CONNECTIONLOSS	Table 69
TC_0314	TCD_DLPC_DID_OVERWRITE_COMP	Table 178
TC_0315	TCD_DLPC_DID_OVERRIDE_INCOMP	Table 179
TC_0316	TCD_IODD_PARV_DEVICERESET	Table 174
TC_0317	TCD_IODD_PARV_APPLRESET	Table 175
TC_0318	TCD_IODD_PARV_BACKTOBOX	Table 177
TC_0319	TCD_PHYL_INTF_TRANSMISSIONERRORS	Table 50
TC_0320	TCD_PHYL_INTF_DOCISD	Table 18
TC_0321	TCD_DSBP_APPL_DSSINGLEPARAM	Table 107
TC_0322	TCD_DS_APP_IDLEFLAGCLEAR	Table 108
TC_0323	TCD_DLIC_DEFP_DSRECORDMARKER	Table 127
TC_0324	TCD_DSBP_APPL_UNEXPECTEDINIDLE	Table 161

TC_0325	TCD_DSBP_APPL_WRITEINUPLOAD	Table 162
TC_0326	TCD_DSBP_APPL_READINDOWNLOAD	Table 163
TC_0327	TCD_DSBP_APPL_UNEXPINUPLOAD	Table 164
TC_0328	TCD_DSBP_APPL_SWITCHSTATES	Table 165
TC_0329	TCD_DSBP_APPL_UPBREAKRESET	Table 166
TC_0330	TCD_DSBP_APPL_DNENDBYUPLOAD	Table 167
TC_0331	TCM_ PHYL_TIME_TINITCYC_PREOP_2	Table 185
TC_0332	TCM_ PHYL_TIME_TINITCYC_PREOP_8	Table 186
TC_0333	TCM_ PHYL_TIME_TINITCYC_PREOP_32	Table 187
TC_0334	TCM_ DLPD_CYCC_ TYPE2VOD1IN0OUT32OCTET	Table 200
TC_0335	TCM_ DLPD_CYCC_ TYPE2VOD1IN32OUT0OCTET	Table 201
TC_0336	TCM_ DLPD_CYCC_TYPE1V_OD8BIT0INBIT0OUT	Table 230
TC_0337	TCM_ DLPD_CYCC_TYPE1V_OD32BIT0INBIT0OUT	Table 231
TC_0338	TCM_DLCC_RTRY_CKSINCORR2STRTUP	Table 239
TC_0339	TCM_DLCC_RTRY_ CKSINCORR2PREOP	Table 240
TC_0340	TCM_DLCC_RTRY_ CKSINCORR2OPERATE	Table 241
TC_0341	TCM_DLCC_RTRY_CKSINCORR3STRUP&RESTOP	Table 242
TC_0342	TCM_DLCC_RTRY_CKSINCORR3PREOP&RESTOP	Table 243
TC_0343	TCM_DLCC_RTRY_CKSINCORR3OPER&RESTOP	Table 244
TC_0344	TCM_DLCC_RTRY_MISSREP2STRTUP	Table 245
TC_0345	TCM_DLCC_RTRY_MISSREP2PREOP	Table 246
TC_0346	TCM_DLCC_RTRY_MISSREP2OPER	Table 247
TC_0347	TCM_DLCC_RTRY_REPMISS3STRTUP&RESTOP	Table 248
TC_0348	TCM_DLCC_RTRY_REPMISS3PREOP&RESTOP	Table 249
TC_0349	TCM_DLCC_RTRY_REPMISS3OPER&RESTOP	Table 250
TC_0350	TCM_ALIC_EVNT_NODEV	Table 292
TC_0351	TCM_ALIC_EVNT_PREOPERATE	Table 293
TC_0352	TCM_ALIC_EVNT_PORT_DIAG	Table 294
TC_0353	TCM_ALIC_EVNT_INVALID_CYCLE_TIME	Table 295
TC_0354	TCM_ALIC_STOR_DELETEDSAFTERRECONF	Table 296
TC_0355	TCM_ALIC_STOR_MAXSIZELIMITS	Table 297
TC_0356	TCM_ALIC_STOR_DSFROMPARAMSERV	Table 298
TC_0357	TCM_ALIC_STOR_INCONSDSFROMPARSERV	Table 299
TC_0358	TCM_ALIC_STOR_DSTOIMPROPPORTCONF	Table 300
TC_0359	TCM_ALIC_STOR_UPLOADREQEVENTOPERBACKREST	Table 301
TC_0360	TCM_ALIC_STOR_UPLOADREQEVENTOPERREST	Table 302
TC_0361	TCM_ALIC_STOR_UPLOADREQEVENTOPERBACKRESTINACT	Table 303
TC_0362	TCM_ALIC_STOR_STARTUPEMPTDSDEACTIV	Table 304
TC_0363	TCM_ALIC_STOR_STRTUPEMPTYDSDSACTIV	Table 305
TC_0364	TCM_ALIC_STOR_STRTUPDEVREPLACEDSDOWN	Table 306
TC_0365	TCM_ALIC_STOR_STRTUPDEVREPLACENODSDOWN	Table 307
TC_0366	TCM_ALIC_STOR_STRTUPDEVREPLACEDSUPFLAG	Table 308
TC_0367	TCM_ALIC_STOR_STRTUPDEVREPLACEDSFLAGNOUPLOAD	Table 309
TC_0368	TCM_ALIC_STOR_DSINDLISTCORRUPT	Table 310
TC_0369	TCM_ALIC_STOR_DSDOWNLOADFAULT	Table 311

TC_0370	TCM_ALIC_STOR_DSUPLOADFAULT	Table 312
TC_0371	TCM_ALIC_STOR_INCOMPDEVICE&DS	Table 313
TC_0372	TCM_ALIC_STOR_POWEROFFONNONVOLATILEDS	Table 314
TC_0373	TCD_DLIC_DEFP_EVENTDEVSTAT	Table 319
TC_0374	TCD_DS_APP_UPLOAD_FLAG_NON_VOLATILE	Table 321
TC_0375	TCD_DLPC_STDP_MINCYCLETIME_VALID	Table 322
TC_0376	TCD_DLPC_OPER_PDVALIDITYINDICATION	Table 318
TC_0377	TCD_DLPC_PROP_RECOVERY_TIME	Table 317
TC_0378	TCD_APPS_DSUP_INDEXLIST	Table 320

Annex D
(informative)
Information on conformity testing of SDCI

Information about testing Masters and Devices for conformity with [1] and [6] can be obtained from the following organization:

IO-Link Community
c/o PROFIBUS Nutzerorganisation
Ohio Straße 8
76149 Karlsruhe
Germany
Phone: +49 (0) 721 / 98 61 97 0
Fax: +49 (0) 721 / 98 61 97 11
e-mail: info@io-link.com
Web site: http://www.io-link.com

Usually, type testing of Master or Device is completed by a manufacturer declaration, which can be downloaded from the IO-Link website www.io-link.com.

Bibliography

-CR033-

[1] IEC 61131-9 Edition 2 (project in progress), Programmable controllers – Part 9: *Single-drop digital communication interface for small sensors and actuators (SDCI)*

[2] ISO/IEC 19505-2:2012 Information technology – OMG Unified Modeling Language (OMG UML), Revision 2

[3] IO-Link Community*, IO Device Description (IODD)*, V1.1.4, 2024, Order No. 10.012

[4] IO-Link Community, *IO-Link Common Profile,* V1.2, 2024, Order No. 10.072

[5] IO-Link Community, *IO-Link Communication,* V1.0, January 2009, Order No. 10.002

[6] IO-Link Community, *IO-Link Interface and System,* V1.1.3, June 2019, Order No. 10.002

[7] IO-Link Community, *IO-Link Interface and System,* V1.1.4, June 2024, Order No. 10.002

[8] IO-Link Community, *IO-Link Exceptions*, V1.0, January 2020, Order No. 10.232

____

-CR032-

IO-Link Community
c/o PROFIBUS Nutzerorganisation
Ohio Straße 8
76149 Karlsruhe
Germany
Phone: +49 (0) 721 / 98 61 97 0
Fax: +49 (0) 721 / 98 61 97 11
e-mail: info@io-link.com
http://www.io-link.com/