



BSI Standards Publication

Road vehicles — Media Oriented Systems Transport (MOST)

Part 3: Application layer conformance test plan

National foreword

This British Standard is the UK implementation of ISO 21806-3:2020.

The UK participation in its preparation was entrusted to Technical Committee AUE/16, Data Communication (Road Vehicles).

A list of organizations represented on this committee can be obtained on request to its committee manager.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2020
Published by BSI Standards Limited 2020

ISBN 978 0 539 03043 3

ICS 43.040.15

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 October 2020.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**Road vehicles — Media Oriented
Systems Transport (MOST) —**

**Part 3:
Application layer conformance test
plan**

*Véhicules routiers — Système de transport axé sur les médias —
Partie 3: Plan d'essais de conformité de la couche d'application*





COPYRIGHT PROTECTED DOCUMENT

© ISO 2020

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

	Page
Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols and abbreviated terms	2
4.1 Symbols	2
4.2 Abbreviated terms	2
5 Conventions	2
6 CTP overview	2
6.1 Test set-up	2
6.2 Conformance test plan organisation	4
7 CTP general information	4
7.1 CTC remarks	4
7.1.1 Timer naming	4
7.1.2 Deadlock prevention	4
7.1.3 Un-initialised logical node address	5
7.1.4 Addresses of MOST nodes in the LT	5
7.1.5 Device manufacturer information list	5
7.1.6 States of the node that contains the IUT	8
7.1.7 Procedures	10
7.1.8 Violation of prerequisites of the CTC	11
7.2 CTC items	11
7.2.1 FBlock EnhancedTestability	11
7.2.2 Multi-node devices	11
7.2.3 Node kinds excluded from conformance testing	11
8 CTC specification	11
8.1 Static FBlock behaviour	11
8.1.1 CTC_2.1.0-1 – Generic FBlock property test	11
8.1.2 CTC_2.1.0-2 – Generic FBlock method test	14
8.2 Power management	16
8.2.1 Power management – PowerMaster	16
8.2.2 Power management – PowerSlave	20
8.3 Error management	24
8.3.1 CTC_2.4.1-2 – Restart continue test	24
8.3.2 CTC_2.4.1-9 – Reaction on network change event test	27
8.4 Central registry	29
8.4.1 Central registry handling (NetworkMaster)	29
8.4.2 Central registry handling test (NetworkSlave)	47
8.4.3 CTC_2.6.4-10 – InstID wildcard test	54
8.5 CTC_2.7-1 – Node addressing test	56
8.6 Notification matrix test	58
8.6.1 CTC_2.8.3-1a – Notification matrix storage test (NetworkMaster)	58
8.6.2 CTC_2.8.3-1b – Notification matrix storage test (NetworkSlave)	61
8.6.3 CTC_2.8.3-2 – NotificationCheck test	62
8.6.4 CTC_2.8.3-7 – Notification matrix double entry test	64
8.6.5 CTC_2.8.3-10 – Notification error test	66
8.7 CTC_3.0-1 – TEST_GSI_GSO_Identification	69
8.8 Obligatory tests for sink and source MOST devices	71
8.8.1 General	71
8.8.2 Sink MOST devices	71

8.8.3 Source MOST devices 79

Annex A (normative) Measurement uncertainty for individual CTCs..... 94

Bibliography 96

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 22, *Road vehicles*, Subcommittee SC 31, *Data communication*.

A list of all parts in the ISO 21806 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The Media Oriented Systems Transport (MOST) communication technology was initially developed at the end of the 1990s in order to support complex audio applications in cars. The MOST Cooperation was founded in 1998 with the goal to develop and enable the technology for the automotive industry. Today, MOST¹⁾ enables the transport of high quality of service (QoS) audio and video together with packet data and real-time control to support modern automotive multimedia and similar applications. MOST is a function-oriented communication technology to network a variety of multimedia devices comprising one or more MOST nodes.

Figure 1 shows a MOST network example.

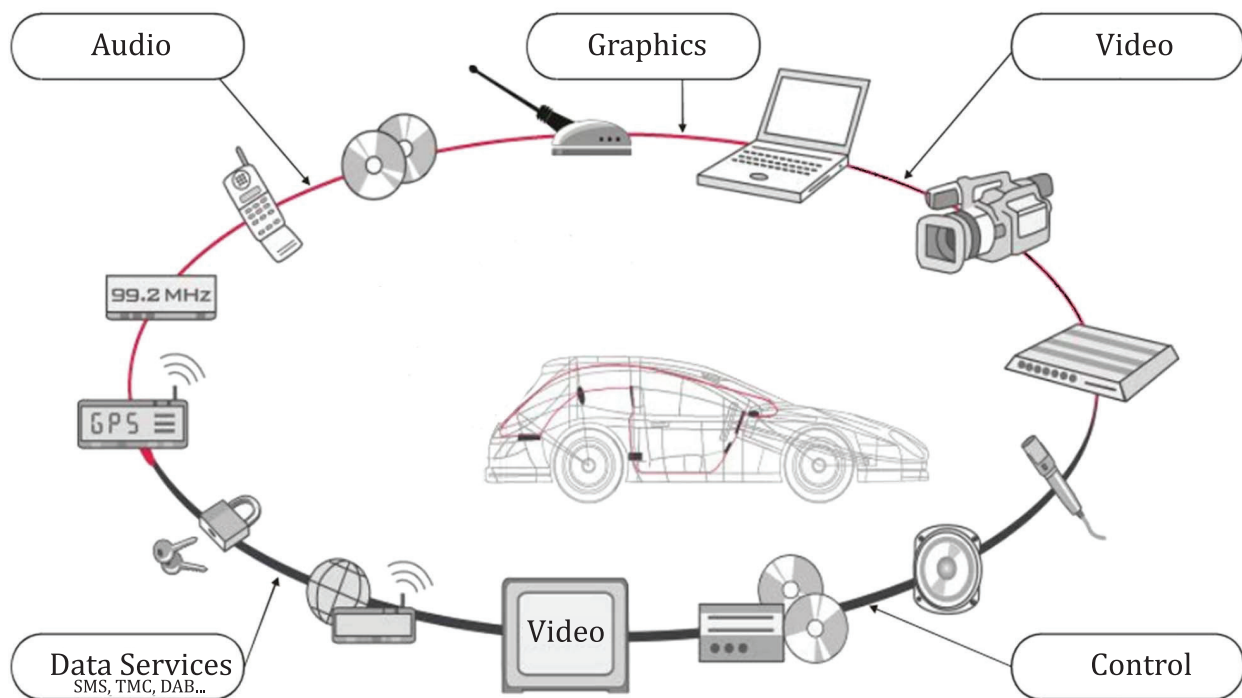


Figure 1 — MOST network example

The MOST communication technology provides:

- synchronous and isochronous streaming,
- small overhead for administrative communication control,
- a functional and hierarchical system model,
- API standardization through a function block (FBlock) framework,
- free partitioning of functionality to real devices,
- service discovery and notification, and
- flexibly scalable automotive-ready Ethernet communication according to ISO/IEC/IEEE 8802-3[2].

MOST is a synchronous time-division-multiplexing (TDM) network that transports different data types on separate channels at low latency. MOST supports different bit rates and physical layers. The network clock is provided with a continuous data signal.

1) MOST® is the registered trademark of Microchip Technology Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO.

Within the synchronous base data signal, the content of multiple streaming connections and control data is transported. For streaming data connections, bandwidth is reserved to avoid interruptions, collisions, or delays in the transport of the data stream.

MOST specifies mechanisms for sending anisochronous, packet-based data in addition to control data and streaming data. The transmission of packet-based data is separated from the transmission of control data and streaming data. None of them interfere with each other.

A MOST network consists of devices that are connected to one common control channel and packet channel.

In summary, MOST is a network that has mechanisms to transport the various signals and data streams that occur in multimedia and infotainment systems.

The ISO standards maintenance portal (<https://standards.iso.org/iso/>) provides references to MOST specifications implemented in today's road vehicles because easy access via hyperlinks to these specifications is necessary. It references documents that are normative or informative for the MOST versions 4V0, 3V1, 3V0, and 2V5.

The ISO 21806 series has been established in order to specify requirements and recommendations for implementing the MOST communication technology into multimedia devices and to provide conformance test plans for implementing related test tools and test procedures.

To achieve this, the ISO 21806 series is based on the open systems interconnection (OSI) basic reference model in accordance with ISO/IEC 7498-1^[1] and ISO/IEC 10731^[3], which structures communication systems into seven layers as shown in [Figure 2](#). Stream transmission applications use a direct stream data interface (transparent) to the data link layer.

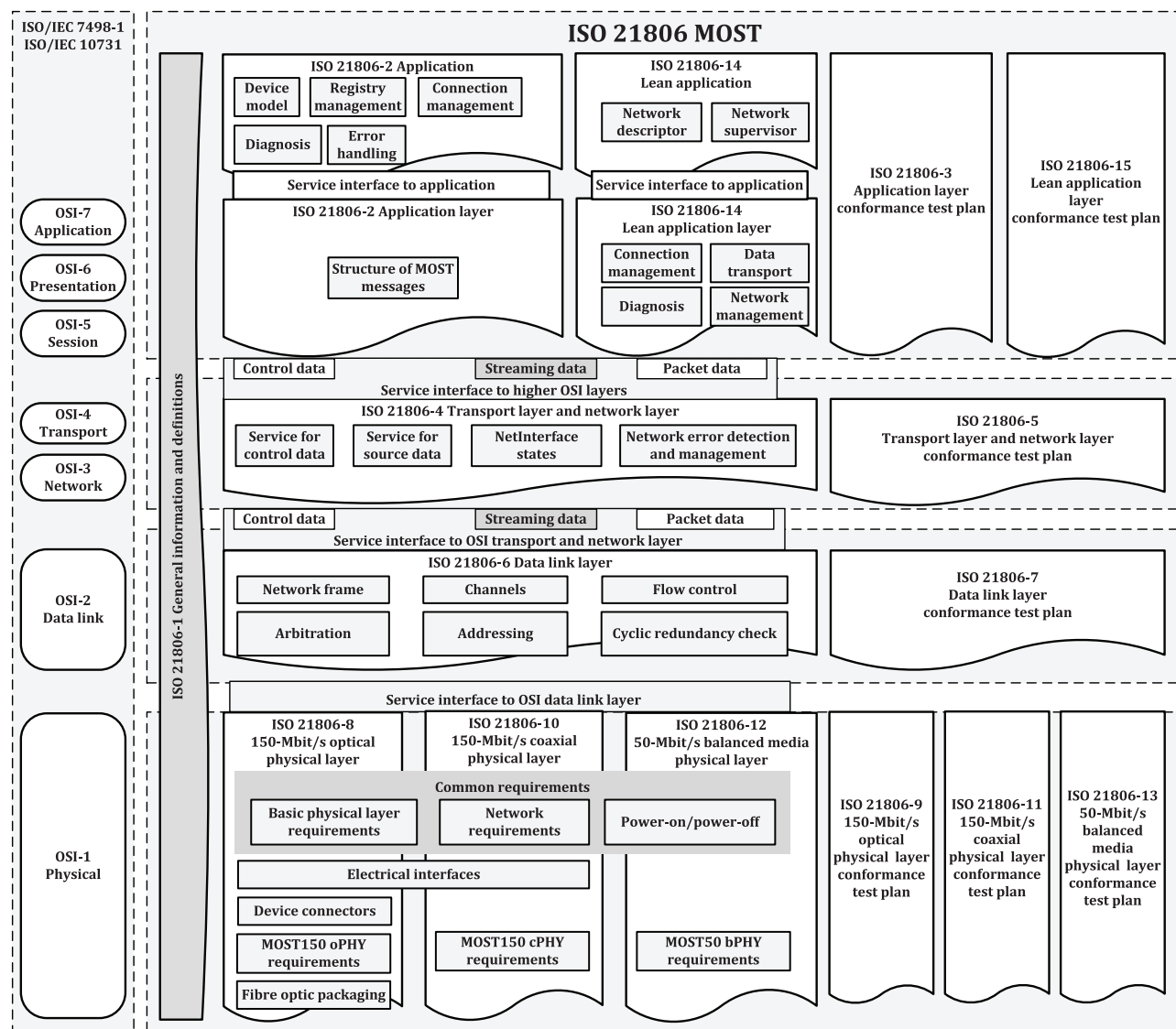


Figure 2 — The ISO 21806 series reference according to the OSI model

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO shall not be held responsible for identifying any or all such patent rights.

Road vehicles — Media Oriented Systems Transport (MOST) —

Part 3: Application layer conformance test plan

1 Scope

This document specifies the conformance test plan (CTP) for the application layer for MOST, a synchronous time-division-multiplexing network, as specified in ISO 21806-2.

This document specifies conformance test cases (CTCs) in the following categories:

- device model;
- data and basic data types;
- registry management;
- connection management;
- error management;
- diagnosis.

Interoperability testing is not in the scope of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

ISO/IEC 9646-1:1994, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts*

ISO 21806-1:2020, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 1: General information and definitions*

ISO 21806-2:2020, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 2: Application layer*

ISO 21806-4:2020, *Road vehicles — Media Oriented Systems Transport (MOST) — Part 4: Transport layer and network layer*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21806-1, ISO 21806-2, ISO 21806-4, ISO/IEC 9646-1, and the following apply.

ISO and IEC maintain terminological databases for use in standardisation at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <http://www.electropedia.org/>

**3.1
REPEAT**

pseudo code command for an iteration

**3.2
REPEAT END**

pseudo code command for ending an iteration

4 Symbols and abbreviated terms

4.1 Symbols

--- empty cell/undefined

4.2 Abbreviated terms

CTC	conformance test case
CTP	conformance test plan
CR	central registry
DR	decentral registry
IUT	implementation under test
LT	lower tester
MPI	maximum position information
MSC	Message Sequence Chart
NCE	network change event
OSI	Open Systems Interconnection
UT	upper tester

5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731^[3] and ISO/IEC 9646-1 for conformance test system set-up.

6 CTP overview

6.1 Test set-up

All CTCs are based on the same test set-up with an upper tester (UT) and a lower tester (LT). The LT contains the lower tester pre-IUT (LT pre-IUT) and the lower tester post-IUT (LT post-IUT).

[Figure 3](#) specifies the test set-up.

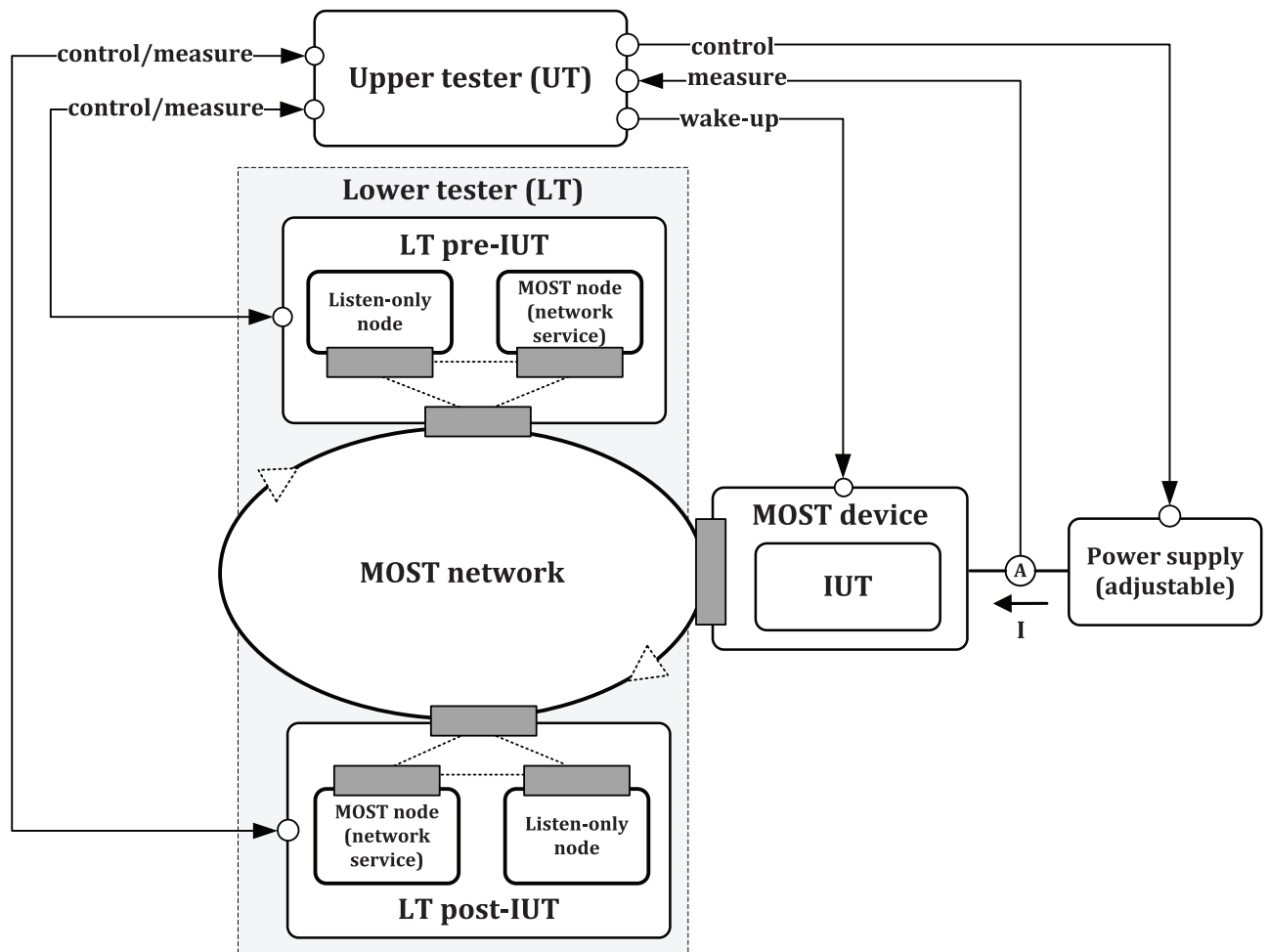


Figure 3 — Test set-up

The LT pre-IUT and the LT post-IUT implement the application layer services and the lower layer services of a MOST node in accordance with the ISO 21806 series. They also contain a listen-only node in front of the MOST node to log the whole communication. The MOST node is able to operate as TimingMaster or TimingSlave; alternatively, it can be physically disconnected from the MOST network. If it is disconnected, the associated LT pre-IUT or LT post-IUT serves as listen-only node.

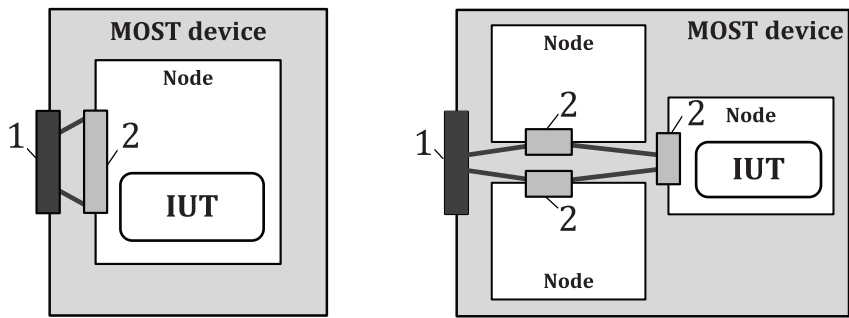
Every CTC specifies the roles of the LT pre-IUT and the LT post-IUT.

During testing of the MOST device that implements the IUT, avoid over-temperature by following the manufacturer recommendations regarding cooling.

The power supply of the MOST device that contains the IUT is adjustable and the power consumption can be monitored by the UT. This is necessary to determine whether a node has entered `s_NetInterface_Sleep`.

A MOST device contains one or more nodes, which are connected to an external MOST physical interface. One of the nodes contains the implementation under test (IUT). All tests and timings, specified by the CTP, are related to the external MOST physical interface.

Figure 4 shows a MOST device with one node and a MOST device with three internal nodes.



- 1 external MOST physical interface
- 2 internal MOST physical interface

Figure 4 — MOST device with one node and MOST device with three nodes

6.2 Conformance test plan organisation

CTCs are independent of one another. Each CTC checks the behaviour of the IUT for requirements stated in ISO 21806-2. Within CTCs, which require variations of individual parameters, each specified value of the parameter is iterated.

The measurement uncertainty for each CTC shall be in accordance with [Annex A](#).

7 CTP general information

7.1 CTC remarks

7.1.1 Timer naming

For conformance testing of the IUT, the UT and LT need minimum and maximum timers. The names of the timers used by this document are based on ISO 21806-2 and ISO 21806-4. To obtain the timer name, for minimum and maximum, “_min” and “_max” are appended, respectively. [Table 1](#) shows a timer naming definition example for t_{Config} .

Table 1 — Timer naming example

Name	Minimum value name	Typical value name	Maximum value name	Unit	Purpose
t_{Config}	t_{Config_min}	t_{Config}	t_{Config_max}	ms	Time before <code>ev_Init_Error_Shutdown</code> or delay for RBD result.

7.1.2 Deadlock prevention

This document specifies the timeouts $t_{DeadLockShort}$, $t_{DeadLockMid}$, and $t_{DeadLockLong}$ to prevent deadlock situations during conformance testing. These are the default values:

- $t_{DeadLockShort}$: 1 s;
- $t_{DeadLockMid}$: 20 s;
- $t_{DeadLockLong}$: 5 min.

These timeouts are only relevant for conformance testing and may be extended.

7.1.3 Un-initialised logical node address

The variable `uninitialised_node_address` is defined as the address of an un-initialised node, which is specified in ISO 21806-2.

7.1.4 Addresses of MOST nodes in the LT

The address of a MOST node in the LT is the default logical node address corresponding to the node position.

If this address is in conflict with the address of a node that contains the IUT (e.g. if a supplier uses static addresses in the dynamic address range), the affected MOST node in the LT shall use a valid free address.

7.1.5 Device manufacturer information list

This list contains all information that is provided by the device manufacturer for conformance testing. It also includes remarks and references to corresponding CTCs.

[Table 2](#) shows the device manufacturer information list, which does not include information stored in FBlock EnhancedTestability.

Table 2 — Device manufacturer information list

Category	Item/property	Description	Reference to CTC
MOST network configuration	IUT in the TimingMaster	Determines whether the IUT is part of the TimingMaster.	All CTCs
	IUT in the NetworkMaster	Determines whether the IUT is part of the NetworkMaster.	All CTCs
	IUT in the PowerMaster	Determines whether the IUT is part of the PowerMaster.	All CTCs
	IUT in the connection manager	---	CTC_3.1-3, CTC_3.1-4, CTC_3.1-5, CTC_3.1-6, CTC_3.2-3, CTC_3.2-4, CTC_3.2-5, CTC_3.2-6, CTC_3.2-7, CTC_3.2-8, CTC_3.2-9, CTC_3.2-14
	Multi-node device	If the IUT is part of a MOST device that contains more than one node, the following information is provided: — number of nodes in the MOST device; — topology of the MOST device (position of PowerMaster and TimingMaster/ NetworkMaster); — position of the node that contains the IUT.	All CTCs
	IUT sample frequency	If the IUT is not part of the TimingMaster, the LT provides the correct network frame rate (44,1 kHz or 48,0 kHz).	All CTCs
	Required value of boundary descriptor (if the TimingMaster is in the LT)	Value of the boundary descriptor. Unless otherwise stated, all CTCs are performed with this value of the boundary descriptor.	All CTCs
	$mi_{MaxInvalidReg}$	The maximum number of permitted conflicting node address registrations by a NetworkSlave.	CTC_2.6.2-3a
	$mi_{MaxSetNewInstID}$	When an invalid InstID registration occurs, the NetworkMaster sends a request to the NetworkSlave for setting a new InstID.	CTC_2.6.2-6
	t_{Config_max}	Time before ev_Init_Error_Shutdown or delay for RBD result.	CTC_2.1.1-6b
	$t_{ConfigurationAnnounce}$	Limit for the NetworkMaster to set the central registry state.	CTC_2.6.2-4a, CTC_2.6.2-5
	$t_{WaitForAnswer_min}$ $t_{WaitForAnswer_max}$	Time the NetworkMaster waits for all NetworkSlaves to respond.	CTC_2.6.2-1, CTC_2.6.2-3b, CTC_2.6.2-5

Table 2 (continued)

Category	Item/property	Description	Reference to CTC
Power management	Node that contains the IUT supports <code>s_NetInterface_Sleep</code>	Determines whether the node that contains the IUT supports <code>s_NetInterface_Sleep</code> : — yes: the MOST device that contains the IUT reduces its power consumption below threshold before timeout expires; — no: the reduction of power consumption is not detectable.	CTC_2.3.2-3
	<code>s_NetInterface_Sleep</code> : <code>I_NetInterfaceSleep_Threshold</code>	Threshold of current for <code>s_NetInterface_Sleep</code> detection	See 7.1.6.
	<code>s_NetInterface_Sleep</code> : <code>t_PwrSwitchOffDelay_min</code> <code>t_PwrSwitchOffDelay_max</code>	<code>t_PwrSwitchOffDelay_min</code> Specific timeout for <code>s_NetInterface_Sleep</code> ; after the end of network activity, the node that contains the IUT does not enter <code>s_NetInterface_Sleep</code> (reduced power consumption) before <code>t_PwrSwitchOffDelay_min</code> expires. <code>t_PwrSwitchOffDelay_max</code> MOST device specific timeout for <code>s_NetInterface_Sleep</code> ; after the end of network activity, the node that contains the IUT enters <code>s_NetInterface_Sleep</code> (reduced power consumption) before <code>t_PwrSwitchOffDelay_max</code> expires.	CTC_2.3.2-3, CTC_2.6.4-1
	Wake-up preconditions	Preconditions for the node that contains the IUT for wake-up. Supplemented by information whether the node that contains the IUT needs additional conditions during operation (e.g. ignition ON) to stay in <code>s_NetInterface_Normal_Operation</code> .	See 7.1.6.
	Node that contains the IUT is capable of waking via network startup (i.e. switching on its MOST output)	---	CTC_2.4.1-2
	Delay between connection to power (of the MOST device that contains the IUT) and the ability of the node that contains the IUT to detect wake-up events	Potentially, the UT (see Figure 3) waits for a short period of time between connecting the MOST device that contains the IUT to power and switching on the MOST output to wake up the node that contains the IUT. Otherwise, the node that contains the IUT does not detect a wake-up event.	All CTCs
	<code>s_NetInterface_Normal_Operation</code> : Delay until all FBlocks of the node that contains the IUT are available after <code>Configuration_Status(OK)</code> (equivalent to <code>t_WaitForApplication</code>)	This delay covers: — NetworkMaster: period of time the node that contains the IUT needs to add own FBlocks to its central registry after <code>Configuration_Status(OK)</code> ; — NetworkSlave: delay between <code>ev_Init_Ready</code> and availability of application.	See 7.1.6. CTC_2.6.2-3b
	<code>t_WaitBeforeScan</code>	Specific limit for the NetworkMaster to start an FBlock scan.	CTC_2.4.1-9, CTC_2.6.2-5, CTC_2.6.4-8

Table 2 (continued)

Category	Item/property	Description	Reference to CTC
Addressing	Node that contains the IUT uses static node address in dynamic address range	If the node that contains the IUT uses a static logical node address that is in the specified dynamic address range, the address is provided.	See 7.1.6. CTC_2.6.4-4
	Free address	Logical node address that may be used by a MOST node in the LT during testing.	CTC_2.6.4-3
	Free FBlock range	FBlocks that are not used by the node that contains the IUT and which may be used by the UT.	CTC_2.6.2-4b, CTC_2.6.2-4c
	Group address of the node that contains the IUT	---	CTC_2.7-1
General communication	t_{Property}	Limit for responding to a command that reads a property.	CTC_2.1.0-1, CTC_2.1.0-2, CTC_2.8.3-2, CTC_2.8.4-3, CTC_3.0-1, CTC_3.1-1, CTC_3.2-1
	$t_{\text{NotificationProperty}}$	Limit for reacting to a Notification.Set message.	CTC_2.8.3-1a, CTC_2.8.3-1b, CTC_2.8.3-7
Physical parameter (voltage levels)	$U_{\text{IUT_Operating}}$	At this voltage level, the MOST device that contains the IUT operates normally. Unless otherwise stated, all CTCs are performed at this voltage level.	All CTCs
Messaging	Node that contains the IUT supports segmented messages	The node that contains the IUT is able to send and receive segmented messages.	CTC_2.8.4-2, CTC_2.8.4-3, CTC_2.8.4-7, CTC_2.8.4-8
Sink/Source	List of FBlocks, containing sink and/or source functionality	The list contains all FBlocks reported by NetBlock.FBlockIDs.Status.	CTC_3.0-1
	MOST devices with sinks: List of all supported sink numbers with ContentType, ContentDescription (data type of the parameter) and TransmissionClass	---	CTC_3.1-1, CTC_3.1-3, CTC_3.1-4, CTC_3.1-5, CTC_3.1-6
	MOST devices with sources: List of all supported source numbers with ContentType, ContentDescription (data type of the parameter) and TransmissionClass	---	CTC_3.2-1, CTC_3.2-3, CTC_3.2-4, CTC_3.2-5, CTC_3.2-6, CTC_3.2-7, CTC_3.2-14
	MOST devices with sources: BlockWidth and ConnectionLabel	---	CTC_3.2-3
	Node that contains the IUT supports SourceActivity	---	CTC_3.2-14

7.1.6 States of the node that contains the IUT

Table 3 specifies how the NetInterface state `s_NetInterface_Normal_Operation` is effectuated and detected in the node that contains the IUT.

Table 3 — Effectuate and detect s_NetInterface_Normal_Operation

Effectuate state	Detect state
a) The IUT is contained in a NetworkSlave: — the UT shall start the network; — wait for the node that contains the IUT to open its bypass (MPI is nominal^a); — send <code>NetworkMaster.Configuration.Status(NotOK);</code> — perform an FBlock scan (including retries if the address of the node that contains the IUT is invalid); — send <code>NetworkMaster.Configuration.Status(OK);</code> — wait for $t_{\text{WaitForApplication}}$	a) The IUT is contained in a NetworkSlave: the node that contains the IUT responds to <code>NetBlock.FBlockIDs.Get</code> .
b) The IUT is contained in the NetworkMaster: — the UT shall behave like a NetworkSlave; it shall process and respond to all requests from the node that contains the IUT so that the node can enter central registry state OK; — the UT shall respond to an FBlock scan by the node that contains the IUT; additionally, the UT shall wait for the node to open its bypass (MPI is nominal^a) if the node is part of a multi-node device; — finally, the UT shall wait for $t_{\text{WaitForApplication}}$	b) The IUT is contained in the NetworkMaster: the node that contains the IUT responds to <code>NetBlock.FBlockIDs.Get</code> .
^a The nominal MPI is the total number of nodes in the set-up, based on the IUT manufacturer information and the test equipment.	

Table 4 specifies how the NetInterface state `s_NetInterface_Sleep` is effectuated and detected in the node that contains the IUT.

Table 4 — Effectuate and detect s_NetInterface_Sleep state

Effectuate state	Detect state
Switch off MOST output	a) When <code>s_NetInterface_Sleep</code> is detectable the following applies: — when monitoring power consumption, the current reaches or drops below $I_{\text{NetInterfaceSleep_Threshold}}$; — if the LT detects network activity when $t_{\text{PwrSwitchOffDelay_max}}$ expires, the test shall be stopped; the timer shall be started as soon as the node that contains the IUT switches off the MOST output; — if the current does not reach or drop below $I_{\text{NetInterfaceSleep_Threshold}}$ within timeout $t_{\text{PwrSwitchOffDelay_max}}$, the test shall be stopped. b) When <code>s_NetInterface_Sleep</code> is not detectable the following applies: — when the timeout $t_{\text{PwrSwitchOffDelay_max}}$ expires and if the LT does not detect network activity, it shall be assumed that the node that contains the IUT is in <code>s_NetInterface_Sleep</code>, independent of power consumption; — if the LT detects network activity when $t_{\text{PwrSwitchOffDelay_max}}$ expires, the test shall be stopped.

Table 5 specifies how the NetInterface state `s_NetInterface_Off` is effectuated and detected in the node that contains the IUT.

Table 5 — Effectuate and detect s_NetInterface_Off state

Effectuate state	Detect state
The LT shall switch off the MOST output.	No network activity

7.1.7 Procedures

Table 6 specifies the procedures of the LT.

Table 6 — Procedures of the LT

Purpose	Description
Perform wake-up	a) If the LT contains the TimingMaster, it shall execute this sequence to perform a wake-up: - switch on the MOST output; - wait for network activity (timeout $t_{\text{DeadLockShort}}$); - wait for stable lock (timeout $t_{\text{DeadLockMid}}$). b) If the LT does not contain the TimingMaster, it shall execute this sequence to perform a wake-up: - generate a wake-up event; - wait for network activity; - switch on the MOST output; - wait for stable lock; - wait for the lock flag to evaluate to true. If the LT does not generate a wake-up event and detects network activity, it shall switch on the MOST output. In some cases, the node that contains the IUT needs some preconditions for wake-up. These preconditions are established before testing is started. The preconditions depend on the device manufacturer. If <code>EnhancedTestability.AutoWakeup</code> is triggered, the node that contains the IUT does not enter <code>s_NetInterface_Sleep</code> before creating the corresponding wake-up event. The node enters <code>s_NetInterface_Off</code>. This state is not detectable by monitoring the power consumption of the MOST device that contains the IUT. With entering state <code>s_NetInterface_Off</code>, the node that contains the IUT switches off the MOST output.
Perform shutdown	a) If the IUT is part of the PowerMaster, the LT shall execute this sequence to perform shutdown: - trigger shutdown by means of <code>FBlock EnhancedTestability</code>; - if no network activity is detected, the node that contains the IUT has performed shutdown; - if $t_{\text{DeadLockMid}}$ expires after triggering shutdown and the LT still detects network activity, it shall switch off the MOST output. b) If the IUT is part of a PowerSlave, the LT shall switch off the MOST output to perform shutdown. c) If the IUT is part of the PowerMaster, no preconditions are established that prevent the node that contains the IUT from performing shutdown.
Generate unlock	To generate an unlock event of predictable duration, the LT

Table 6 (continued)

Purpose	Description
	— shall invalidate or delay the preamble at the beginning of at least every third network frame during the period of unlock (see ISO 21806-6), and — shall avoid a PLL unlock.
Network change event with unlock	The NCE shall be generated between the TimingMaster and the node that contains the IUT.
Network change event without unlock	The NCE shall be generated between the node that contains the IUT and the TimingMaster.

7.1.8 Violation of prerequisites of the CTC

If the node that contains the IUT does not meet the prerequisites of the CTC (such as network activity, lock, FBlock scan performed, central registry state OK), the CTC results in "IUT not ok: The IUT does not meet the prerequisites".

7.2 CTC items

7.2.1 FBlock EnhancedTestability

The FBlock EnhancedTestability is used to trigger sequences that are specified in the CTC. These are normally triggered by a project specific, sometimes complicated, mechanism. FBlock EnhancedTestability implements neither notification nor processing messages. The UT shall initialise FBlock EnhancedTestability every time the `s_NetInterface_Normal_Operation` state is reached. The FBlock is only available during `s_NetInterface_Normal_Operation`. All properties are reset to their default state when entering `s_NetInterface_Normal_Operation`, unless otherwise stated. The functions in this FBlock describe a general interface for starting functionality partly implemented in the application, partly in the MOST network service. If an application callback returns wrong or unexpected values, the FBlock sends an error message with `ErrorCode 0B16` (device malfunction).

If the FBlock EnhancedTestability returns errors, the corresponding CTC result is indicated as "IUT not ok".

7.2.2 Multi-node devices

A multi-node device is a MOST device with an external MOST physical interface, which is connected to more than one internal node.

When dealing with devices with several external MOST physical interfaces, each interface shall be treated as a separate IUT. One of those interfaces may belong to a multi-node device. In a multi-node device, each node shall be treated individually.

NOTE A MOST device passes conformance testing successfully if all contained IUTs pass all relevant CTCs.

7.2.3 Node kinds excluded from conformance testing

CTCs for remote controlled nodes according to ISO 21806-2 are not in the scope of this document.

8 CTC specification

8.1 Static FBlock behaviour

8.1.1 CTC_2.1.0-1 – Generic FBlock property test

[Table 7](#) specifies the CTC_2.1.0-1 - Generic FBlock property test.

Table 7 — CTC_2.1.0-1 - Generic FBlock property test

Item	Content
CTC # – Title	CTC_2.1.0-1 - Generic FBlock property test
Purpose	This CTC verifies that FBlock communication with properties of the NetBlock, the Network-Master, sinks, and sources is possible. This CTC applies to all MOST devices.
Reference	ISO 21806-2:2020, 7.6 AL – Operation type (OPType)
Prerequisite	The UT shall effectuate <code>s_NetInterface_Normal_Operation</code> in the node that contains the IUT.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall set the <code>FBlockID</code> and <code>InstID</code> to the values of the current test iteration. 2. The UT, using the LT pre-IUT, shall send <code>FBlockID.InstID.FktIDs.Get</code> to the IUT to retrieve the list of implemented functions of the IUT. 3. The UT, in an outer loop, shall loop through the <code>FktIDs</code> that were returned in the list and correspond to properties. 4. The UT, in an inner loop, shall loop through the <code>OPTypes</code> (0_{16} to F_{16}). 5. UT: for every iteration of the loop, the UT, using the LT pre-IUT, shall send <code>FBlockID.InstID.FktID.OPType</code> to the IUT. 6. The UT shall evaluate the response of the IUT as specified in Table 8. 7. The UT shall shut down the IUT and re-establish the prerequisites for the next test iteration, after processing all inner and outer loop iterations.
Iteration	REPEAT Step 1 to Step 7 with the NetBlock and every available <code>InstID</code> (obtained by <code>NetBlock.FBlockIDs</code>) of the following FBlocks of the — IUT: NetworkMaster; — IUT: FBlocks containing sink functions (identified by CTC TEST_GSI_GSO_identification); — IUT: FBlocks containing source functions (identified by CTC TEST_GSI_GSO_identification). REPEAT END.
Expected response	Step 6: IUT ok: the IUT provides permitted responses as specified in Table 8 . Step 6: IUT not ok: the IUT does not respond as expected to the requests to properties.

Table 7 (continued)

Item	Content
Remark	- Responses are only evaluated from actually tested FBlockID, InstID and FktID. - Any other message of the node that contains the IUT is ignored during the CTC. - The FBlock library determines whether the function is a property or a method. - Usually, a MOST device does not implement all functions and OPTypes of the FBlock library; if implemented, they are expected to behave according to the FBlock library. - No range checks are performed. - In the case that the OPType test allows “no answer” as permitted response, the observation time period is $2 \times t_{\text{Property}}$ for properties respectively $t_{\text{DeadLockMid}}$ for methods. - For the NetBlock, error 03₁₆ (function not available) is also a permitted response. - All commands (OPType 0₁₆ to 8₁₆) from the node that contains the IUT are ignored for all FBlocks. - For the NetBlock and the NetworkMaster FBlock, OPTypes 9₁₆ to F₁₆ are not tested.

Table 8 specifies the OPTypes for CTC_2.1.0-1 - Generic FBlock property test.

Table 8 — OPTypes for CTC_2.1.0-1 - Generic FBlock property test

OPType	OPType name and parameters	Permitted response
0 ₁₆	Set (without parameter)	— Status; — No answer (within $2 \times t_{\text{Property}}$); — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
1 ₁₆	Get (without parameter)	— Status; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
2 ₁₆	SetGet (without parameter)	— Status; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
3 ₁₆	Increment(1)	— Status; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
4 ₁₆	Decrement(1)	— Status; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
5 ₁₆	Not allowed	Not allowed
6 ₁₆	StartResultAck (SenderHandle 1234 ₁₆)	— ErrorAck(SenderHandle 1234₁₆, ErrorCode 04₁₆); — Error(ErrorCode 04₁₆).
7 ₁₆	AbortAck (SenderHandle 1234 ₁₆)	— ErrorAck(SenderHandle 1234₁₆, ErrorCode 04₁₆); — Error(ErrorCode 04₁₆).
8 ₁₆	StartAck (SenderHandle 1234 ₁₆)	— ErrorAck(SenderHandle 1234₁₆, ErrorCode 04₁₆); — Error(ErrorCode 04₁₆).
9 ₁₆	ErrorAck	No answer
A ₁₆	ProcessingAck	No answer
B ₁₆	Processing	No answer
C ₁₆	Status	No answer

Table 8 (continued)

OType	OType name and parameters	Permitted response
D ₁₆	ResultAck	No answer
E ₁₆	Not allowed	Not allowed
F ₁₆	Error	No answer

8.1.2 CTC_2.1.0-2 – Generic FBlock method test

[Table 9](#) specifies the CTC_2.1.0-2 - Generic FBlock method test.

Table 9 — CTC_2.1.0-2 - Generic FBlock method test

Item	Content
CTC # – Title	CTC_2.1.0-2 - Generic FBlock method test
Purpose	This CTC verifies that FBlock communication with methods of the NetBlock, the NetworkMaster, sinks, and sources is possible. This CTC applies to all MOST devices.
Reference	ISO 21806-2:2020, 7.6 AL – Operation type (OType)
Prerequisite	The UT shall effectuate <code>s_NetInterface_Normal_Operation</code> in the node that contains the IUT.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node.
Step	1. The UT shall set the <code>FBlockID</code> and <code>InstID</code> to the values of the current test iteration. 2. The UT shall use the LT pre-IUT to send <code>FBlockID.InstID.FktIDs.Get</code> to the IUT to retrieve the list of implemented functions of the IUT. 3. The UT in an outer loop, the UT shall loop through the <code>FktIDs</code> that correspond to methods, which are reported in the list. 4. The UT in an inner loop, the UT shall loop through the <code>OTypes</code> (O₁₆ to F₁₆). 5. The UT for every iteration of the loop, the UT, using the LT pre-IUT, shall send <code>FBlockID.InstID.FktID.OType</code> to the IUT. 6. The UT shall evaluate the response of the IUT as specified in Table 10. 7. The UT after processing all inner and outer loop iterations, the UT shall shutdown the IUT and re-establish the prerequisites for the next test iteration.
Iteration	Repeat Step 1 to Step 7 with the NetBlock and every available <code>InstID</code> (obtained by <code>NetBlock.FBlockIDs</code>) of the following FBlocks of the — IUT: NetworkMaster; — IUT: FBlocks containing sink functions (identified by <code>TEST_GSI_GSO_identification</code>); — IUT: FBlocks containing source functions (identified by <code>TEST_GSI_GSO_identification</code>). Repeat END.
Expected response	Step 6: IUT ok: the IUT responds as expected to the requests to methods. Step 6: IUT not ok: the IUT does not respond as expected to the requests to methods.

Table 9 (continued)

Item	Content
Remark	- Responses are only evaluated from actually tested FBlockID, InstID and FktID. - Any other message of the node that contains the IUT are ignored during the CTC. - The FBlock library determines whether the function is a property or a method. - Usually, a MOST device does not implement all functions and OPTypes of the FBlock library; if implemented, they are expected to behave according to the FBlock library. - There are no range checks. - Each OPType test presumes that the previous test completes the communication process, for example if after StartResult the Result is not received, then the process is expected to be terminated by sending Abort. If Abort fails, the IUT is reset to finish the test step and continue with the next step. - In the case that the OPType test allows “no answer” as permitted response, the observation time period is $2 \times t_{\text{Property}}$ for properties respectively $t_{\text{DeadLockMid}}$ for methods. - In the case that the OPType test allows Processing as permitted response, Processing might be received before result or an error, maximum 50 Processing are accepted by the UT, then Abort is sent. - For the NetBlock, error 03₁₆ (function not available) is also a permitted response. - All commands (OPType 0₁₆ to 8₁₆) from the node that contains the IUT are ignored for all FBlocks. - For the NetBlock and the NetworkMaster FBlock, OPTypes 9₁₆ to F₁₆ not tested. - During this CTC it is considered that sending of an error is optional in function DTCP_Control.

Table 10 specifies the OPTypes for CTC_2.1.0-2 - Generic FBlock method test.

Table 10 — OPTypes for CTC_2.1.0-2 - Generic FBlock method test

OPType	OPType name and parameters	Permitted response
0 ₁₆	Start	— None; — Result; — Processing; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
1 ₁₆	Abort	— None; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
2 ₁₆	StartResult	— ProcessingResult; — All errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
3 ₁₆	Increment	— Error 04 ₁₆
4 ₁₆	Decrement	— Error 04 ₁₆
5 ₁₆	Not allowed	Not allowed

Table 10 (continued)

OType	OType name and parameters	Permitted response
6 ₁₆	StartResultAck (SenderHandle 1234 ₁₆)	— ProcessingAck(SenderHandle 1234₁₆); — ResultAck(SenderHandle 1234₁₆); — All ErrorAcks (SenderHandle 1234₁₆) except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆; — All Errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
7 ₁₆	AbortAck (SenderHandle 1234 ₁₆)	— None; — All ErrorAcks (SenderHandle 1234₁₆) except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆; — All Errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
8 ₁₆	StartAck (SenderHandle 1234 ₁₆)	— None; — ResultAck(SenderHandle 1234₁₆); — ProcessingAck(SenderHandle 1234₁₆); — All ErrorAcks (SenderHandle 1234₁₆) except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆; — All Errors except 01₁₆, 02₁₆, 03₁₆, 0A₁₆, 0C₁₆.
9 ₁₆	ErrorAck	No answer
A ₁₆	ProcessingAck	No answer
B ₁₆	Processing	No answer
C ₁₆	Result	No answer
D ₁₆	ResultAck	No answer
E ₁₆	Not allowed	Not allowed
F ₁₆	Error	No answer

8.2 Power management

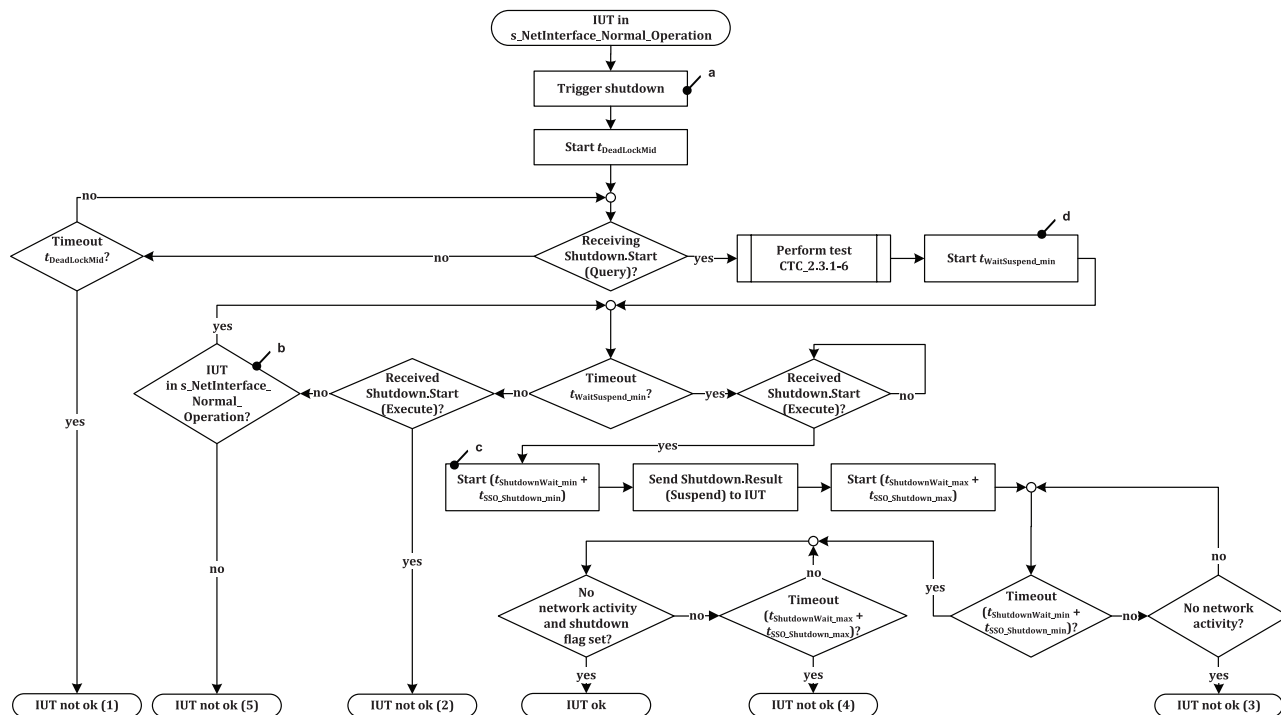
8.2.1 Power management – PowerMaster

8.2.1.1 CTC_2.3.1-3 – Timeout execute/timeout suspend test

[Table 11](#) specifies the CTC_2.3.1-3 – Timeout execute/timeout suspend test.

Table 11 — CTC_2.3.1-3 – Timeout execute/timeout suspend test

Item	Content
CTC # – Title	CTC_2.3.1-3 – Timeout execute/timeout suspend test
Purpose	This CTC verifies that the PowerMaster is capable of correctly performing a shutdown, when there are no shutdown-suspend requests from the PowerSlaves within the specified time constraints: — for <code>Shutdown.Start(Query)</code>, $t_{\text{WaitSuspend}}$ is used; — for <code>Shutdown.Start(Execute)</code>, $t_{\text{ShutdownWait}} + t_{\text{SSOShutdown}}$ is used. This CTC applies to all MOST devices that contain the PowerMaster.
Reference	ISO 21806-2:2020, 6.9.3 APP – Network shutdown
Prerequisite	— The UT shall effectuate <code>s_NetInterface_Normal_Operation</code> in the node that contains the IUT. — Preconditions that prevent the node that contains the IUT from performing shutdown shall be switched off.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node. — PowerSlaves in multi-node devices shall be configured to not send <code>NetBlock.Shutdown.Result(Suspend)</code>.
Step	1. The UT shall execute the sequence as specified in Figure 5 .
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT performs a shutdown correctly. Step 1: IUT not ok (1): the IUT does not start the shutdown procedure. Step 1: IUT not ok (2): the IUT sends <code>Shutdown.Start(Execute)</code> too early. Step 1: IUT not ok (3): the IUT switches off the MOST output too early. Step 1: IUT not ok (4): the IUT switches off the MOST output too late or does not set the shutdown flag. Step 1: IUT not ok (5): the IUT does not perform <code>s_NetInterface_Normal_Operation</code> during $t_{\text{WaitSuspend}}$
Remark	---



- a The UT, using the LT pre-IUT, shall send `EnhancedTestability.Shutdown.Start(0016)` to instruct the IUT to perform a shutdown.
The IUT performs a shutdown by sending `NetBlock.Shutdown.Start(Query)`.
- b `s_NetInterface_Normal_Operation` shall be checked every 500 ms.
- c The IUT waits for $t_{\text{ShutdownWait_min}} + t_{\text{SSO_Shutdown_min}}$ before switching off the MOST output. Message is received within $t_{\text{DeadLockLong}}$. Otherwise, the CTC is stopped to avoid deadlocks.
- d The IUT waits for $t_{\text{WaitSuspend_min}}$ (CTC_2.3.1-6 can be performed) before sending `NetBlock.Shutdown.Start(Execute)`. It ignores `NetBlock.Shutdown.Result(Suspend)`.

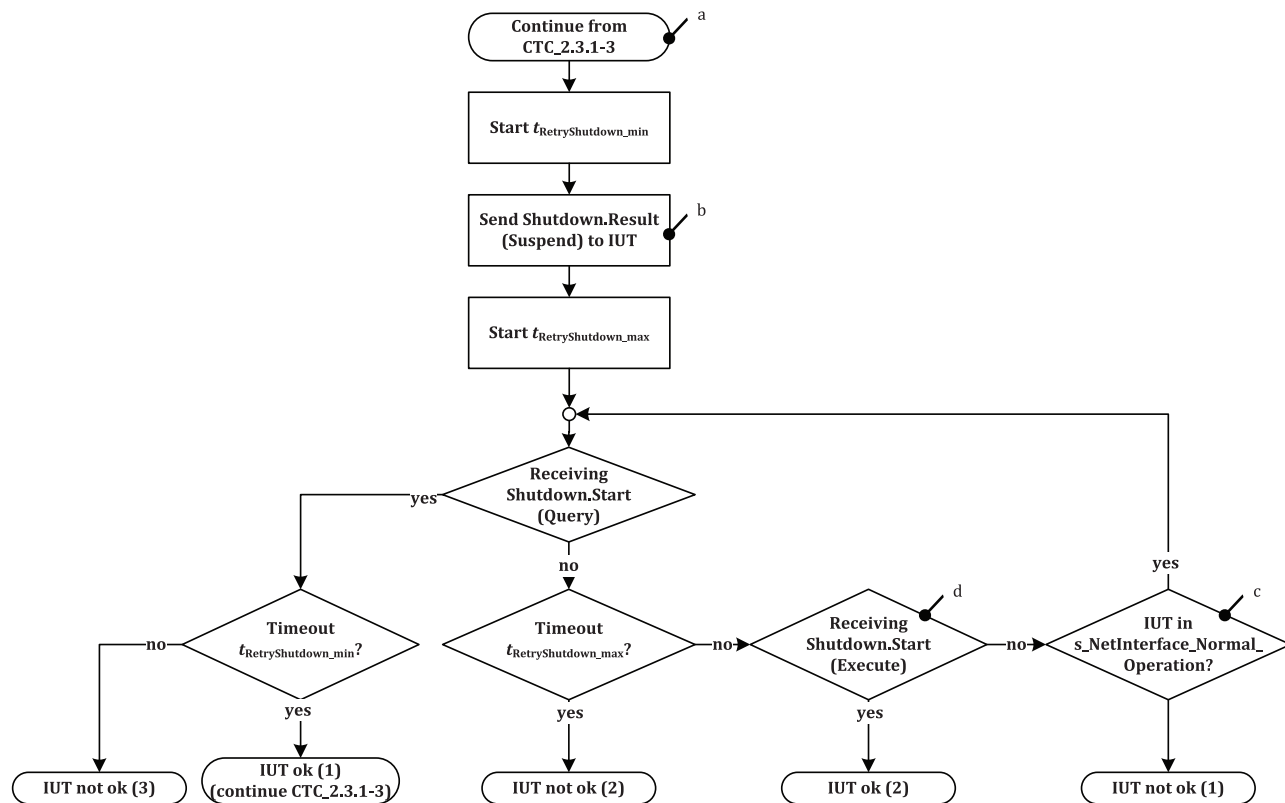
Figure 5 — CTC_2.3.1-3 - Timeout execute/timeout suspend test

8.2.1.2 CTC_2.3.1-6 – Timeout suspend test

Table 12 specifies the CTC_2.3.1-6 – Timeout suspend test.

Table 12 — CTC_2.3.1-6 – Timeout suspend test

Item	Content
CTC # – Title	CTC_2.3.1-6 – Timeout suspend test
Purpose	This CTC verifies that the PowerMaster is capable of postponing the execution of a shutdown due to shutdown-suspend requests from the PowerSlaves and that it repeats the shutdown attempt within the specified time constraints. This CTC applies to all MOST devices that contain the PowerMaster.
Reference	ISO 21806-2:2020, 6.9.3 APP – Network shutdown
Prerequisite	The UT shall effectuate <code>s_NetInterface_Normal_Operation</code> in the node that contains the IUT.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node. — PowerSlaves in multi-node devices shall be configured to not send <code>Shutdown.Result(Suspend)</code>.
Step	1. The UT shall execute the sequence as specified in Figure 6 .
Iteration	Not applicable
Expected response	Step 1: IUT ok (1): the IUT postpones the execution of the shutdown. Step 1: IUT ok (2): the IUT overrides the suspend request. Step 1: IUT not ok (1): the IUT does not remain in <code>s_NetInterface_Normal_Operation</code> during $t_{\text{RetryShutdown}}$. Step 1: IUT not ok (2): the IUT does not send <code>Shutdown.Start(Query)</code>. Step 1: IUT not ok (3): the IUT sends <code>Shutdown.Start(Query)</code> too early.
Remark	---



- a The CTC shall continue from CTC_2.3.1-3.
- b The UT, using the LT pre-IUT, shall send a Shutdown.Result (Suspend) to the IUT before $t_{WaitSuspend}$ expires.
- c The node that contains the IUT stays in `s_NetInterface_Normal_Operation` until $t_{RetryShutdown_min}$ expires. `s_NetInterface_Normal_Operation` shall be checked every 500 ms.
- d The IUT starts a new shutdown attempt before $t_{RetryShutdown_max}$ expires.

Figure 6 — CTC_2.3.1-6 - Timeout suspend test

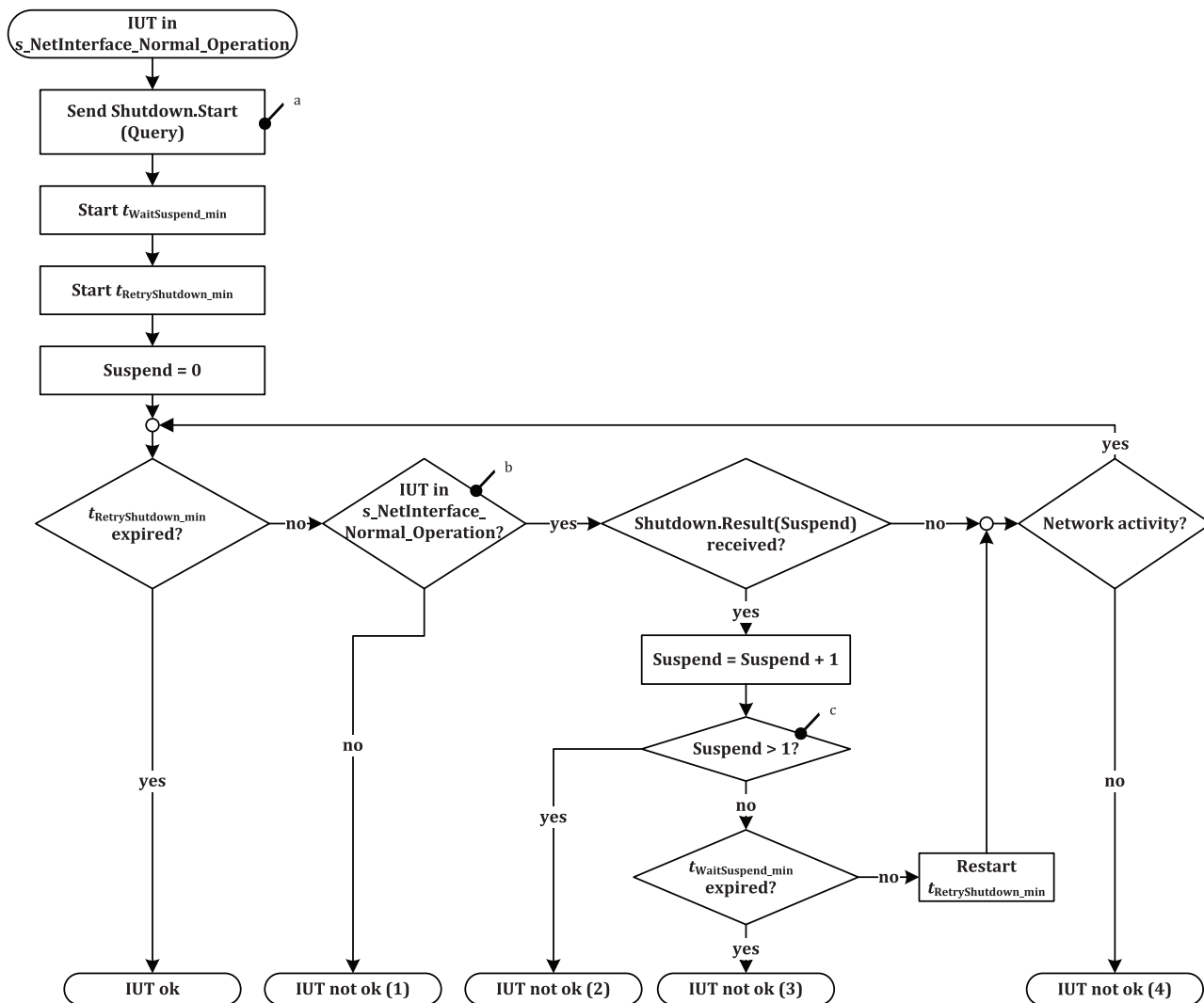
8.2.2 Power management – PowerSlave

8.2.2.1 CTC_2.3.2-2 – Shutdown.Start(Query) test

[Table 13](#) specifies the CTC_2.3.2-2 – Shutdown.Start(Query) test.

Table 13 — CTC_2.3.2-2 – Shutdown.Start(Query) test

Item	Content
CTC # – Title	CTC_2.3.2-2 – Shutdown.Start(Query) test
Purpose	This CTC verifies that a PowerSlave does not switch off the MOST output due to reception of <code>Shutdown.Start(Query)</code>. The CTC also verifies that, if the PowerSlave attempts to suspend the shutdown, it does so within the specified time constraints. This CTC applies to all MOST devices that do not contain the PowerMaster.
Reference	ISO 21806-2:2020, 6.9.3 APP – Network shutdown
Prerequisite	The UT shall effectuate <code>s_NetInterface_Normal_Operation</code> in the node that contains the IUT.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be in listen-only node.
Step	1. The UT shall execute the sequence as specified in Figure 7 .
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT remains in <code>s_NetInterface_Normal_Operation</code>. Step 1: IUT not ok (1): the IUT does not remain in <code>s_NetInterface_Normal_Operation</code> after <code>Shutdown.Start(Query)</code>. Step 1: IUT not ok (2): the IUT sends <code>Shutdown.Result(Suspend)</code> too often. Step 1: IUT not ok (3): the IUT sends <code>Shutdown.Result(Suspend)</code> too late. Step 1: IUT not ok (4): the IUT switches off the MOST output by itself.
Remark	If CTC ends with “IUT ok”, continue with CTC_2.3.2-3 – CTC_2.3.2-3 – Shutdown.Start(Execute) test.



- a The UT, using the LT pre-IUT, shall send `Shutdown.Start (Query)`.
- b The node that contains the IUT remains in `s_NetInterface_Normal_Operation`. `s_NetInterface_Normal_Operation` shall be checked every 500 ms.
- c The IUT is allowed to send `Shutdown.Result (Suspend)` once. In that case, `Shutdown.Result (Suspend)` is sent within `tWaitSuspend_min` after `Shutdown.Start (Query)`.

Figure 7 — CTC_2.3.2-2 - Shutdown.Start(Query) test

8.2.2.2 CTC_2.3.2-3 - Shutdown.Start(Execute) test

Table 14 specifies the CTC_2.3.2-3 - Shutdown.Start(Execute) test.

Table 14 — CTC_2.3.2-3 – Shutdown.Start(Execute) test

Item	Content
CTC # – Title	CTC_2.3.2-3 – Shutdown.Start(Execute) test
Purpose	This CTC verifies that a PowerSlave, after receiving <code>Shutdown.Start(Execute)</code>, remains in <code>s_NetInterface_Normal_Operation</code> until the LT pre-IUT switches off the MOST output. Additionally, it verifies that the IUT does not enter <code>s_NetInterface_Sleep</code> before $t_{PwrSwitchOffDelay}$ expires. This CTC applies to all MOST devices that do not contain the PowerMaster.
Reference	ISO 21806-2:2020, 6.9.3 APP – Network shutdown
Prerequisite	The CTC shall continue directly from CTC_2.3.2-2 – CTC_2.3.2-2 – Shutdown.Start(Query) test.
Set-up	— If the IUT is part of a MOST device that contains the TimingMaster, the LT pre-IUT shall be a TimingSlave. — If the IUT is part of a MOST device that does not contain the TimingMaster, the LT pre-IUT shall be the TimingMaster. — The LT post-IUT shall be a listen-only node. — Power consumption of the IUT is monitored.
Step	1. The UT shall execute the sequence as specified in Figure 8 .
Iteration	Not applicable
Expected response	Step 1: IUT ok: the IUT remains in <code>s_NetInterface_Normal_Operation</code>. Step 1: IUT not ok (1): the IUT does not remain in <code>s_NetInterface_Normal_Operation</code> after reception of <code>Shutdown.Start(Execute)</code>. Step 1: IUT not ok (2): the IUT switches off the MOST output by itself too early. Step 1: IUT not ok (3): the IUT does not switch off the MOST output within $t_{DeadLockShort}$. Step 1: IUT not ok (4): the IUT enters <code>s_NetInterface_Sleep</code> too early.
Remark	---