BS ISO 21111-11:2021

BSI Standards Publication

Road vehicles - In-vehicle Ethernet

Part 11: Application layer to session layer conformance test plans

Documents
Document Sharing Hub

Full document details: BS ISO 21111-11:2021

National foreword

This British Standard is the UK implementation of ISO 21111-11:2021.

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.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2021
Published by BSI Standards Limited 2021

ISBN 978 0 539 16322 3

ICS 43.040.10

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 December 2021.

Amendments/corrigenda issued since publication

Date	Text affected

Documents
Document Sharing Hub

Full document details: BS ISO 21111-11:2021

INTERNATIONAL STANDARD

ISO 21111-11

First edition
2021-12

Road vehicles — In-vehicle Ethernet —

Part 11: Application layer to session layer conformance test plans

Véhicules routiers — Ethernet embarqué —

Partie 11: Plans de test de conformité des couches application et session

Reference number
ISO 21111-11:2021(E)

Documents
Document Sharing Hub
© ISO 2021

Full document details: BS ISO 21111-11:2021

COPYRIGHT PROTECTED DOCUMENT

© ISO 2021

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

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO 21111-11:2021

Contents

Page

Foreword ... **iv**
Introduction ... **v**
1 Scope ... **1**
2 Normative references ... **1**
3 Terms and definitions ... **1**
4 Symbols and abbreviated terms ... **1**
4.1 Symbols ... 1
4.2 Abbreviated terms ... 2
5 Conventions ... **3**
6 CTP test system set-up and CTC structure ... **3**
6.1 General ... 3
6.2 Test system set-up ... 3
6.3 CTC definition ... 4
6.4 Terminology used in CTCs ... 5
6.5 IUT prerequisites – TCP/IP TestStub ... 6
6.5.1 General ... 6
6.5.2 TCP/IP TestStub methods (service primitives) ... 6
6.5.3 Result codes ... 6
6.6 IUT prerequisites – SOME/IP TestStub ... 7
6.6.1 General ... 7
6.6.2 SOME/IP TestStub methods ... 7
6.6.3 SOME/IP TestStub events and fields. ... 14
6.6.4 ETS service interface description ... 16
7 Application, presentation, and session layers CTCs ... **20**
7.1 AL – SOME/IP ... 20
7.1.1 General ... 20
7.1.2 Referenced specification ... 20
7.1.3 Test system topology – AL – SOME/IP, serialisation, and service discovery ... 20
7.1.4 Test system topology and related CTC configuration ... 21
7.1.5 SOME/IP parameters used in CTCs ... 21
7.1.6 SOME/IP server CTCs ... 23
7.1.7 SOME/IP ETS CTCs ... 128
7.2 SL – Dynamic host configuration protocol version 4 (DHCPv4) client ... 213
7.2.1 General ... 213
7.2.2 Referenced specification ... 213
7.2.3 Test system topology – SL – DHCPv4 client ... 213
7.2.4 Test system topology with two interfaces in the IUT ... 214
7.2.5 Test system topology and related CTC configuration ... 215
7.2.6 DHCPv4 parameters and constants used in CTCs ... 215
7.2.7 DHCPv4 client CTCs ... 217
Bibliography ... **255**

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

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 21111 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.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Introduction

The ISO 21111 series includes in-vehicle Ethernet requirements and test plans that are disseminated in other International Standards and complements them with additional test methods and requirements. The resulting requirement and test plans are structured in different documents following the Open Systems Interconnection (OSI) reference model and grouping the documents that depend on the physical media and bit rate used.

In general, the Ethernet requirements are specified in ISO/IEC/IEEE 8802-3. The ISO 21111 series provides supplemental specifications (e.g. wake-up, I/O functionality), which are required for in-vehicle Ethernet applications. In road vehicles, Ethernet networks are used for different purposes requiring different bit-rates. Currently, the ISO 21111 series specifies the 1-Gbit/s optical and 100-Mbit/s electrical physical layer.

The ISO 21111 series contains requirement specifications and test methods related to the in-vehicle Ethernet. This includes requirement specifications for physical layer entity (e.g. connectors, physical layer implementations) providers, device (e.g. electronic control units, gateway units) suppliers, and system (e.g. network systems) designers. Additionally, there are test methods specified for conformance testing and for interoperability testing.

Safety (electrical safety, protection, fire, etc.) and electromagnetic compatibility (EMC) requirements are out of the scope of the ISO 21111 series.

The structure of the specifications given in the ISO 21111 series complies with the Open Systems Interconnection (OSI) reference model specified in ISO/IEC 7498-1[1] and ISO/IEC 10731[4].

ISO 21111-1 defines the terms which are used in this series of standards and provides an overview of the standards for in-vehicle Ethernet including the complementary relations to ISO/IEC/IEEE 8802-3[2], the document structure, type of physical entities, in-vehicle Ethernet specific functionalities and so on.

ISO 21111-2 specifies the interface between reconciliation sublayer and physical entity including reduced gigabit media independent interface (RGMII), and the common physical entity wake-up and synchronized link sleep functionalities, independent from physical media and bit rate.

ISO 21111-3 specifies supplemental requirements to a physical layer capable of transmitting 1-Gbit/s over plastic optical fibre compliant with ISO/IEC/IEEE 8802-3, with specific application to communications inside road vehicles, and a test plan for physical entity conformance testing.

ISO 21111-4 specifies the optical components requirements and test methods for 1-Gbit/s optical invehicle Ethernet.

ISO 21111-5 specifies, for 1-Gbit/s optical in-vehicle Ethernet, requirements on the physical layer at system level, requirements on the interoperability test set-ups, the interoperability test plan that checks the requirements for the physical layer at system level, requirements on the device-level physical layer conformance test set-ups, and device-level physical layer conformance test plan that checks a set of requirements for the OSI physical layer that are relevant for device vendors.

ISO 21111-6 specifies advanced features of an ISO/IEC/IEEE 8802-3 in-vehicle Ethernet physical layer (often also called transceiver), e.g. for diagnostic purposes for in-vehicle Ethernet physical layers. It specifies advanced physical layer features, wake-up and sleep features, physical layer test suite, physical layer control requirements and conformance test plan, physical sublayers test suite and physical sublayers requirements and conformance test plan.

ISO 21111-7 specifies the implementation for ISO/IEC/IEEE 8802-3, which defines the interface implementation for automotive applications together with requirements on components used to realize this Bus Interface Network (BIN). ISO 21111-7 also defines further testing and system requirements for systems implemented according to the system specification. In addition, ISO 21111-7 defines the channels for tests of transceivers with a test wiring harness that simulates various electrical communication channels.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

ISO 21111-8 specifies the transmission media, the channel performance and the tests for ISO/IEC/IEEE 8802-3 in-vehicle Ethernet.

ISO 21111-9 specifies the data link layer requirements and conformance test plan. It specifies the requirements and test plan for devices and systems with bridge functionality.

ISO 21111-10 specifies the transport to network layer requirements and conformance test plans. It specifies the requirements and conformance test plan for devices and systems that include functionality related with OSI layers from 4 and 3.

This document specifies the application to session layer requirements and conformance test plans. It specifies the requirements and conformance test plan for devices and systems that include functionality related with OSI layers from 7 to 5.

Figure 1 shows the parts of the ISO 21111 series and the document structure.

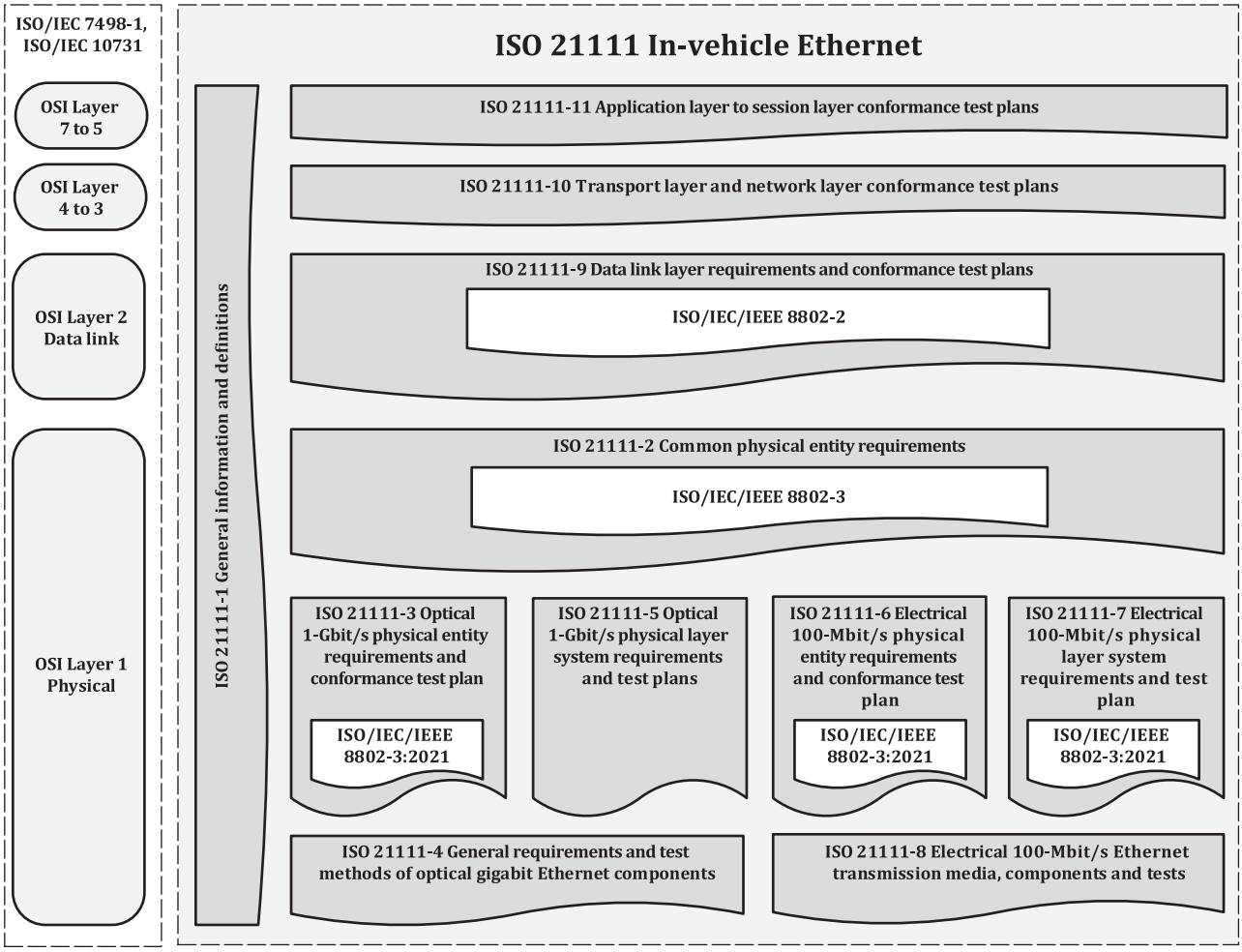


Figure 1 — In-vehicle Ethernet documents reference according to OSI model

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Road vehicles — In-vehicle Ethernet —

Part 11: Application layer to session layer conformance test plans

1 Scope

This document specifies in-vehicle Ethernet application layer, presentation layer, and session layer conformance test plans (CTP) for electronic control units (ECUs). This document is a collection of all conformance test cases which are recommended to be considered for automotive use and should be referred by car manufacturers within their quality control processes.

The document specifies the scalable Service-Oriented MiddlewarE over Internet Protocol (SOME/IP) and Dynamic Host Configuration Protocol (DHCP) version 4 conformance test cases.

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, *Information technology — Open Systems Interconnection — Conformance testing methodology and framework — Part 1: General concepts*

ISO 21111-1, *Road vehicles — In-vehicle Ethernet — Part 1: General information and definitions*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21111-1, ISO/IEC 9646-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
PORT1
port number of the upper tester used for UDP communication

3.2
service interface description
description of the implemented SOME/IP services of a IUT including, e.g. all methods, events, and method parameters

4 Symbols and abbreviated terms

4.1 Symbols

--- empty table cell or feature undefined

© ISO 2021 – All rights reserved

Documents Document Sharing Hub
Full document details: BS ISO 21111-11:2021

V_{BAT}	voltage of the battery
X	feature in table cell selected
$t_{\texttt{Param-Tolerance-Time}}$	tolerance time taken from parameter file
$t_{\texttt{Client-Listen-Time}}$	listen time used by ANVL taken from parameter file
$t_{\texttt{Param-Process-Time}}$	amount of time the test system Host-1 waits for the IUT to process the PDU
$t_{\texttt{Param-Lease-Time}}$	value of DHCP IP address lease time in seconds which is offered to the IUT
$t_{\texttt{Duration-Of-Lease}}$	lease time in seconds
$t_{\texttt{High-Lease-Time}}$	lease time in seconds set to high value

4.2 Abbreviated terms

ACK	acknowlege
ADDR	address
AL	application layer
ANVL	Automated Network Validation Library
ARP	address resolution protocol
BOM	byte order mark
CTC	conformance test case
CTP	conformance test plan
DHCP	Dynamic Host Configuration Protocol
EMC	electromagnetic compatibility
ETS	enhanced testability service
FIN	finish
ICMP	internet control message protocol
IUT	implementation under test
MSL	maximum segment lifetime
MSS	maximum segment size
PCO	point of control and observation
PHY	physical layer
PRS	protocol requirement specification
PSH	push
QOS	qualitiy of service and audio/video bridging
RPC	remote procedure call

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

SL	session layer
SOME/IP	Service-Oriented MiddlewarE over Internet Protocol
SUT	system under test
SYN	synchonise
TCP	transmission control protocol
TIME	time synchronisation
TTL	Time-To-Live
UDP	user datagram protocol

5 Conventions

This document is based on OSI service conventions as specified in ISO/IEC 10731[4].

6 CTP test system set-up and CTC structure

6.1 General

This document specifies a CTP according to the requirements as specified in the ISO/IEC 9646 series. A CTP does not provide qualification of test results but expected responses of the IUT. A CTP is used by a test house to develop a conformance test plan specific for the test system used in their lab environment.

The CTCs specified in this document are organized in such a manner as to simplify the identification of information related to a test and to facilitate in the actual testing process. CTCs are organized into groups, primarily in order to reduce set-up time in the lab environment. The different groups typically also tend to focus on specific aspects of device functionality.

A CTC reference name, for example, "CTC_SOMEIP_ETS_07 – EchoBitfields" is used to organize the CTC name, where the following is included:

— "CTC" which indicates that this is a conformance test case;

— name/subject of CTC;

— supplemental name, for example, ETS, which is enhanced test system;

— CTC number;

— after the hyphen a descriptive name of the CTC follows.

The CTC definitions themselves are intended to provide a high-level description of the purpose, references, prerequisite, steps/procedures, expected responses, remarks, and methodologies pertinent to each test (see 6.3).

6.2 Test system set-up

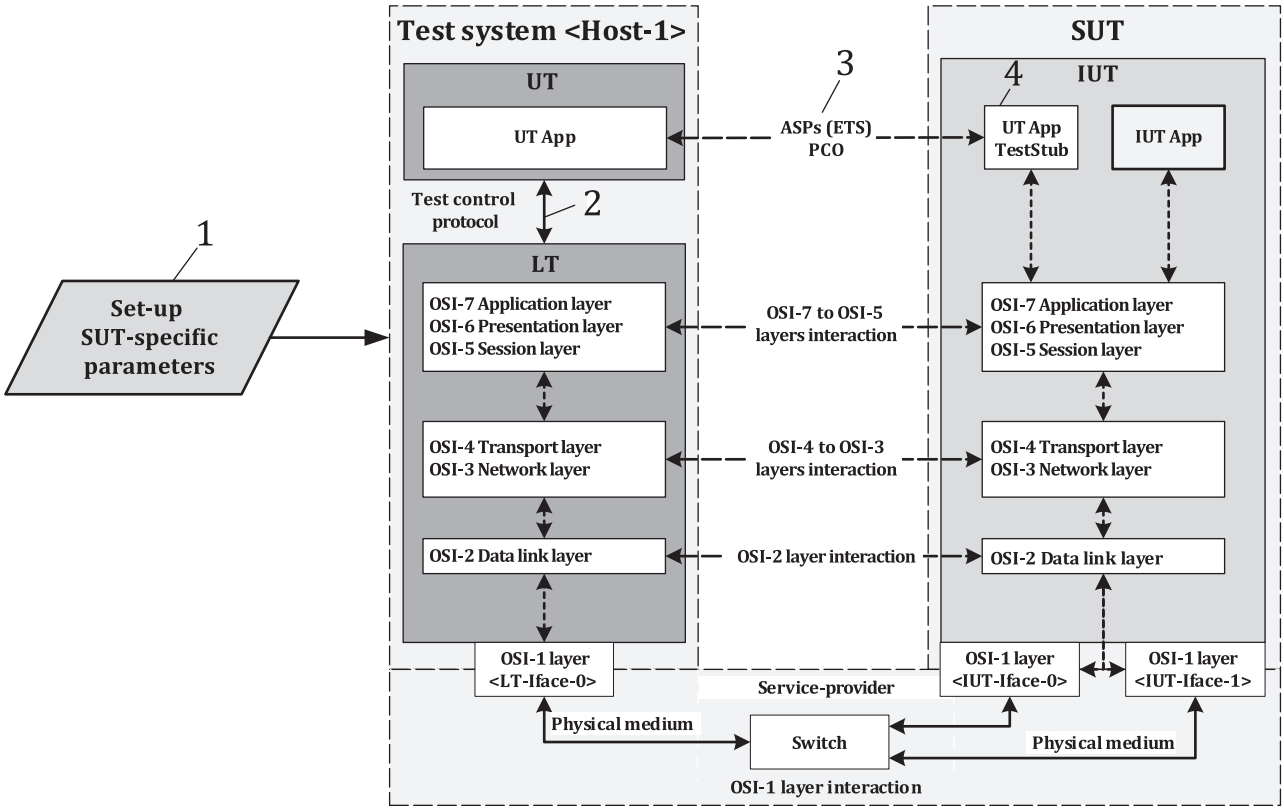
The test system topology follows ISO/IEC 9646-1:1994 and consists of a test set-up which consists of a test system and a system under test (SUT) connected via the physical medium. The test system implements an UT and an LT. The UT uses the test control protocol (Figure 2, key 2) to control the LT. The LT supports the functionality required to test the OSI layer (Figure 2, key 5, key 6, and key 7) of the IUT. The test system uses IUT-specific set-up parameters (Figure 2, key 1) for testing the communication with the IUT.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

The control and measurement functionality is provided by direct logical access to the service interface (dashed line) (Figure 2, key 3) and the associated parameters of the OSI layer. The UT in the IUT (Figure 2, key 4) supports an equivalent part of the abstract service interface (ASPs, PCO) (dashed line) (Figure 2, key 3) and the associated parameters to control and measure the state(s) of the IUT.

The UT conformance test controller in the test system manipulates the service primitive interface parameters in the IUT via the ASPs (ETSs) and PCO of the OSI layers to fulfil the purpose of each CTC.



Key

1 set-up parameters (node's electronic data sheet)

2 test control protocol

3 PCO and ASPs based on ETS

4 UT application with ETS interface

5 OSI layer 7 to 5 protocol

6 OSI layer 4 to 3 protocol

7 OSI layer 2 protocol

Figure 2 — Test system set-up

6.3 CTC definition

CTCs are independent of one another. Each CTC checks the behaviour of the IUT for a particular purpose of this document. CTCs, which require variations of individual parameters, shall be repeated for each value of the parameter. Each CTC is specified according to a common CTC structure as shown in Table 1.

Table 1 — CTC structure

Item	Content
CTC # – Title	CTC_x.y.z-a – CTC structure

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 1 *(continued)*

Item	Content
Purpose	The purpose is a brief statement outlining what the test attempts are to achieve. The test is written at the functional level. It is recommended to begin the description of purpose with "This CTC verifies ...".
Reference	The purpose of reference is to specify source material external to the test suite, including any other references that can be helpful in understanding the test methodology and/or test results. External sources are always referenced by number when mentioned in the test description. Any other references not specified by number are stated with respect to the test suite document itself. EXAMPLE AUTOSAR SOME/IP Example of a Serialization Protocol V1.1.0 R4.1 rev 3: PRS_SOMEIP_00042, PRS_SOMEIP_00099, ...
Prerequisite	The purpose of prerequisites is to specify the test hardware and/or software needed to perform the CTC. This is generally expressed in terms of minimum requirements. In some cases, specific equipment manufacturer/model information may be provided. EXAMPLE The IUT is running and offering the enhanced testability service.
Set-up	The purpose of set-up is to describe the initial configuration of the test environment. Small changes in the configuration should not be included here and are generally covered in the test procedure below. EXAMPLE The test system set-up shall be in accordance with Figure 2.
Step	The test procedure includes the test description, which contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing and may be interspersed with observable results. Each test step shall have a numeric number in ascending order. 1. Configure the IUT as master or as slave. 2. Establish a valid link with the IUT. 3. Monitor the transmissions from the IUT and cause the management to request a PMA reset while simultaneously ceasing transmissions from the test system.
Iteration	The purpose of test iterations is to include test procedure definitions, which are repeated more than once. a) REPEAT step 2 to step 3 with the IUT configured as master, 1 time. b) REPEAT step 2 to step 3 with the IUT configured as slave, 1 time.
Expected response	The purpose of expected response is to describe the expected results to be examined by the test system Host-1 in order to verify that the IUT is operating. When multiple values for an observable are possible, this description provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable. After iteration a): The IUT stops transmitting with tx_mode = SEND_N and starts transmitting with tx_mode = SEND_I for 1 iteration a). The IUT sets link_status = FAIL. After iteration b): The IUT stops transmitting for 1 iteration b). The IUT sets link_status = FAIL.
Remark	The purpose of remarks is to describe known issues with the test procedure, which can affect test results in certain situations. It can also refer the reader to test suite annexes and/or white papers that can provide more detail regarding these issues.

6.4 Terminology used in CTCs

Table 2 specifies the terminology used in CTCs.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 2 — Terminology used in CTCs

Name	Content
Upper tester (UT)	Entity which is responsible for controlling the LT via the test control protocol and the IUT UT enhanced testability services (ETS) via the abstract service primitives (ASPs).
Lower tester (LT)	Entity which is responsible for validating the implementation under test (IUT).
IUT_CONFIGURE	This entry causes the IUT to configure/execute various commands for clearing cache, adding static address, send echo request, etc.
IUT	Implementation under test in the SUT.
CLEANUP	This is a command, which causes the IUT to remove the static entry from its ARP cache.

6.5 IUT prerequisites – TCP/IP TestStub

6.5.1 General

The TCP/IP TestStub defines interfaces required to test TCP/IP communication stack functionality. This is an enabler for generic test tools and conformance test cases (CTCs).

The protocol parts covered by the TCP/IP TestStub include:

— UDP and TCP – socket connection establishment and termination;

— UDP and TCP – message transmission and reception.

The TCP/IP TestStub is specified in Reference [12] (AUTOSAR). This document references a subset of the AUTOSAR specification.

6.5.2 TCP/IP TestStub methods (service primitives)

Table 3 references *AUTOSAR Testability Protocol and Service Primitives, TC Release 1.2.0*, '6.10 Service Primitives'[12] and indicates the service primitives which are applicable.

Table 3 — TCP/IP TestStub methods (service primitives)

Method name (service primitive)	Identifier	General	UDP	TCP
GET_VERSION	1_{16}	optional	---	---
START_TEST	2_{16}	mandatory	---	---
END_TEST	3_{16}	mandatory	---	---
CLOSE_SOCKET	0_{16}	---	mandatory	mandatory
CREATE_AND_BIND	1_{16}	---	mandatory	mandatory
SEND_DATA	2_{16}	---	mandatory	mandatory
RECEIVE_AND_FORWARD	3_{16}	---	mandatory	mandatory
LISTEN_AND_ACCEPT	4_{16}	---	---	mandatory
CONNECT	5_{16}	---	---	mandatory
SHUTDOWN	7_{16}	---	---	optional

6.5.3 Result codes

Due to different stack implementations there is no generic way to retrieve specific result codes. Only generic result codes are accepted. The referenced TestStub methods are changed as follows.

The TestStub methods in Table 4 have no specific result codes, only generic result codes (E_OK / E_NOK) are allowed.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 4 — Result codes

Chapter in Reference [12]"	TestStub method (service primitive)
6.10.4	CLOSE_SOCKET
6.10.5	CREATE_AND_BIND
6.10.6	SEND_DATA
6.10.7	RECEIVE_AND_FORWARD
6.10.8	LISTEN_AND_ACCEPT
6.10.9	CONNECT
6.11.1	SHUTDOWN

6.6 IUT prerequisites – SOME/IP TestStub

6.6.1 General

The SOME/IP TestStub enhanced testability service (ETS)[12] defines interfaces required to test SOME/IP functionality. This is an enabler for generic test tools and conformance test cases (CTCs).

The protocol parts covered by the SOME/IP TestStub includes:

— SOME/IP Stack – Service discovery;

— SOME/IP Stack – Serialisation;

— SOME/IP Stack – Remote procedure call;

— SOME/IP Stack – Publish/Subscribe.

6.6.2 SOME/IP TestStub methods

6.6.2.1 Overview about methods

Table 5 specifies the list of SOME/IP TestStub methods.

Table 5 — List of SOME/IP TestStub methods

SOME/IP TestStub method name	Method-Id	Fire and forget
checkByteOrder()	$1F_{16}$	---
clientServiceActivate()	$2F_{16}$	X
clientServiceDeactivate()	30_{16}	X
clientServiceSubscribe-Event-Group()	32_{16}	X
echoCommonDatatypes()	23_{16}	---
echoENUM()	17_{16}	---
echoFLOAT64()	12_{16}	---
echoINT8()	$0E_{16}$	---
echoStaticUINT8Array()	36_{16}	---
echoUINT8()	08_{16}	---
echoUINT8Array()	09_{16}	---
echoUINT8Array8BitLength()	$3E_{16}$	---

Key

--- empty cell/undefined

X method supported

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 5 *(continued)*

SOME/IP TestStub method name	Method-Id	Fire and forget
echoUINT8Array16BitLength()	$3F_{16}$	---
echoUINT8Array2Dim()	35_{16}	---
echoUINT8ArrayMinSize()	37_{16}	---
echoUINT8E2E()	$0B_{16}$	---
echoUINT8RELIABLE()	$0A_{16}$	---
echoUNION()	19_{16}	---
echoUTF16DYNAMIC()	16_{16}	---
echoUTF16FIXED()	14_{16}	---
echoUTF8DYNAMIC()	15_{16}	---
echoUTF8FIXED()	13_{16}	---
resetInterface()	01_{16}	X
suspendInterface()	02_{16}	X
triggerEventUINT8()	03_{16}	X
triggerEventUINT8Array()	04_{16}	X
triggerEventUINT8E2E()	06_{16}	X
triggerEventUINT8Reliable()	05_{16}	X
triggerEventUINT8Multicast()	$3A_{16}$	X
clientServiceGetLastValueOfEventTCP()	$3B_{16}$	---
clientServiceGetLastValueOfEventUDPUnicast()	$3C_{16}$	---
clientServiceGetLastValueOfEventUDPMulticast()	$3D_{16}$	---
echoBitfields()	41_{16}	---
Key --- empty cell/undefined X method supported		

6.6.2.2 SOME/IP service discovery (SOME/IP-SD) control methods

6.6.2.2.1 SOME/IP-SD – State machine of testing methods

Several methods of the SOME/IP TestStub are used to test SOME/IP-SD. Figure 3 shows the SOME/IP-SD– State machine of testing methods.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

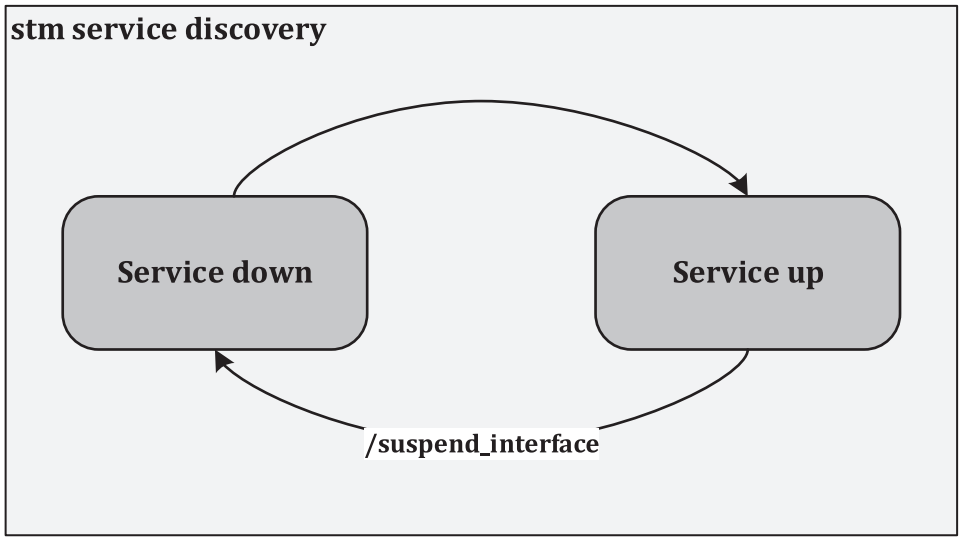


Figure 3 — SOME/IP-SD– State machine of testing methods

6.6.2.2.2 SOME/IP-SD – Method resetInterface()

Table 6 defines the method resetInterface().

Table 6 — Method resetInterface()

Method name	Method content	Method-Id
resetInterface()	This method resets the interface to default values.	01_{16}

Figure 4 shows the method resetInterface().

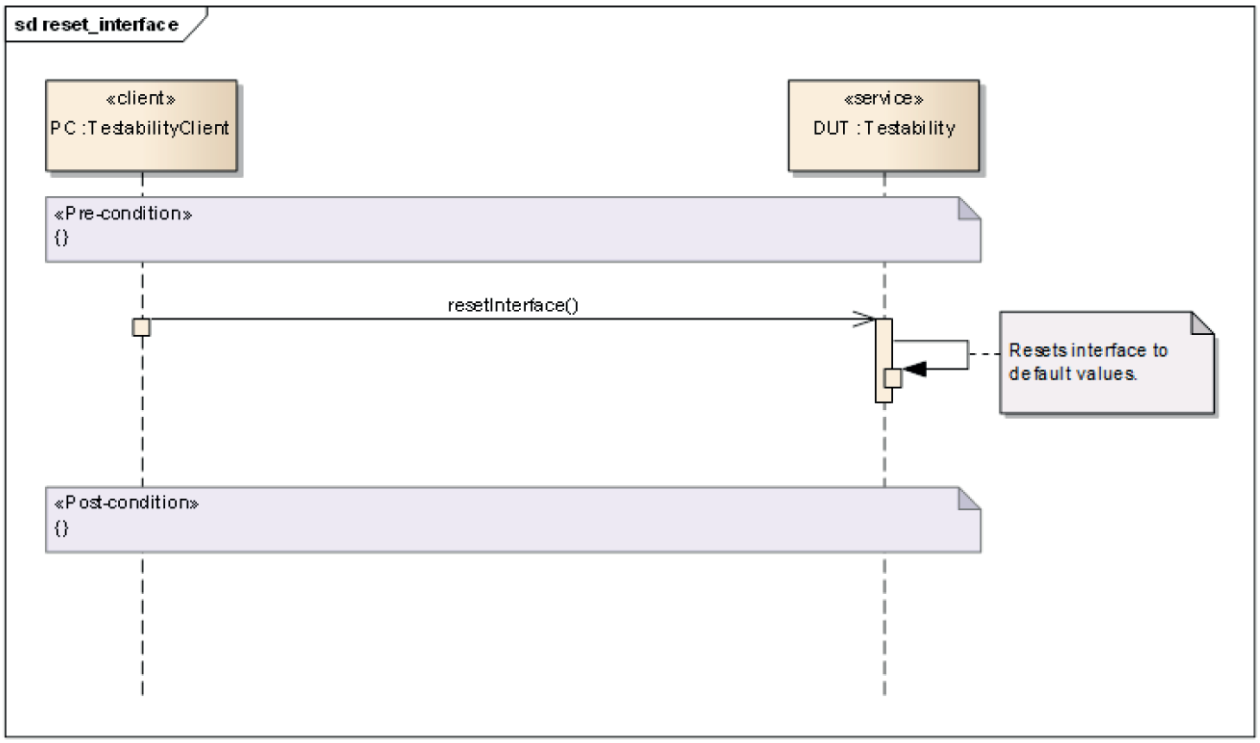


Figure 4 — Method resetInterface()

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

6.6.2.2.3 SOME/IP-SD – Method suspendInterface()

Table 7 defines the method suspendInterface().

Table 7 — Method suspendInterface()

Method name	Method content	Method-Id
suspendInterface()	This method allows to suspend the SOME/IP TestStub; thus, forcing the IUT to stop offering the service and start to reoffer the service after a given time. The input parameters include: — delay – wait time in seconds before the service shall be stopped; — duration – wait time in seconds during between stopping the service and reoffering it again.	02_{16}

Figure 5 shows the method suspendInterface().

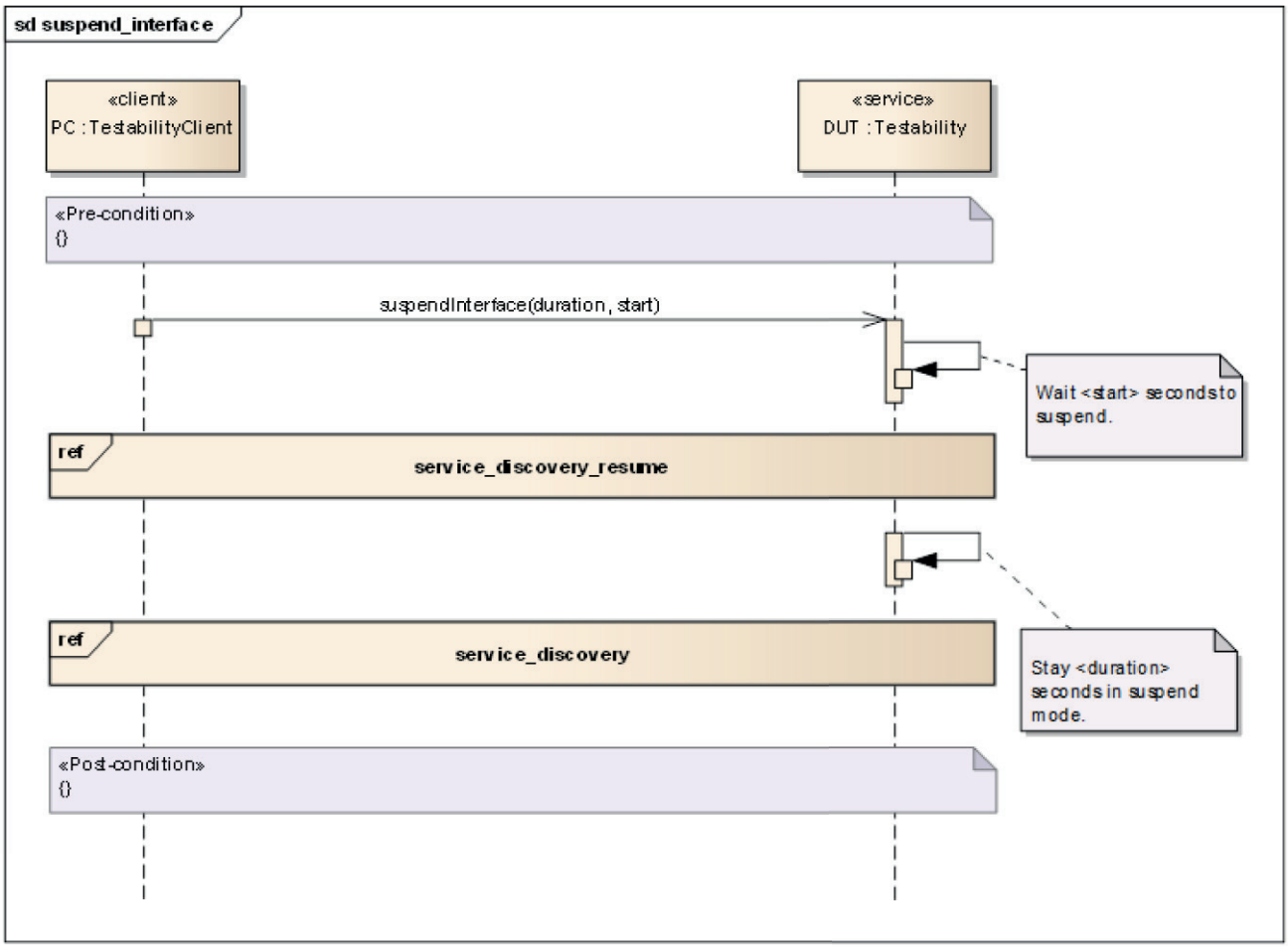


Figure 5 — Method suspendInterface()

6.6.2.3 SOME/IP-SD – Client interaction methods

In order to test the correct function as a client, the SOME/IP TestStub shall support the methods as defined in Table 8.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 8 — Testing client interaction methods

Method name	Method content	Method-Id
clientServiceActivate()	The commands clientServiceActivate() and clientServiceDeactivate() shall instruct the IUT if it shall start its SOME/IP client; thus, they control when the system shall start finding the ETS with the configurable instance identifier (e.g. $00F4_{16}$).	$2F_{16}$
clientServiceDeactivate()		30_{16}
clientServiceSubscribe-Event-Group()	The clientServiceSubscribe-Event-Group shall trigger the subscription behaviour in service discovery.	32_{16}
clientServiceGetLastValueOfEventTCP()	These methods return the last received value of TestEventUINT8Reliable, TestEventUINT8, and TestEventUINT8Multicast respectively.	$3B_{16}$
clientServiceGetLastValueOfEventUDPUnicast()		$3C_{16}$
clientServiceGetLastValueOfEventUDPMulti-cast()		$3D_{16}$

6.6.2.4 SOME/IP serialisation verification methods

6.6.2.4.1 Check byte order method

Table 9 defines the method checkByteOrder().

Table 9 — Method checkByteOrder()

Method name	Method content	Method-Id
checkByteOrder()	This method is used to test the correct handling of the byte order. The input parameters include a uint8 and a big endian uint16, which shall be added and returned as output parameter in UINT32 big endian.	$1F_{16}$

Figure 6 shows the method checkByteOrder().

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

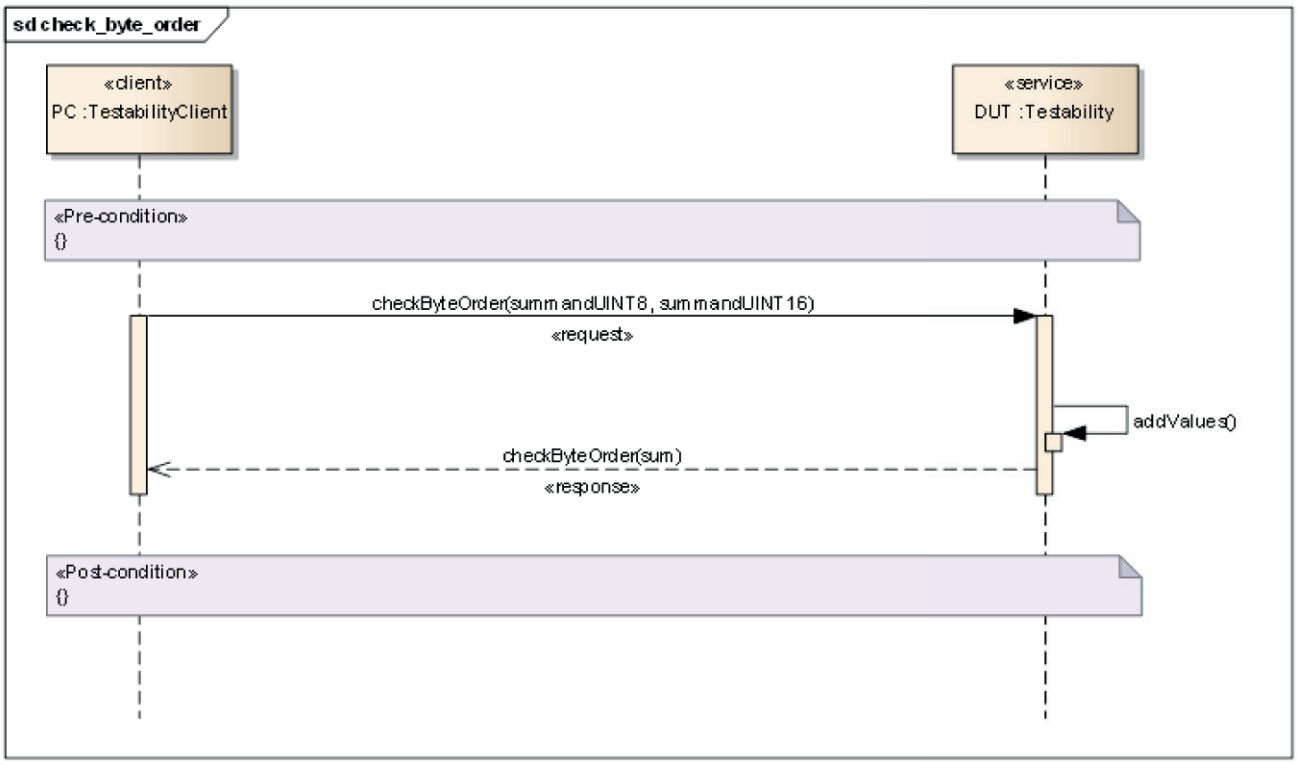


Figure 6 — Method checkByteOrder()

6.6.2.4.2 Echo data types methods

Table 10 defines the methods for echoing data types.

Table 10 — Method echoing data types

Method name	Method content	Method-Id
echoCommonDatatypes()()	This method can be used to test the common data types. The input parameters shall be echoed back in reversed order. The service interface description (6.6.4) describes this in detail.	23_{16}
echoENUM()	The echo-routines echo back the parameters, partly in another order as the parameters coming in. The service interface description (6.6.4) describes this in detail.	17_{16}
echoFLOAT64()		12_{16}
echoINT8()		$0E_{16}$
echoStaticUINT8Array()		36_{16}
echoUINT8()		08_{16}
echoUINT8Array()		09_{16}
echoUINT8Array8 BitLength()		$3E_{16}$
echoUINT8Array16 BitLength()		$3F_{16}$
echoUINT8Array2Dim()		35_{16}
echoUINT8ArrayMinSize()		37_{16}
echoUINT8E2E()		$0B_{16}$
echoUINT8RELIABLE()	This method uses TCP as OSI layer 4 transport protocol.	$0A_{16}$

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 10 *(continued)*

Method name	Method content	Method-Id
echoUNION()	The echo-routines echo back the parameters, partly in another order as the parameters coming in. The service interface description (6.6.4) describes this in detail.	19_{16}
echoUTF16DYNAMIC()		16_{16}
echoUTF16FIXED()		14_{16}
echoUTF8DYNAMIC()		15_{16}
echoUTF8FIXED()		13_{16}
echoBitfields()		41_{16}

Three examples are shown in the diagram below.

Figure 7 shows the methods echoUINT8(), echoUINT8reliable(), echoUINT8E2E().

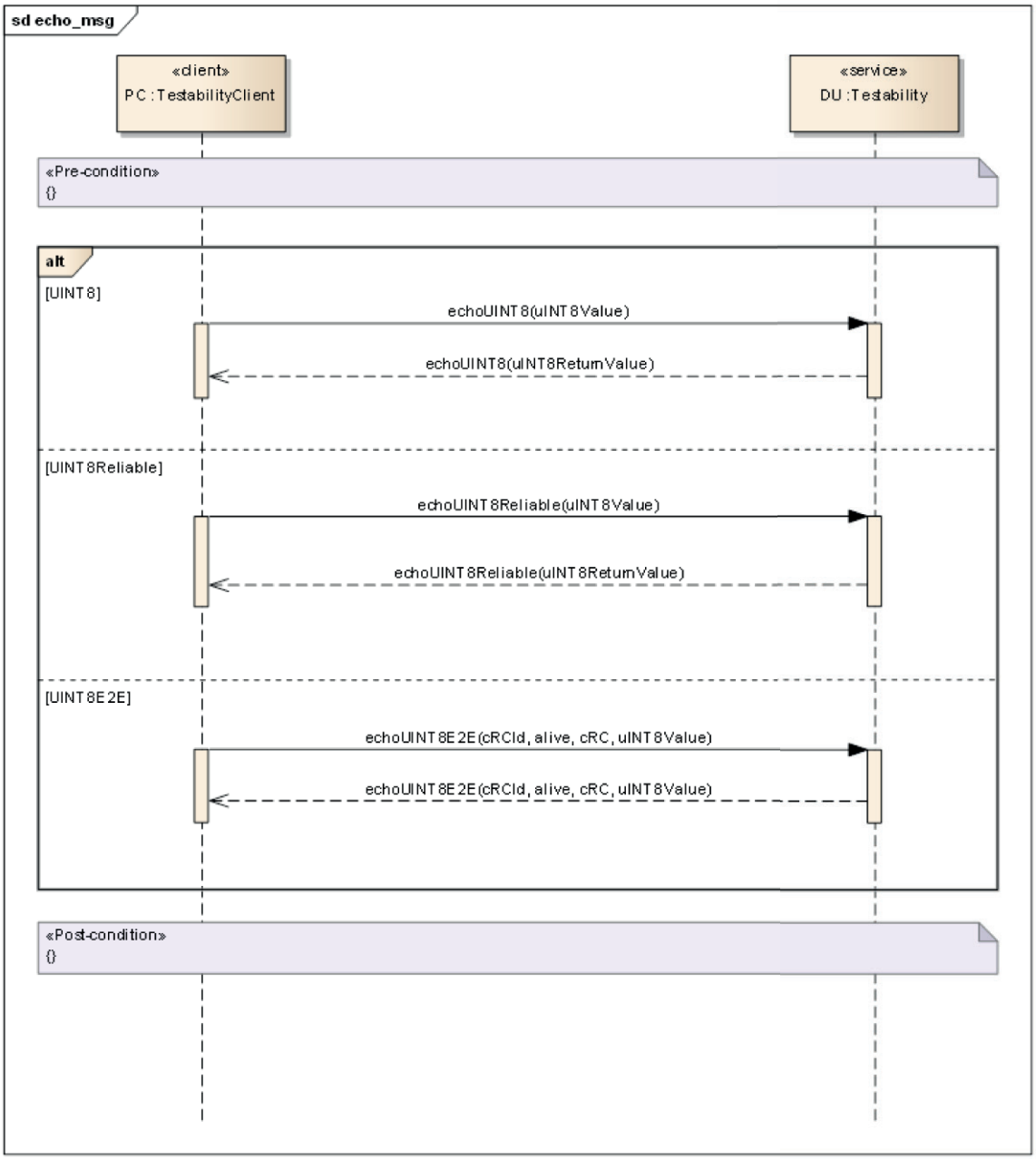


Figure 7 — Methods echoUINT8(), echoUINT8reliable(), echoUINT8E2E()

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

6.6.3 SOME/IP TestStub events and fields.

6.6.3.1 Event and field types

Table 11 specifies the SOME/IP TestStub events and fields.

Table 11 — List of SOME/IP TestStub events and fields

Type	SOME/IP TestStub event/field name	Event-Id/Field-Id	Event-Group-Id		
			0002_{16}	0005_{16}	0006_{16}
Event	TestEventUINT8	8001_{16}	X	X	---
Event	TestEventUINT8Array	8002_{16}	X	X	---
Event	TestEventUINT8E2E	8004_{16}	X	X	---
Event	TestEventUINT8Reliable	8003_{16}	X	---	---
Event	TestEventUINT8Multicast	$800B_{16}$	X	X	X
Field	InterfaceVersion	8005_{16} (notify)	X	X	---
		0025_{16} (getter)	X	X	---
Field	TestFieldUINT8	8006_{16} (notify)	X	X	---
		0026_{16} (getter)	X	X	---
		0027_{16} (setter)	X	X	---
Field	TestFieldUINT8Array	8007_{16} (notify)	X	X	---
		0028_{16} (getter)	X	X	---
		0029_{16} (setter)	X	X	---
Field	TestFieldUINT8Reliable	8008_{16} (notify)	X	---	---
		$002A_{16}$ (getter)	X	---	---
		$002B_{16}$ (setter)	X	---	---

Key

--- empty cell/undefined

X feature in table cell selected

6.6.3.2 Event trigger methods

Table 12 defines the event trigger methods.

Table 12 — Event trigger methods

Method name	Method content	Method-Id
triggerEventUINT8()	After the method triggerEventUINT8() (delay, duration, debounceTime) was invoked, the ETS waits t_{delay} seconds and triggers a periodical event TestEventUINT8(UINT8Value) with the passed duration and debounce time. The service interface description (6.6.4) describes this in detail. The transport protocol of the event message is UDP.	03_{16}
triggerEventUINT8Array()	After the method triggerEventUINT8Array()(delay, duration, debounceTime) was invoked, the ETS waits t_{delay} seconds and triggers a periodical event TestEventUINT8Array(UINT8ArrayValue) with the passed duration and debounce time. The service interface description (6.6.4) describes this in detail. The transport protocol of the event message is UDP.	04_{16}

© ISO 2021 – All rights reserved
Documents Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 12 *(continued)*

Method name	Method content	Method-Id
triggerEventUINT8E2E()	After the method triggerEventUINT8E2E()(delay, duration, debounceTime) was invoked, the ETS waits t_{delay} seconds and triggers a periodical event TestEventUINT8E2E(UINT8E2EValue) with the passed duration and debounce time. The service interface description (6.6.4) describes this in detail. The transport protocol of the event message is UDP.	06_{16}
triggerEventUINT8Reliable()	After the method triggerEventUINT8Reliable()(delay, duration, debounceTime) was invoked, the ETS waits t_{delay} seconds and triggers a periodical event TestEventUINT8Reliable(UINT8Value) with the passed duration and debounce time. The service interface description (6.6.4) describes this in detail. The transport protocol of the event message is TCP.	05_{16}
triggerEventUINT8Multicast()	After the method triggerEventUINT8Multicast()(delay, duration, debounceTime) was invoked, the ETS waits t_{delay} seconds and triggers a periodical event TestEventUINT8Multicast(UINT8Value) with the passed duration and debounce time. The event is sent to a multicast address. The service interface description (6.6.4) describes this in detail. The transport protocol of the event message is UDP.	$3A_{16}$

6.6.3.3 Field methods (getter, setter, notifier)

Table 13 specifies the field methods.

Table 13 — Field methods

Field	Method name	Parameter	Data type	Method-Id
InterfaceVersion	testFieldUINT8_Getter	---	---	25_{16}
	testFieldUINT8_Notifier	testFieldUINT8_Notifier_Arg1 (majorVersion) testFieldUINT8_Notifier_Arg2 (minorVersion)	Uint8 Uint32	8005_{16}
TestFieldUINT8	testFieldUINT8_Getter	---	---	26_{16}
	testFieldUINT8_Setter	testFieldUINT8_Setter_Arg1	Uint8	27_{16}
	testFieldUINT8_Notifier	testFieldUINT8_Notifier_Arg1	Uint8	8006_{16}
TestFieldUINT8Array	testFieldUINT8Array_Getter	---	---	28_{16}
	testFieldUINT8Array_Setter	testFieldUINT8Array_Setter_Arg1 (array len) testFieldUINT8Array_Setter_Arg2 (array)	Uint8 Array of Uint8 elements	29_{16}
	testFieldUINT8Array_Notifier	testFieldUINT8Array_Notifier_Arg1 (array len) testFieldUINT8Array_Notifier_Arg2 (array)	Uint8 Array of Uint8 elements	8007_{16}
TestFieldUINT8Reliable	testFieldUINT8Reliable_Getter	---	---	$2A_{16}$
	testFieldUINT8Reliable_Setter	testFieldUINT8Reliable_Setter_Arg1	Uint8	$2B_{16}$
	testFieldUINT8Reliable_Notifier	testFieldUINT8Reliable_Notifier_Arg1	Uint8	8008_{16}

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

6.6.4 ETS service interface description

The SOME/IP TestStub service interface description reflects the requirements to be supported by the IUT. Therefore, the manufacturer of the IUT provides the service interface description in case an IUT-specific service interface description is not available. Table 14 specifies the default SOME/IP TestStub interface description to be used.

Table 14 — Default SOME/IP TestStub interface description

Method name	Default parameter	Data type	Comment
checkByteOrder_Req	checkByteOrder_ReqArg1 checkByteOrder_ReqArg2	Uint8 Uint16	ResArg1 = sum of ReqArg1+ ReqArg2
checkByteOrder_Res	checkByteOrder_ResArg1	Uint32	
clientServiceActivate_Req	clientServiceActivate_ReqArg1 (delay)	Uint8	Message Type is REQ_NO_RETURN
clientServiceDeactivate_Req	clientServiceDeactivate_ReqArg1 (delay)	Uint8	Message Type is REQ_NO_RETURN
clientServiceSubscribe-Event-Group_Req	clientServiceSubscribe-Event-Group_ReqArg1 (delay) clientServiceSubscribe-Event-Group_ReqArg2 (duration)	Uint32 Uint32	Message Type is REQ_NO_RETURN
echoCommonDatatypes_Req	echoCommonDatatypes_ReqArg1 echoCommonDatatypes_ReqArg2 echoCommonDatatypes_ReqArg3 echoCommonDatatypes_ReqArg4 echoCommonDatatypes_ReqArg5 echoCommonDatatypes_ReqArg6 echoCommonDatatypes_ReqArg7 echoCommonDatatypes_ReqArg8 echoCommonDatatypes_ReqArg9	Boolean uint8 uint16 uint32 int8 int16 int32 float32 float64	echoCommonDatatypes_ReqArg1 = echoCommonDatatypes_ResArg9 echoCommonDatatypes_ReqArg2 = echoCommonDatatypes_ResArg8 etc.
echoCommonDatatypes_Res	echoCommonDatatypes_ResArg1 echoCommonDatatypes_ResArg2 echoCommonDatatypes_ResArg3 echoCommonDatatypes_ResArg4 echoCommonDatatypes_ResArg5 echoCommonDatatypes_ResArg6 echoCommonDatatypes_ResArg7 echoCommonDatatypes_ResArg8 echoCommonDatatypes_ResArg9	float64 float32 int32 int16 int8 uint32 uint16 uint8 Boolean	
echoENUM_Req	echoENUM_ReqArg1	Uint8	echoENUM_ReqArg1 = echoENUM_ResArg1
echoENUM_Res	echoENUM_ResArg1	Uint8	
echoFLOAT64_Req	echoFLOAT64_ReqArg1	Float64	echoFLOAT64_ReqArg1 = echoFLOAT64_ResArg1
echoFLOAT64_Res	echoFLOAT64_ResArg1	Float64	
echoINT8_Req	echoINT8_ReqArg1	Sint8	echoINT8_ReqArg1 = echoINT8_ResArg1
echoINT8_Res	echoINT8_Res_Arg1	Sint8	

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 14 *(continued)*

Method name	Default parameter	Data type	Comment
echoStaticUINT8Array_Req	echoStaticUINT8Array_ReqArg1	Array	Static array of 5 Uint8 elements (without array lenght field) echoStaticUINT8Array_ReqArg1 = echoStaticUINT8Array_ResArg1
echoStaticUINT8Array_Res	echoStaticUINT8Array_ResArg1	Array	
echoUINT8_Req	echoUINT8_ReqArg1	Uint8	echoUINT8_Req = echoUINT8_Res
echoUINT8_Res	echoUINT8_ResArg1	Uint8	
echoUINT8Array_Req	echoUINT8Array_ReqArg1 (array len) echoUINT8Array_ReqArg2 (array)	Uint32 Array	Dynamic array of n Uint8 elements echoUINT8Array_ReqArg1 = echoUINT8Array_ResArg1, echoUINT8Array_ReqArg2 = echoUINT8Array_ResArg2
echoUINT8Array_Res	echoUINT8Array_ResArg1 (array len) echoUINT8Array_ResArg2 (array)	Uint32 Array	
echoUINT8Array8BitLength_Req	echoUINT8Array8BitLength_ReqArg1 (array len) echoUINT8Array8BitLength_ReqArg2 (array)	Uint8 Array	Dynamic array of n Uint8 elements echoUINT8Array8BitLength_ReqArg1 = echoUINT8Array8BitLength_ResArg1, echoUINT8Array8BitLength_ReqArg2 = echoUINT8Array8BitLength_ResArg2
echoUINT8Array8BitLength_Res	echoUINT8Array8BitLength_ResArg1 (array len) echoUINT8Array8BitLength_ResArg2 (array)	Uint8 Array	
echoUINT8Array16BitLength_Req	echoUINT8Array16BitLength_ReqArg1 (array len) echoUINT8Array16BitLength_ReqArg2 (array)	Uint16 Array	Dynamic array of n Uint8 elements echoUINT8Array16BitLength_ReqArg1 = echoUINT8Array16BitLength_ResArg1, echoUINT8Array16BitLength_ReqArg2 = echoUINT8Array16BitLength_ResArg2
echoUINT8Array16BitLength_Res	echoUINT8Array16BitLength_ResArg1 (array len) echoUINT8Array16BitLength_ResArg2 (array)	Uint16 Array	
echoUINT8Array2Dim_Req	echoUINT8Array2Dim_ReqArg1 (array len) echoUINT8Array2Dim_ReqArg2 (array of arrays)	Uint32 Array of arrays	Uint8array: Len (Uint32) Array of n Uint8 elements echoUINT8Array2Dim_ReqArg1 = echoUINT8Array2Dim_ResArg1 echoUINT8Array2Dim_ReqArg2 = echoUINT8Array2Dim_ ResArg2
echoUINT8Array2Dim_Res	echoUINT8Array2Dim_ResArg1 (array len) echoUINT8Array2Dim_ResArg2 (array of arrays)	Uint32 Array of Uint8Arrays	

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 14 *(continued)*

Method name	Default parameter	Data type	Comment
echoUINT8ArrayMinSize_Req	echoUINT8ArrayMinSize_ReqArg1 (array len) echoUINT8ArrayMinSize_ReqArg1 (array)	Uint32 Array	Dynamic array of n Uint8 elements with n = 3 .. 5 echoUINT8ArrayMinSize_ReqArg1 = echoUINT8ArrayMinSize_ResArg1, echoUINT8ArrayMinSize_ReqArg2 = echoUINT8ArrayMinSize_ResArg2
echoUINT8ArrayMinSize_Res	echoUINT8ArrayMinSize_ResArg1 (array len) echoUINT8ArrayMinSize_ResArg1 (array)	Uint32 Array	
echoUINT8E2E_Req	echoUINT8E2E_ReqArg1 (len) echoUINT8E2E_ReqArg2 (counter) echoUINT8E2E_ReqArg3 (dataID) echoUINT8E2E_ReqArg4 (crc) echoUINT8E2E_ReqArg5 (value)	Uint32 Uint32 Uint32 Uint32 Uint8	echoUINT8E2E_ReqArg1 = echoUINT8E2E_ResArg1, etc.
echoUINT8E2E_Res	echoUINT8E2E_ResArg1 (len) echoUINT8E2E_ResArg2 (counter) echoUINT8E2E_ResArg3 (dataID) echoUINT8E2E_ResArg4 (crc) echoUINT8E2E_ResArg5 (value)	Uint32 Uint32 Uint32 Uint32 Uint8	
echoUINT8RELIABLE_Req	echoUINT8RELIABLE_ReqArg1	Uint8	echoUINT8RELIABLE_ReqArg1 = echoUINT8RELIABLE_ResArg1 The used Layer 4 protocol is TCP
echoUINT8RELIABLE_Res	echoUINT8RELIABLE_ResArg1	Uint8	
echoUNION_Req	echoUNION_ReqArg1 (len) echoUNION_ReqArg2 (type) echoUNION_ReqArg3 (value)	Uint32 Uint32 See comment	Type 1 = Boolean Type 2 = uint8 Type 3 = uint16 Type 4 = uint32 Type 5 = sint8 Type 6 = sint16 Type 7 = sint32 Type 8 = float32
echoUNION_Res	echoUNION_ResArg1 (len) echoUNION_ResArg2 (type) echoUNION_ResArg3 (value)	Uint32 Uint32 See comment	
echoUTF16DYNAMIC_Req	echoUTF16DYNAMIC_ReqArg1 (len) echoUTF16DYNAMIC_ReqArg2 (string)	Unit32 UTF16	echoUTF16DYNAMIC_ReqArg1 = echoUTF16DYNAMIC_ResArg1, etc.
echoUTF16DYNAMIC_Res	echoUTF16DYNAMIC_ResArg1 (len) echoUTF16DYNAMIC_ResArg2 (string)	Unit32 UTF16	
echoUTF16FIXED_Req	echoUTF16FIXED_ReqArg1 (string)	UTF16Fixed	UTF16Fixed = 64 bytes of string including BOM echoUTF16FIXED_ReqArg1 = echoUTF16FIXED_ResArg1
echoUTF16FIXED_Res	echoUTF16FIXED_ResArg1 (string)	UTF16Fixed	

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 14 *(continued)*

Method name	Default parameter	Data type	Comment
echoUTF8DYNAMIC_Req	echoUTF8DYNAMIC_ReqArg1 (len) echoUTF8DYNAMIC_ReqArg2 (string)	Uint32 UTF8	echoUTF8DYNAMIC_ResArg1 = echoUTF8DYNAMIC_ReqArg1, etc
echoUTF8DYNAMIC_Res	echoUTF8DYNAMIC_ResArg1 (len) echoUTF8DYNAMIC_ResArg2 (string)	Uint32 UTF8	
echoUTF8FIXED_Req	echoUTF8FIXED_ReqArg1 (string)	UTF8Fixed	UTF8Fixed = 64 bytes of string including BOM echoUTF8FIXED_ReqArg1 = echoUTF8FIXED_ResArg1
echoUTF8FIXED_Res	echoUTF8FIXED_ResArg1 (string)	UTF8Fixed	
resetInterface_Req	N/A	N/A	Message Type is REQ_NO_RETURN without parameters
suspendInterface_Req	suspendInterface_ReqArg1 (delay) suspendInterface_ReqArg2 (duration)	Uint32 Uint32	Message Type is REQ_NO_RETURN
triggerEventUINT8_Req	triggerEventUINT8_ReqArg1 (delay) triggerEventUINT8_ReqArg2 (duration) triggerEventUINT8_ReqArg3 (debounceTime)	Uint32 Uint32 Uint32	Message Type is REQ_NO_RETURN
triggerEventUINT8Array_Req	triggerEventUINT8Array_ReqArg1 (delay) triggerEventUINT8Array_ReqArg2 (duration) triggerEventUINT8Array_ReqArg3 (debounceTime)	Uint32 Uint32 Uint32	Message Type is REQ_NO_RETURN
triggerEventUINT8E2E_Req	triggerEventUINT8E2E_ReqArg1 (delay) triggerEventUINT8E2E_ReqArg2 (duration) triggerEventUINT8E2E_ReqArg3 (debounceTime)	Uint32 Uint32 Uint32	Message Type is REQ_NO_RETURN
triggerEventUINT8Reliable_Req	triggerEventUINT8Reliable_ReqArg1 (delay) triggerEventUINT8Reliable_ReqArg2 (duration) triggerEventUINT8Reliable_ReqArg3 (debounceTime)	Uint32 Uint32 Uint32	Message Type is REQ_NO_RETURN The used Layer 4 protocol is TCP
triggerEventUINT8Multicast_Req	triggerEventUINT8Multicast_ReqArg1 (delay) triggerEventUINT8Multicast_ReqArg2 (duration) triggerEventUINT8Multicast_ReqArg3 (debounceTime)	Uint32 Uint32 Uint32	Message Type is REQ_NO_RETURN
clientServiceGetLastValueOfE ventTCP_Req	N/A	N/A	---
clientServiceGetLastValueOfE ventTCP_Res	clientServiceGetLastValueOfEventTCP_Res Arg1	Uint8	---
clientServiceGetLastValueOfE ventUDPUnicast_Req	N/A	N/A	---

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 14 *(continued)*

Method name	Default parameter	Data type	Comment
clientServiceGetLastValue-OfE ventUDPUnicast_Res	clientServiceGetLastValueOfEventUD-PUni cast_ResArg1	Uint8	---
clientServiceGetLastValue-OfE ventUDPMulticast_Req	N/A	N/A	---
clientServiceGetLastValue-OfE ventUDPMulticast_Res	clientServiceGetLastValueOfEventUDP-Mul ticast_ResArg1	Uint8	---
echoBitfields_Req	echoBitfields_ReqArg1 echoBitfields_ReqArg2 echoBitfields_ReqArg3	uint8 uint16 uint32	echoBitfields_ResArg1 = reversed bit order of echoBitfields_ReqArg1 echoBitfields_ResArg2 = reversed bit order of echoBitfields_ReqArg2 echoBitfields_ResArg3 = reversed bit order of echoBitfields_ReqArg3
echoBitfields_Res	echoBitfields_ResArg1 echoBitfields_ResArg2 echoBitfields_ResArg3	uint8 uint16 uint32	

7 Application, presentation, and session layers CTCs

7.1 AL – SOME/IP

7.1.1 General

The scope of this subclause is to specify CTCs for the scalable Service-Oriented MiddlewarE over IP protocol (SOME/IP).

7.1.2 Referenced specification

All CTCs in 7.1 are in accordance with these RFCs:

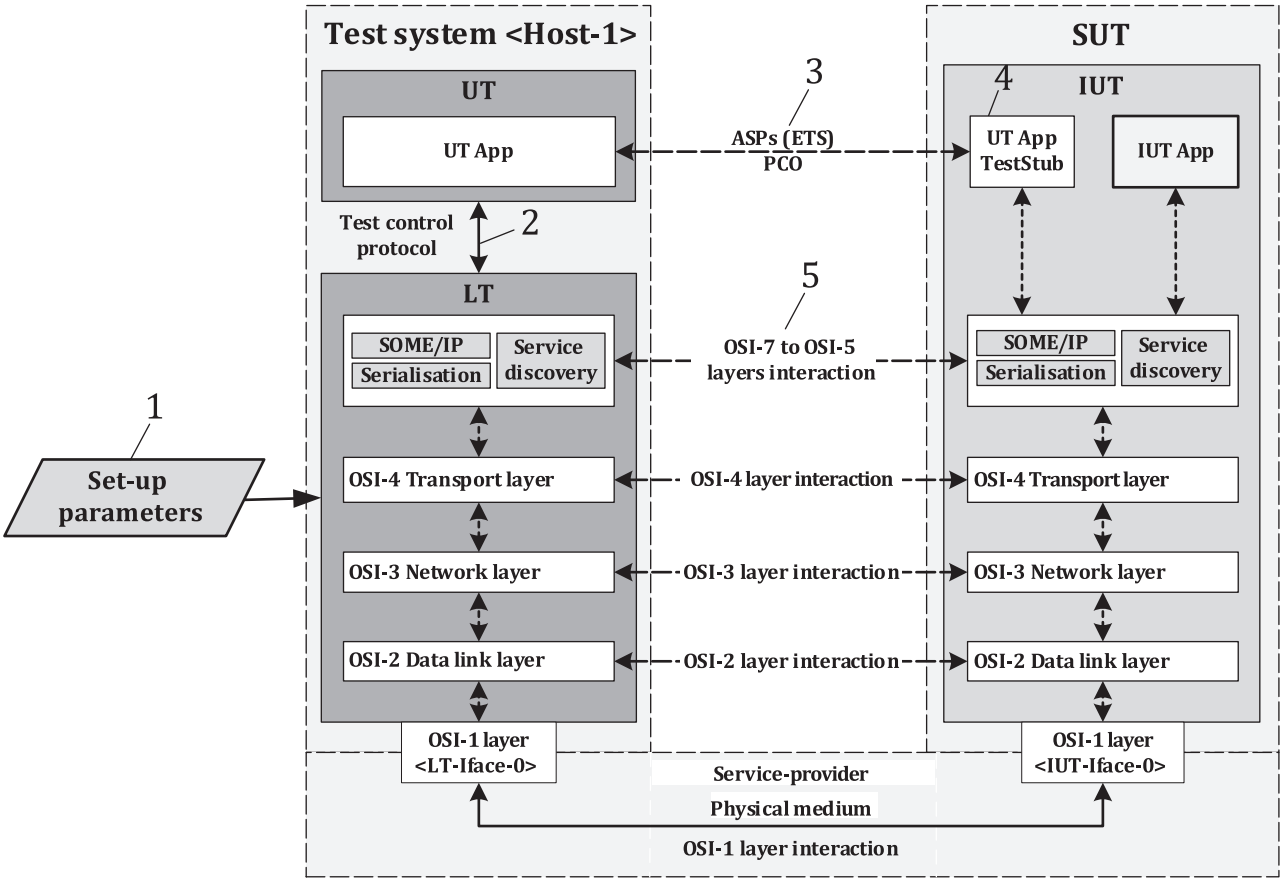
— Reference [10];

— Reference [11].

7.1.3 Test system topology – AL – SOME/IP, serialisation, and service discovery

Figure 8 shows the test system topology – AL – SOME/IP, serialisation, and service discovery.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021



Key

1 IUT-specific set-up parameters (electronic data sheet)

2 test control protocol

3 PCO and ASPs based on ETS

4 UT TestStub application

5 OSI layer 7 SOME/IP, serialisation, and service discovery interaction

Figure 8 — Test system topology – AL – SOME/IP, serialisation, and service discovery

7.1.4 Test system topology and related CTC configuration

The test system Host-1 configuration is required for all CTCs pertaining to SOME/IP tests:

— with correct IUT IP address for the IUT interface connected to the test system interface; and

— with only one IP interface in the IUT.

7.1.5 SOME/IP parameters used in CTCs

7.1.5.1 User defined configuration parameters in SOME/IP for IUT

Table 15 specifies the user defined configuration parameters in SOME/IP for IUT.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 15 — User defined configuration parameters in SOME/IP for IUT

Parameter used in CTCs	Content
Listen-Time-Setting	This is the maximum time interval for which ANVL waits for a packet for cases when a certain event is triggered on the IUT either by some protocol timer or using some external mechanism (script). Default: $0A_{16}$
Millisec-Toler-ance-Time-Setting	This is the tolerance time associated with some event in ms. When waiting or listening this number is added with the actual wait-time or listen-time. This is currently used if the IUT's responses are in ms interval. Default: 64_{16}
Process-Time-Setting	This is the time the IUT appoximately takes to process a packet. ANVL uses this process time added to listen time for some time-critical situations while listening for the IUT responses. Default: 02_{16}
Tolerance-Time-Setting	This is the tolerance time associated with an event. When waiting or listening, this number is added to the actual wait-time or listen-time. Default: 01_{16}

7.1.5.2 User defined configuration parameters in SOME/IP for UT and LT

Table 16 specifies the user defined configuration parameters in SOME/IP for UT and LT.

Table 16 — User defined configuration parameters in SOME/IP for UT and LT

Parameter used in CTCs	Content
IUT-Iface-n	n-th interface of IUT
Service-Id-1	service identifier of service 1
Service-Id-2	service identifier of service 2
Service-Id-1-Instance-Id	instance identifier of service 1
Service-Id-1-Maj-Ver	major version supported by service 1
Service-Id-1-Intf-Maj-Ver	interface version of service 1
Service-Id-2-Intf-Maj-Ver	interface version of service 2
Service-Id-1-Time-To-Live	Time-To-Live value of service 1
Service-Id-1-Minor-Ver	minor version supported by service 1
Service-Id-1-Inst-1	first instance identifier of service 1
Evt-Grp-Id-2	event group identifier 2
Evt-Grp-Id-1-Service-1	event group identifier 1 of service 1
Service-Id-1-UDP-Port	UDP port number where service 1 runs
SOMEIP-Multicast-IP-Addr	multicast address used by SOME/IP-SD
Client-1-Curr-Request-Id	current request identifier of client 1
Evt-Id-1-Evt-Grp-Id-1	field event identifier 1 of event group identifier 1 that shall be sent out initially at initial subscription
Evt-Id-1-Evt-Grp-Id-2	field event identifier 1 of event group identifier 2 that shall be sent out initially at initial subscription
Evt-Id-2-Evt-Grp-Id-1	event identifier 2 of event group identifier 1
Event-Id-2-Evt-Grp-Id-2	event identifier 2 of event group identifier 2
Evt-Id-2-Evt-Grp-Id-1-Cyclic-Time	cycle time of event identifier-2 of event group 1 (in this CTC the event identifier 1 is assumed to be a cyclic event)
Evt-Grp-Id-1-Service-1-Multicast-Addr	multicast address of event group identifier 1 of service 1 where the event and notifications shall be multicast

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 16 *(continued)*

Parameter used in CTCs	Content
Unknown-Service-Id	invalid service identifier
Unknown-Evt-Grp-Id	invalid event group identifier
Unknown-Method-Id-Service-1	invalid method identifier of service identifier 1
Unknown-Iface-Ver-Maj-Service-1	invalid interface version of service identifier 1
Service-Id-Rep-Base-Intval	repetition base interval of service 1
Service-Id-1-Total-Rep-Intval	total repetition period of service 1 calculated using Formula (1)
Service-Id-1-Initial-Wait-Time	initial wait time for service 1
Service-Id-1-Cycle-Intval	offer cycle interval of service 1
Method-Id-1-Service-Iface-1	first InOut type method listed in service interface description for service 1
Method-Id-1-Service-Iface-2	first InOut type method listed in service interface description for service 2
Service-Id-2-Transport-Port	transport port of service identifier 2 (UDP/TCP)
Service-Id-2-Initial-Wait-Time	time taken by the IUT before sending offer service message
Server-1-IP-Addr	IP address of SOME/IP-SD server 1
Client-1-IP-Addr	ANVL Client-1 IP address
Client-1-UDP-Port	ANVL Client-1 UDP port
Boolean-Foreground	boolean value set to false to indicate script running in foreground
Boolean-Background	boolean value set to true to indicate script running in background
$t_{\texttt{Param-Tolerance-Time}}$	tolerance time taken from parameter file
$t_{\texttt{Client-Listen-Time}}$	listen time used by ANVL taken from parameter file
Busy-Wait	indication of ANVL listening to incoming packets while waiting
Lazy-Wait	indication of ANVL not listening to incoming packets while waiting
$t_{\texttt{Listen-Time}}$	maximum time interval for which ANVL waits for a packet for cases when a certain event has been triggered on the IUT either by some protocol timer or using some external mechanism (script) [Default: 10 s]
$t_{\texttt{Tolernace-Time}}$	This is the tolerance time associated with an event. When waiting or listening then this number shall be added with the actual wait-time or listen-time. [Default: 1 s]
Source-Port	UDP source port used by the LT, e.g. 12345

$$d_{\text{Repetition_phase_duration}} = R_{\text{base_intval}} \times \left(\left(2^{R_{\max}}\right) - 1\right) \tag{1}$$

where

$d_{\text{Repetition_phase_duration}}$ is the total repetition period of, for example, service;

$R_{\text{base_intval}}$ is the repetition base interval;

$R_{\max}$ is the maximum repetition value.

7.1.6 SOME/IP server CTCs

7.1.6.1 CTC_SOMEIPSRV_FORMAT – Message format

Table 17 specifies the CTC_SOMEIPSRV_FORMAT_01 – Client-Id test case.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 17 — CTC_SOMEIPSRV_FORMAT_01 – Client-Id

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_01 – Client-Id
Purpose	This CTC verifies that the CLIENT-ID is set statically to 0000_{16}.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00033 '
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends a SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the Client-Id set to 0000_{16}.
Remark	---

Table 18 specifies the CTC_SOMEIPSRV_FORMAT_02 – SessionID test case.

Table 18 — CTC_SOMEIPSRV_FORMAT_02 – SessionID

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_02 – SessionID
Purpose	This CTC verifies that after initialisation of the service discovery module, the SessionID for messages sent by the local ECU is 0001_{16}.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00034'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to (Service-Id-1-Initial-Wait-Time + $t_{\text{Client-Listen-Time}}$) second on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 18 *(continued)*

Item	Content
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the SessionID which is set to 0001_{16}.
Remark	---

Table 19 specifies the CTC_SOMEIPSRV_FORMAT_03 – Protocol version test case.

Table 19 — CTC_SOMEIPSRV_FORMAT_03 – Protocol version

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_03 – Protocol version
Purpose	This CTC verifies that the value for the protocol version field is statically set to 01_{16}.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00141'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the protocol version which is set to 01_{16}.
Remark	---

Table 20 specifies the CTC_SOMEIPSRV_FORMAT_04 – Interface version test case.

Table 20 — CTC_SOMEIPSRV_FORMAT_04 – Interface version

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_04 – Interface version
Purpose	This CTC verifies that the value for the interface version field is statically set to 01_{16}.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00143'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 20 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the interface version which is set to 01_{16}.
Remark	---

Table 21 specifies the CTC_SOMEIPSRV_FORMAT_05 – Message type test case.

Table 21 — CTC_SOMEIPSRV_FORMAT_05 – Message type

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_05 – Message type
Purpose	This CTC verifies that the value for the message type field is statically set to 02_{16}.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00145 '
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the message type field which is set to 02_{16}.
Remark	---

Table 22 specifies the CTC_SOMEIPSRV_FORMAT_06 – Return code test case.

Table 22 — CTC_SOMEIPSRV_FORMAT_06 – Return code

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_06 – Return code

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 22 *(continued)*

Item	Content
Purpose	This CTC verifies that the return code field is statically set to 00_{16}. This test case is for SD message.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00147'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the return code which is set to 00_{16}.
Remark	---

Table 23 specifies the CTC_SOMEIPSRV_FORMAT_07 – Reboot flag test case.

Table 23 — CTC_SOMEIPSRV_FORMAT_07 – Reboot flag

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_07 – Reboot flag
Purpose	This CTC verifies that the reboot flag is set to 1_2 for all messages after reboot until the Session-ID of the request ID field wraps and thus starts with 0001_{16} again. After that the reboot flag is set to 0_2.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00151'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 23 *(continued)*

Item	Content
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the reboot flag bit which is set to 1.
Remark	---

Table 24 specifies the CTC_SOMEIPSRV_FORMAT_08 – Unicast flag test case.

Table 24 — CTC_SOMEIPSRV_FORMAT_08 – Unicast flag

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_08 – Unicast flag
Purpose	This CTC verifies that the unicast flag of the flag field is set to the unicast flag and is set to 1_2, meaning: this ECU supports receiving unicast messages.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00153'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains the unicast flag bit which is set to 1.
Remark	---

Table 25 specifies the CTC_SOMEIPSRV_FORMAT_09 – Undefined bits in the flag field test case.

Table 25 — CTC_SOMEIPSRV_FORMAT_09 – Undefined bits in the flag field

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_09 – Undefined bits in the flag field
Purpose	This CTC verifies that the undefined bits within the flag field are statically set to 0_2.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00154'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 25 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains six flag bits other than reboot and unicast bits which are set to zero.
Remark	---

Table 26 specifies the CTC_SOMEIPSRV_FORMAT_10 – Reserved bits test case.

Table 26 — CTC_SOMEIPSRV_FORMAT_10 – Reserved bits

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_10 – Reserved bits
Purpose	This CTC verifies that all bits of the reserved field are statically set to 0_2.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00156'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains reserved field set to $00\ 0000_{16}$.
Remark	---

Table 27 specifies the CTC_SOMEIPSRV_FORMAT_11 – Length of the type 1 entry test case.

Table 27 — CTC_SOMEIPSRV_FORMAT_11 – Length of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_11 – Length of the type 1 entry
Purpose	This CTC verifies that the length of the type 1 entry is 16 bytes.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 27 *(continued)*

Item	Content
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3*[10], ' 7.3 SWS_SD_00160'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0. 3. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 2: The IUT sends the SOME/IP notification message. After step 2: The LT verifies that the received SOME/IP notification message contains: — entry type is set to 01_{16}; — entry length field is set to (Number-Of-Entries × 16).
Remark	---

Table 28 specifies the CTC_SOMEIPSRV_FORMAT_12 – Index first option run of the type 1 entry test case.

Table 28 — CTC_SOMEIPSRV_FORMAT_12 – Index first option run of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_12 – Index first option run of the type 1 entry
Purpose	This CTC verifies that the "Index First Option Run" field of the type 1 entry format layout carries the index of the first option of the first option run of this entry in the option array.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00164'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 28 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the index 1st option in entry which is set to 0.
Remark	---

Table 29 specifies the CTC_SOMEIPSRV_FORMAT_13 – Number of option 1 of the type 1 entry test case.

Table 29 — CTC_SOMEIPSRV_FORMAT_13 – Number of option 1 of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_13 – Mumber of option 1 of the type 1 entry
Purpose	This CTC verifies that the "Number of Option 1" of the type 1 entry format layout carries the number of options the first option run uses.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00168'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 29 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the Number of Option 1 in entry array which is greater than or equal to 1.
Remark	---

Table 30 specifies the CTC_SOMEIPSRV_FORMAT_14 – Service identifier field of the type 1 entry test case.

Table 30 — CTC_SOMEIPSRV_FORMAT_14 – Service identifier field of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_14 – Service identifier field of the type 1 entry
Purpose	This CTC verifies that the service identifier field of the type 1 entry format layout carries the service identifier of the service, statically configured using the parameter SdServerservice identifier and SdClientservice identifier, depending on being a server or client entry.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00173'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 30 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the following information the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains service identifier field which is set to Service-Id-1.
Remark	---

Table 31 specifies the CTC_SOMEIPSRV_FORMAT_15 – InstanceID field of the type 1 entry test case.

Table 31 — CTC_SOMEIPSRV_FORMAT_15 – InstanceID field of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_15 – InstanceID field of the type 1 entry
Purpose	This CTC verifies that the instance identifier field of the type 1 entry format layout carries the instance identifier of the service, statically configured using the parameter SdServerServiceInstanceID and SdClientServiceInstanceID, depending on being a server or client entry.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00175'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 31 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the InstanceID field in entry which is set to Service-Id-1-Instance-Id.
Remark	---

Table 32 specifies the CTC_SOMEIPSRV_FORMAT_16 – Major version field of the type 1 entry test case.

Table 32 — CTC_SOMEIPSRV_FORMAT_16 – Major version field of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_16 – Major version field of the type 1 entry
Purpose	This CTC verifies that the major version field of the type 1 entry format layout carries the SdServerServiceMajorVersion and SdClientServiceMajorVersion, depending on being a server or client entry.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00178'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 32 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the major version field in the entry which is set to Service-Id-1-Maj-Ver.
Remark	---

Table 33 specifies the CTC_SOMEIPSRV_FORMAT_17 – Time-To-Live field of the type 1 entry test case.

Table 33 — CTC_SOMEIPSRV_FORMAT_17 – Time-To-Live field of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_17 – Time-To-Live field of the type 1 entry
Purpose	This CTC verifies that the Time-To-Live field of the type 1 entry format layout defines the lifetime of the entry in seconds configured using the parameter SdServerTimerTTL and SdClientTimerTTL, except for Stop-entries, which have a Time-To-Live of 0.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00180'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 33 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the Time-To-Live field in the entry which is set to Service-Id-1-Time-To-Live.
Remark	---

Table 34 specifies the CTC_SOMEIPSRV_FORMAT_18 – Minor version field of the type 1 entry test case.

Table 34 — CTC_SOMEIPSRV_FORMAT_18 – Minor version field of the type 1 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_18 – Minor version field of the type 1 entry
Purpose	This CTC verifies that the minor version field of the type 1 entry format layout carries the SdServerServiceMinorVersion and SdClientServiceMinorVersion.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00182'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 34 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 3: The LT verifies that the received SOME/IP notification message contains the minor version field in the entry which is set to Service-Id-1-Minor-Ver.
Remark	---

Table 35 specifies the CTC_SOMEIPSRV_FORMAT_19 – Type field of the type 2 entry test case.

Table 35 — CTC_SOMEIPSRV_FORMAT_19 – Type field of the type 2 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_19 – Type field of the type 2 entry
Purpose	This CTC verifies that the type field of the type 2 entry format layout carries the value 07_{16} to encode Subscribe-Event-Group-ACK.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00385'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 35 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry array set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry array set to Service-Id-1. 4. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 06_{16} (SOMEIP_ENTRY_SUBSCRIBE); — service identifier in entry array set to Service-Id-1; — Event-Group-Id in Subscribe-Event-Group entry set to Evt-Grp-Id-1-Service-1. 5. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 07_{16} (SOMEIP_ENTRY_SUBSCRIBE_ACK); — service identifier in entry array set to Service-Id-1; — SOME/IP expected Event-Group-Id in Subscribe-Event-Group ACK entry set to Evt-Grp-Id-1-Service-1. 6. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 5: The IUT sends the SOME/IP notification message.
Remark	---

Table 36 specifies the CTC_SOMEIPSRV_FORMAT_20 – Length of type 2 entries test case.

Table 36 — CTC_SOMEIPSRV_FORMAT_20 – Length of type 2 entries

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_20 – Length of type 2 entries
Purpose	This CTC verifies that the length of type 2 entries is 16 bytes.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00183'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 36 *(continued)*

Item	Content
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry set to Service-Id-1. 4. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 06_{16} (SOMEIP_ENTRY_SUBSCRIBE); — service identifier in entry array set to Service-Id-1; — Event-Group-Id in Subscribe-Event-Group entry set to Evt-Grp-Id-1-Service-1. 5. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 07_{16} (SOMEIP_ENTRY_SUBSCRIBE_ACK); — service identifier in entry array set to Service-Id-1; — SOME/IP expected Event-Group-Id in Subscribe-Event-Group ACK entry set to Evt-Grp-Id-1-Service-1. 6. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 5: The IUT sends the SOME/IP notification message. After step 5: The LT verifies that the received SOME/IP notification message contains the entry length field which is set to (Number-Of-Entries × 16).
Remark	---

Table 37 specifies the CTC_SOMEIPSRV_FORMAT_21 – Index first option run field of the type 2 entry test case.

Table 37 — CTC_SOMEIPSRV_FORMAT_21 – Index first option run field of the type 2 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_21 – Index first option run field of the type 2 entry
Purpose	This CTC verifies that the "Index First Option Run" field of the type 2 entry format layout carries the index of the first option of the first option run of this entry in the option array.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00386'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 37 *(continued)*

Item	Content
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry array set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry array set to Service-Id-1. 4. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 06_{16} (SOMEIP_ENTRY_SUBSCRIBE); — service identifier in entry array set to Service-Id-1; — Event-Group-Id in Subscribe-Event-Group entry set to Evt-Grp-Id-1-Service-1. 5. The LT Client-1 shall listen up to $t_{\text{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 07_{16} (SOMEIP_ENTRY_SUBSCRIBE_ACK); — service identifier in entry array set to Service-Id-1; — SOME/IP expected Event-Group-Id in Subscribe-Event-Group ACK entry set to Evt-Grp-Id-1-Service-1. 6. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 5: The IUT sends the SOME/IP notification message. After step 5: The LT verifies that the received SOME/IP notification message contains the index 1st option in entry which is set to 0.
Remark	---

Table 38 specifies the CTC_SOMEIPSRV_FORMAT_23 – Service identifier field of the type 2 entry test case.

Table 38 — CTC_SOMEIPSRV_FORMAT_23 – Service identifier field of the type 2 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_23 – Service identifier field of the type 2 entry
Purpose	This CTC verifies that the service identifier field of the type 2 entry format layout carries the service identifier of the event groups service, statically configured using the parameter SdServerservice identifier and SdClientservice identifier, depending on being a server or client entry.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 38 *(continued)*

Item	Content
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00193'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry array set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry array set to Service-Id-1. 4. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 06_{16} (SOMEIP_ENTRY_SUBSCRIBE); — service identifier in entry array set to Service-Id-1; — Event-Group-Id in Subscribe-Event-Group entry set to Evt-Grp-Id-1-Service-1. 5. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 07_{16} (SOMEIP_ENTRY_SUBSCRIBE_ACK); — service identifier in entry array set to Service-Id-1; — SOME/IP expected Event-Group-Id in Subscribe-Event-Group ACK entry set to Evt-Grp-Id-1-Service-1. 6. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 5: The IUT sends the SOME/IP notification message.
Remark	---

Table 39 specifies the CTC_SOMEIPSRV_FORMAT_24 – InstanceID field of the type 2 entry test case.

Table 39 — CTC_SOMEIPSRV_FORMAT_24 – InstanceID field of the type 2 entry

Item	Content
CTC # – Title	CTC_SOMEIPSRV_FORMAT_24 – InstanceID field of the type 2 entry

© ISO 2021 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021

Table 39 *(continued)*

Item	Content
Purpose	This CTC verifies that the instance identifier field of the type 2 entry format layout carries the instance identifier of the event groups service statically configured using the parameter SdServerServiceInstanceID and SdClientServiceInstanceID, depending on being a server or client entry.
Reference	*AUTOSAR Specification of Service Discovery, V1.2.0, R4.1 Rev 3* [10], '7.3 SWS_SD_00195'
Prerequisite	The IUT shall be powered on and shall be linked-up with the test system. The IUT shall implement the TCP/IP TestStub in accordance with 6.6. The IUT shall be running and offering the ETS.
Set-up	The test system set-up shall be in accordance with Figure 8. The test system shall be parameterized in accordance with the configuration set-up parameters as specified in 7.1.5.1 and 7.1.5.2.
Step	1. IUT CONFIGURE: Shall start service on IUT-Iface-0 with the following information: — service identifier: Service-Id-1; — number of instances: 1. 2. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 00_{16} (SOMEIP_ENTRY_FIND_SERVICE); — service identifier in entry array set to Service-Id-1. 3. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 01_{16} (SOMEIP_ENTRY_OFFER_SERVICE); — service identifier in entry array set to Service-Id-1. 4. The LT Client-1 shall send a SOME/IP notification message to IUT through IUT-Iface-0 containing: — entry type set to 06_{16} (SOMEIP_ENTRY_SUBSCRIBE); — service identifier in entry array set to Service-Id-1; — Event-Group-Id in Subscribe-Event-Group entry set to Evt-Grp-Id-1-Service-1. 5. The LT Client-1 shall listen up to $t_{\texttt{Param-Listen-Time}}$ on IUT-Iface-0 containing: — entry type set to 07_{16} (SOMEIP_ENTRY_SUBSCRIBE_ACK); — service identifier in entry array set to Service-Id-1; — SOME/IP expected Event-Group-Id in Subscribe-Event-Group ACK entry set to Evt-Grp-Id-1-Service-1. 6. IUT CONFIGURE: Shall stop service on IUT-Iface-0 with the service identifier: Service-Id-1.
Iteration	Not applicable
Expected response	After step 3: The IUT sends the SOME/IP notification message. After step 5: The IUT sends the SOME/IP notification message. After step 5: The LT verifies that the received SOME/IP notification message contains the InstanceID field in the entry which is set to Service-Id-1-Instance-Id.
Remark	---

Table 40 specifies the CTC_SOMEIPSRV_FORMAT_25 – Major version field of the type 2 entry test case.

© ISO 2021 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO 21111-11:2021