ETSI TS 103 813 V15.0.0 (2021-02)

TECHNICAL SPECIFICATION

Smart Secure Platform (SSP);
Test Specification,
SPI interface
(Release 15)

Reference

DTS/SCP-00T103813vf00

Keywords

SSP, SPI, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

- 0 early working draft;
- 1 presented to TC SCP for information;
- 2 presented to TC SCP for approval;
- 3 or greater indicates TC SCP approved document under change control.

Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document defines tests for the specific Serial Peripheral Interface (SPI) implementations defined in ETSI TS 103 713 [1] independently of the respective manufacturer.

1 Scope

The present document covers the minimum characteristics which are considered necessary for the Serial Peripheral Interface (SPI) communication of an SSP in order to provide compliance to ETSI TS 103 713 [1].

The present document specifies the test cases for:

- the physical layer (electrical characteristics);
- the MAC layer;
- the data link layer; and
- the SHDLC layer (as defined in ETSI TS 102 613 [5]);

of the SPI.

Tests for the usage of an SPI different to what is defined in ETSI TS 103 713 [1] are out of scope of the present document.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 103 713: "Smart Secure Platform (SSP); SPI interface".

[2] ISO/IEC 9646-7: "Information technology -- Open Systems Interconnection - Conformance testing methodology and framework -- Part 7: Implementation Conformance Statements".

[3] IEEE 1149.1™-2013: "IEEE Standard for Test Access Port and Boundary-Scan Architecture".

[4] ETSI TS 103 666-1: "Smart Secure Platform (SSP); Part 1: General characteristics".

[5] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Physical and data link layer characteristics".

[6] ISO/IEC 13239: "Information technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non-specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definition of terms, symbols, abbreviations and formats

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 103 713 [1] and the following apply:

JTAG: Joint Test Action Group, synonym for IEEE 1149.1-2013 [3].

SPI Master: expression used to describe the *Master* entity connected to an SPI

NOTE: Descriptive text may use "master" or "SPI_M" as synonyms.

SPI Slave: expression used to describe the *Slave* entity connected to an SPI

NOTE: Descriptive text may use "slave" or "SPI_S" as synonyms.

test case: textual description of conditions, procedures and functions appropriate to verify specific conformance requirements

NOTE: The present document is not providing test cases as test scripts or part of an ATS. In general, a test case as defined here consists of initial conditions and a specific test case description.

test tool: test implementation fulfilling the requirements of test environment defined for the particular test case

NOTE: The present document uses "test tool" and "conformance test system" as synonyms.

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI TS 103 713 [1] apply:

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 103 713 [1] and the following apply:

ATS	Abstract Test Suite
ICS	Implementation Conformance Statement
PCB	Printed Circuit Board
POT	Power On Time
SPI_M	SPI Master
SPI_S	SPI Slave
SUT	System Under Test

TT Test Tool

3.4 Formats

3.4.1 Format of the table of optional features

The columns in the optional features table, Table 4.1, have the following meaning:

Column	Meaning
Item	Item number, incrementing with each item added to the table
Option	Description of the optional feature that might be supported by the implementation
Status	The status of the optional feature is described following notations defined in ISO/IEC 9646-7 [2]: O optional - the feature may be supported or not (default value) O.i qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table
Release	Number of the version the feature was introduced in.
Support	The column is blank in the pro forma and shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [2], are used for the support column in Table 4.1: Y or y supported by the implementation N or n not supported by the implementation N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status)
Mnemonic	The "Mnemonic" column contains mnemonic identifiers for each optional feature

3.4.2 Format of the applicability table

The columns in the applicability tables, Table 4.2a, Table 4.3a and Table 4.4, have the following meaning:

Column	Meaning
Test/Seq.	A reference to the test case number(s) detailed in the present document and required to validate the implementation of the corresponding item in the "Description" column. Tests consisting of more than one logical part are split into sequences. The sequence number is given after the slash.
Description	A short non-exhaustive description of the test purpose is given here. In general, the description text used will equal the test case name used in the present document.
Release	Number of the version the tested feature was introduced in.
Rel-<x>	For a given Release, the corresponding "Rel-<x>" column lists the tests required for the SPI to be declared compliant to this Release. Each entry shows the status following notations defined in ISO/IEC 9646-7 [2]: M mandatory - the capability is required to be supported. O optional - the capability may be supported or not. N/A not applicable - in the given context, it is impossible to use the capability. X prohibited (excluded) - there is a requirement not to use this capability in the given context. Oi qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies a unique group of related optional items and the logic of their selection which is defined immediately following the table. Ci conditional - the requirement on the capability ("M", "O", "X" or "N/A") depends on the support of other optional or conditional items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE ...) ELSE ..." shall be used to avoid ambiguities.
Support	Is blank in the pro forma and is to be completed by the manufacturer in respect of each particular requirement to indicate the choices, which have been made in the implementation.

3.4.3 Format of the conformance requirements tables

The columns in the requirement tables in clause 5 have the following meaning:

Column	Meaning
Req.ID	This column shows the ordinal term assigned to a requirement identified in the referenced specification. The following syntax has been used to define the unique R(equirement) terms: R<n><XX><YY>_<ZZZ> n: Identification letter for the referenced specification Q: ETSI TS 103 713 [1] X: ETSI TS 102 613 [5] XX: Main clause of the core specification in which the conformance requirement is listed. YY: Subclause of the main clause in the core specification in which the conformance requirement is listed. ZZZ: Continuously increasing number starting with '001'.
Clause	The "Clause" column helps to identify the location of a requirement by listing the clause hierarchy down to the sub-clause the requirement is located in.
Release	An optional column that is used if the listed requirement is valid for a specific release or a specific range of releases only, up to a specific release, or from a specific release onwards.
Description	In this column the requirement text is shown. Where the text can either be a copy of the original requirement as found ETSI TS 103 713 [1] or ETSI TS 102 613 [5], or a text analogous to the requirement text (e.g.: if the requirement text is descriptive and can be shortened or truncated).

3.4.4 Numbers and Strings

The conventions used for decimal numbers, binary numbers and strings.

Table 3.1: Convention of Numbering and Strings

Convention	Description
nnnnn	A decimal number, e.g. PIN value or phone number
'b'	A single digit binary number
'bbbbbbbb'	An 8-bit binary number
'hh'	A single octet hexadecimal number
'hh hh…hh'	A multi-octet hexadecimal number or string
"SSSS"	A character string
NOTE: If an 'X' is present in a binary or hexadecimal number, then the digit might have any allowed value. This 'X' value does not need to be interpreted within the particular coding shown.	

3.4.5 Format of the sequences in the test procedure

The columns in the sequence tables for the test procedures have the following meaning:

Column	Meaning
Step	The "Step" column contains a step number, incrementing with each step added to the test procedure. A step may include a number of actions that have to be executed in parallel e.g.: on different instances. In such cases the step column will group all rows of the different instances and provide a single step number.
Direction	The "Direction" column helps to identify initiator and recipient of an action, e.g. SPI_M → SPI_S indicates that data is send from master to slave. The column may also show a single instance e.g. TT, if an action has no direction or is executed solely on the named instance.
Action/Task	The "Action/Task" column shall provide information about an action or task or a set of actions or tasks that are to be executed within the related step.
Description/Expectation	"Description/Expectation" is a column that can be used to provide more details about an action or task, describe functions that are implicit to the action or the result of the action to other instances. When needed, it provides information about the procedure needed to verify a requirement.
REQ	The "REQ" column shows a reference to the requirements that the test tool shall verify in the related step. In cases where it is not obvious how the verification has to be done the "Description" column shall contain explanatory text for the verification task.

4 Test environments

4.0 Description of the test environments

The test environment shall comply with the requirements specified in to ETSI TS 103 713 [1].

The tests are executed in dedicated test environments, offering interfaces to the SUT to activate, operate and trace the communication on the SUT. The tests are performed at open standardized interfaces that are not accessible to a normal user. The tests are specified at the detailed protocol level and are not usually based on functionality as experienced by a user. By providing this high degree of control over the sequence and contents of the protocol messages exchanged between the SUT and the TT, the TT shall be able to explore a wide range of expected and unexpected SUT behaviour.

Figure 4.1 and Figure 4.2 illustrate the dedicated test environments for master and slave testing involving:

- the SUT;
- the Test Case Description;
- the Test Tool;
- the Test Tool Connector;
- a vault for the Test Results.

For testing specific functions, to activate the SIP bus in a particular way or to initiate communication with predefined test data the emulation of either Master or Slave is required.

For testing Master functionality an Emulated Slave (ES) (see Figure 4.1) can take over the role of the Slave.

In cases where Slave functionality is tested an Emulated Master (EM) (see Figure 4.2) is provided by the Test Tool.

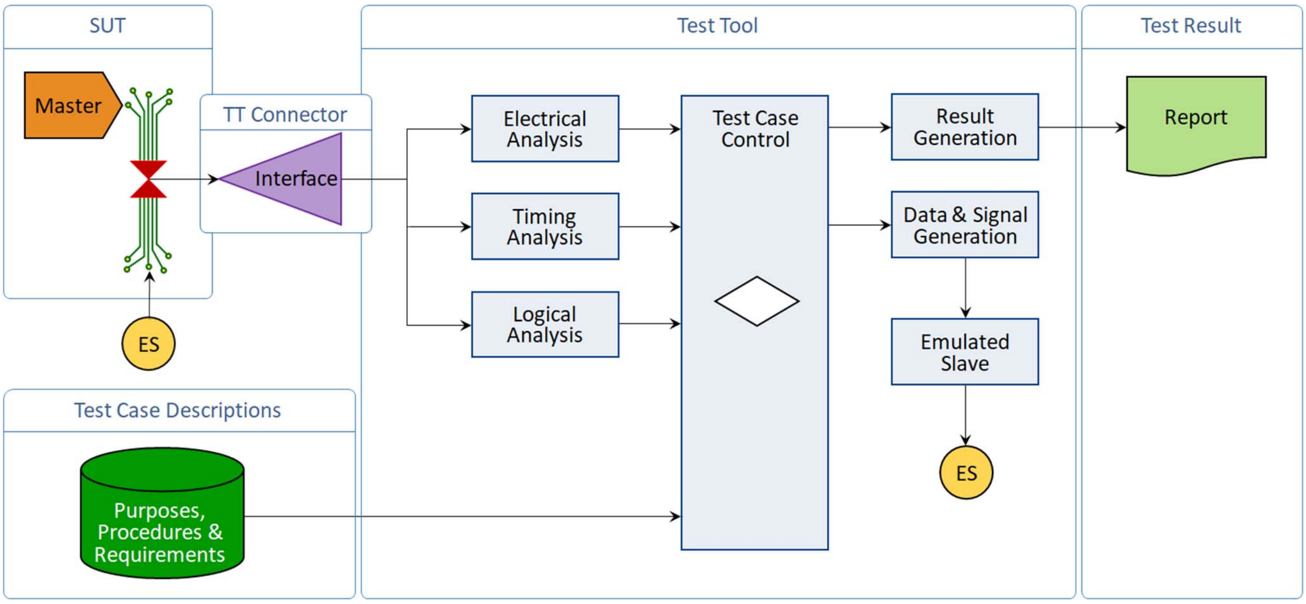

Figure 4.1: Test environment for SPI master testing with emulated slave

The SUT interfaces with the SPI bus via a TT Connector that allows signal tracing and interaction with the test tool.

The building blocks of the SUT in a test environment for SPI Master testing are:

- Master;
- Emulated Master;
- SPI (bus system).

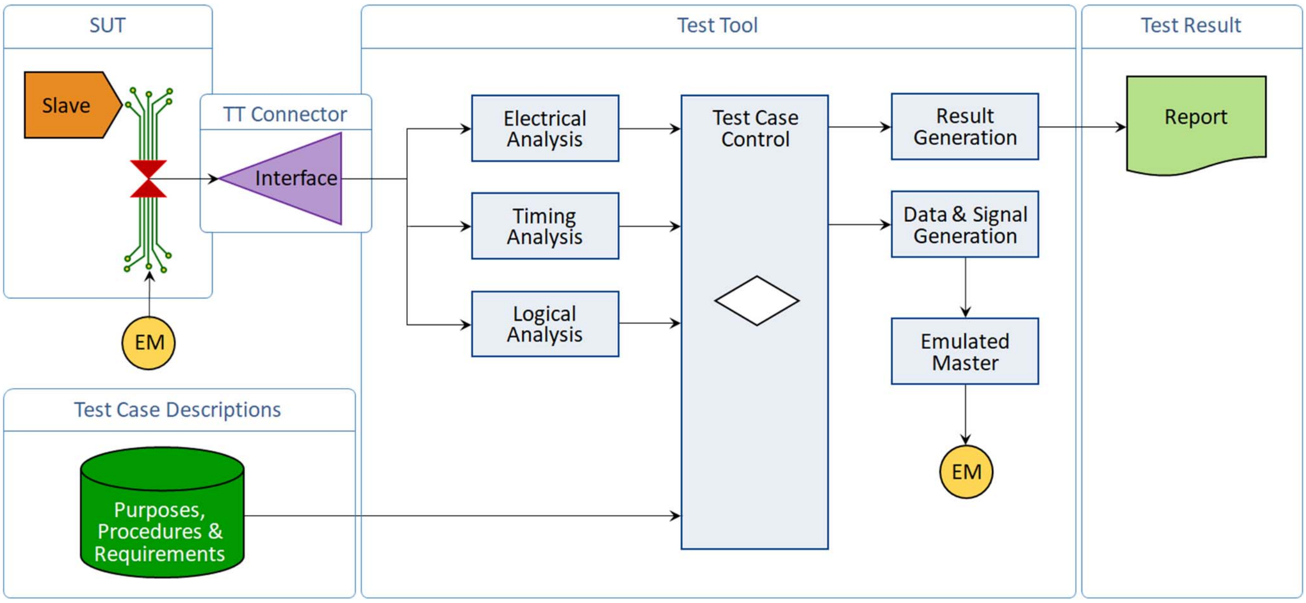

Figure 4.2: Test environment for SPI slave testing with emulated master

The building blocks of the SUT in a test environment for SPI Slave testing are:

- Slave;
- Emulated Master;
- SPI (bus system).

NOTE: Emulated instances will be used to operate tests but will not be under test themselves or become part of the SUT.

The Test Case Descriptions are a repository of test purposes, conformance requirements, initial conditions and information about the test procedures steps generated from this test specification.

The *Test Tool* shall allow electrical measurement of the SPI I/O behaviour. To interpret the current and voltage measurements taken, processes for Electrical Analysis and Timing Analysis shall be implemented. The I/O behaviour on the SPI shall also undergo a Logical Analysis to extract protocol and payload data.

A Test Case Control entity combines all data generated from measurements and analysis with Test Case Descriptions to keep track of test steps and conformance requirements. Depending on the required action the Test Case Control can either provide information to the Data & Signal Generation or initiate the Result Generation.

If required to execute a particular test step, the Data & Signal Generation can provide data, a clock signal and the supply voltage to operate an Emulated Master (EM) or an Emulated Slave (ES).

The Result Generation will generate a *Report* and transfer it as the Test Result.

It is recommended to run JTAG testing as defined in IEEE 1149.1-2013 [3], to cope with typical SPI bus issues like the lack of methods to identify cabling or connection issues or missing hardware slave acknowledgment.

Testing in accordance to IEEE 1149.1-2013 [3] is not mandatory, may be executed separately, and is out of scope of this specification.

4.1 Testing architectures

4.1.0 Description of the testing architectures

ETSI TS 103 713 [1] defines two different implementations of the SPI:

- SPI with a physical interface using 5 signals (see ETSI TS 103 713 [1], clause 6.2);
- SPI with a physical interface using 4 signals (see ETSI TS 103 713 [1], clause 6.3).

For testing the SPI, the test tool is connected to all SPI signals and to the power supply for the master and slave under test to ensure that all signalling can be traced and that the electrical behaviour of the SPI signals can be analysed in conjunction with and in relation to the supply power at the time of the test.

To operate tests in the defined way a physical connection between the test tool and the SPI is required. These Test Tool Connectors are shown in Figures 4.3 and 4.4 below (see elements marked with *). Each physical connection shall allow noise free and unimpeded operation (see the measurement uncertainties in Annex A). If the test tool is emulating the master or the slave, the contacts of the physical instance, replaced by the emulation, shall be switched off, set to idle mode or set to a de-selected sub-state depending on the emulation (master or slave) being used.

4.1.1 Testing architecture - SPI with an electrical interface using 5 signals

The testing architectures for a 5 signals SPI with connection and emulation options are shown in Figure 4.3 below.

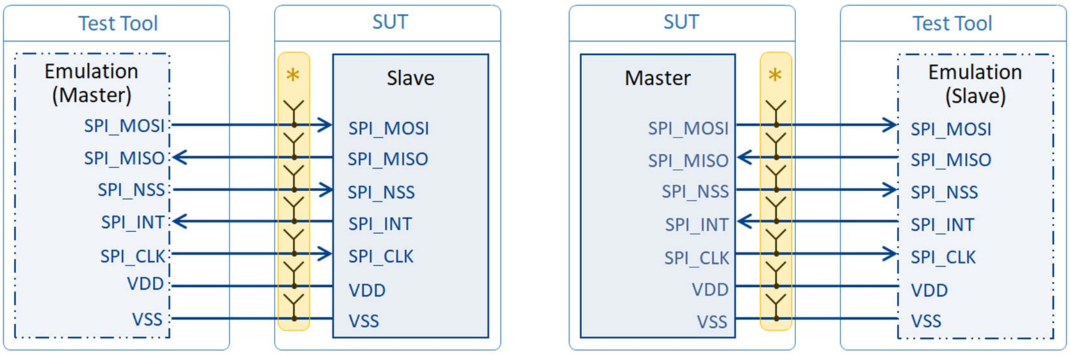

Figure 4.3: Testing architecture - 5 signals SPI

4.1.2 Testing architecture - SPI with an electrical interface using 4 signals

The testing architecture for a 4 signals SPI with connection and emulation options are shown in Figure 4.4 below.

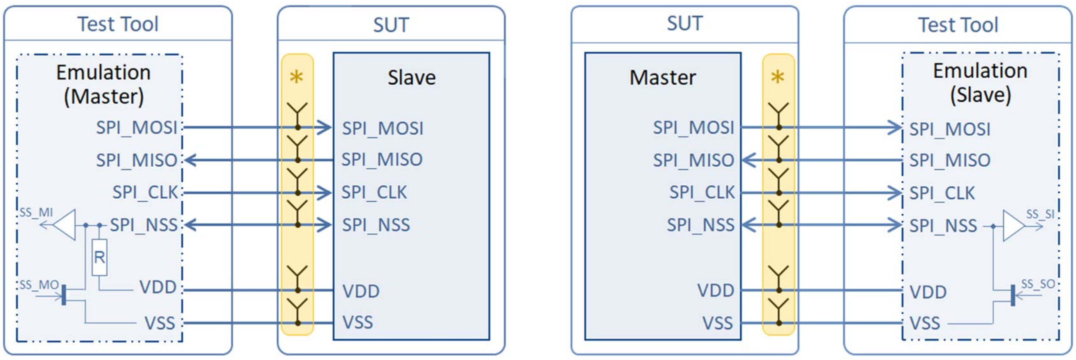

Figure 4.4: Testing architecture - 4 signals SPI

4.2 Test tool

4.2.0 Capabilities of the test tool

The test tool shall provide emulation capabilities for the master and the slave:

- In cases where the TT acts as a master it shall be able to adjust the communication to the parameters provided by the slave (e.g.: SPI_CLK).
- In cases where the TT acts as a slave it shall be able to receive the data sent by the master in the same manner as the 'original' slave does (in terms of clock frequency, trigger conditions, etc.)
- In cases where the test tool acts as a slave initiating a communication it shall operate the SPI_INT in conjunction with the SPI_NSS in the manner defined in ETSI TS 103 713 [1].
- Measurement capabilities and uncertainties, signal tracing and generation capabilities of the test tool shall be in accordance with the definitions in Annex A.
- Test data to be sent to generate the expected data to be received is defined in Annex B.

4.2.1 Adaption of the test tool

For the electrical characteristics input and output values are defined. An appropriate measurement can only be performed if the measurement equipment can be directly connected to the buffer under test.

It is assumed that the TT is connected to the SPI signal lines via one single connector. Thus, separate measurement of input and output voltages is not possible. As the transmission lines are expected to have a few millimetres of length only and the transmission frequency is within the lower MHz range, electrical losses on the SPI bus system itself do not need to be calculated.

Electrical losses arising from the connection of the TT to the SPI bus system have to be known by the TT and are to be subtracted out from the measurement result in a reproducible way. The maximum additional capacitance caused by the connection of the test tool is defined in Annex A.

4.3 Table of optional features

The product vendor shall declare which optional features are supported by their implementation. ICS relevant optional features are listed in Table 4.1.

See clause 3.4.1 for the format of the table.

Table 4.1: Table of optional features

Item	Option	Status	Release	Support	Mnemonic
1	Physical Interface with 5 signals		Rel-15	M.1	O_PI_5_SIGNAL
2	Physical Interface with 4 signals		Rel-15	M.1	O_PI_4_SIGNAL
3	Support of voltage class B		Rel-15	O	O_V_CLASS_B
4	Support of voltage class C		Rel-15	O	O_V_CLASS_C
5	MTU = 64 byte		Rel-15	O	O_MTU_64
6	MTU = 128 byte		Rel-15	O	O_MTU_128
7	MTU = 258 byte		Rel-15	O	O_MTU_256
8	CLT supported		Rel-15	O	O_CLT
9	Support of slave driven flow control		Rel-15	O	O_SLAVE_FLOW_CONTROL
10	Support of Full Power Mode 2		Rel-15	O	O_FULL_POWER_MODE_2
11	Support of Full Power Mode 3		Rel-15	O	O_FULL_POWER_MODE_3
12	Supports of Case 1 data transfer		Rel-15	O	O_CASE_1
13	Supports of Case 2 data transfer		Rel-15	O	O_CASE_2
14	Supports of Case 3 data transfer		Rel-15	O	O_CASE_3
15	Supports of Case 4 data transfer		Rel-15	O	O_CASE_4
16	Supports of Case 5 data transfer		Rel-15	O	O_CASE_5
17	Supports of Case 6 data transfer		Rel-15	O	O_CASE_6
18	Supports of Case 7 data transfer		Rel-15	O	O_CASE_7

Item	Option	Status	Release	Support	Mnemonic
19	Master supports the power saving mode		Rel-15	O	O_MASTER_PSM
20	Slave supports the power saving mode		Rel-15	O	O_SLAVE_PSM
M.1	The physical interface of the SUT has to be identified by selecting the appropriate item. The selection of either Item 1 or item 2 is mandatory				

4.4 Applicability tables

4.4.1 Applicability table - testing of the SPI Master

Table 4.2a specifies the applicability of each test case to the SUT in accordance to Figure 4.1. For testing in accordance to Table 4.2a the SPI master is the SUT, testing is performed with an emulated SPI slave.

The applicability table in this clause is formatted as described in clause 3.4.2.

Table 4.2a: Applicability table - Master testing

Test/Seq.	Description	Release	Rel-15	Support
6.1.1/1	5 signals SPI - DC characteristics for operational voltage class B	Rel-15	C001	
6.1.2/1	5 signals SPI - DC characteristics for operational voltage class C	Rel-15	C002	
6.1.3 - 5 signals SPI -AC characteristics for operational voltage class B				
6.1.3/1	Determination of clock specific AC characteristics	Rel-15	C001	
6.1.3/2	Determination of assertion related AC characteristics	Rel-15	C001	
6.1.3/3	Determination of AC characteristic related to data transfer	Rel-15	C001	
6.1.4 - 5 signals SPI - AC characteristics for operational voltage class C				
6.1.4/1	Determination of clock specific AC characteristics	Rel-15	C002	
6.1.4/2	Determination of assertion related AC characteristics	Rel-15	C002	
6.1.4/3	Determination of AC characteristic related to data transfer	Rel-15	C002	
6.3.1/1	4 signals SPI - DC characteristics for operational voltage class B	Rel-15	C003	
6.3.2/1	4 signals SPI - DC characteristics for operational voltage class C	Rel-15	C004	
6.3.3 - 4 signals SPI - AC characteristics for operational voltage class B				
6.3.3/1	Determination of clock specific AC characteristics	Rel-15	C003	
6.3.3/2	Determination of assertion related AC characteristics	Rel-15	C003	
6.3.3.3	Determination of AC characteristic related to data transfer	Rel-15	C003	
6.3.4 - 4 signals SPI - AC characteristics for operational voltage class C				
6.3.4/1	Determination of clock specific AC characteristics	Rel-15	C004	
6.3.4/2	Determination of assertion related AC characteristics	Rel-15	C004	
6.3.4/3	Determination of AC characteristic related to data transfer	Rel-15	C004	
7.1.1/1	5 signals SPI - Master behaviour during initial data transfer initiation	Rel-15	C005	
7.1.2/1	5 signal SPI - Master behaviour during data transfer initiation	Rel-15	C005	
7.1.3/1	5 signals SPI - Master behaviour during simultaneous data transfer initiation	Rel-15	C005	
7.1.4/1	5 signals SPI - MAC deactivation	Rel-15	C005	
7.3.1/1	4 signal SPI - Master behaviour during initial data transfer initiation	Rel-15	C006	
7.3.2/1	4 signal SPI - Master behaviour during data transfer initiation	Rel-15	C006	
7.3.3/1	4 signal SPI - Master behaviour during simultaneous data transfer initiation	Rel-15	C006	
7.3.4/1	4 signal SPI - Master behaviour during data flow control	Rel-15	C006	
7.3.5/1	4 signal SPI - MAC deactivation	Rel-15	C006	
8.1.1 - Link layer - Master frame generation				
8.1.1/1	Link layer - Master frame generation - MTU = 32 bytes	Rel-15	M	
8.1.1/2	Link layer - Master frame generation - MTU = 64 bytes	Rel-15	C007	
8.1.1/3	Link layer - Master frame generation - MTU = 128 bytes	Rel-15	C008	
8.1.1/4	Link layer - Master frame generation - MTU = 256 bytes	Rel-15	C009	
8.1.2/1	Link layer - Master frame generation - SPI access longer than frame	Rel-15	M	
8.1.3/1	Slave frame retrieval by Master	Rel-15	M	
8.1.4/1	FFS			
8.3.1/1	Case 2 - Slave MAC access request, retrieve the slave frame	Rel-15	C012	

Test/Seq.	Description	Release	Rel-15	Support
8.3.2/1	Case 3 - Slave frame transferred in a single access, NSD attached	Rel-15	C013	
8.3.3/1	Case 4 - Slave MAC access request, frame partially retrieved	Rel-15	C014	
11.1.1/1	Behaviour during MCT activation - no MCT_READY response	Rel-15	M	
11.1.2/1	MCT_MASTER_REQ values	Rel-15	M	
13.1.1/1	SPI Master entering power saving mode	Rel-15	C015	
13.1.2/1	SPI Master resuming from power saving mode	Rel-15	C015	

Table 4.2b: Execution clauses for applicability Table 4.2a

C001	IF O_PI_5_SIGNAL AND O_V_CLASS_B THEN M ELSE N/A
C002	IF O_PI_5_SIGNAL AND O_V_CLASS_C THEN M ELSE N/A
C003	IF O_PI_4_SIGNAL AND O_V_CLASS_B THEN M ELSE N/A
C004	IF O_PI_4_SIGNAL AND O_V_CLASS_C THEN M ELSE N/A
C005	IF O_PI_5_SIGNAL THEN M ELSE N/A
C006	IF O_PI_4_SIGNAL THEN M ELSE N/A
C007	IF O_MTU_64 THEN M ELSE N/A
C008	IF O_MTU_128 THEN M ELSE N/A
C009	IF O_MTU_256 THEN M ELSE N/A
C010	IF O_FULL_POWER_MODE_2 THEN M ELSE N/A
C011	IF O_FULL_POWER_MODE_3 THEN M ELSE N/A
C012	IF O_CASE_2 THEN M ELSE N/A
C013	IF O_CASE_3 THEN M ELSE N/A
C014	IF O_CASE_4 THEN M ELSE N/A
C015	IF O_MASTER_PSM THEN M ELSE N/A

4.4.2 Applicability table - testing of the SPI Slave

Table 4.3a specifies the applicability of each test case to the SUT in accordance to Figure 4.2. For testing in accordance to Table 4.3a the SPI slave is the SUT, testing is performed with an emulated master.

The applicability table in this clause is formatted as described in clause 3.4.2.

Table 4.3a: Applicability table - Slave testing

Test/Seq.	Description	Release	Rel-15	Support
6.2.1/1	5 signals SPI - Class B, AC characteristics for slave driven signal	Rel-15	C101	
6.2.2/1	5 signals SPI - Class C, AC characteristics for slave driven signals	Rel-15	C102	
6.4.1/1	4 signals SPI - Class B, AC characteristics for slave driven signals	Rel-15	C103	
6.4.2/1	4 signals SPI - Class C, AC characteristics for slave driven signals	Rel-15	C104	
6.5.1 - Verification of Slave states - Initial state				
6.5.1/1	Initial state after VDD valid	Rel-15	M	
6.5.1/2	Initial state after reset	Rel-15	M	
6.5.2 - Verification of Slave states - Configured state				
6.5.2/1	Configured state, de-selected sub-state during activation	Rel-15	M	
6.5.2/2	Configured state, de-selected sub-state after issuing a MAC access	Rel-15	M	
6.5.2/3	Configured state, selected sub-state, return to de-selected sub-state	Rel-15	M	
6.5.2/4	Configured state, selected sub-state, return to initial state	Rel-15	M	

Test/Seq.	Description	Release	Rel-15	Support
6.5.3 - Verification of Slave states, 5 signals SPI - Pro-active state				
6.5.3/1	Pro-active state, MAC access from slave, return to configured/de-selected state	Rel-15	C105	
6.5.3/2	Pro-active state, simultaneous initiation, return to configured/selected state	Rel-15	C105	
6.5.3/3	Pro-active state, MAC access from slave, return to initial state	Rel-15	C105	
6.5.4 - Verification of Slave states, 4 signals SPI - Pro-active state				
6.5.4/1	Pro-active state, MAC access from slave, return to configured/de-selected state	Rel-15	C106	
6.5.4/2	Pro-active state, simultaneous initiation, return to configured/selected state	Rel-15	C106	
6.5.4/3	Pro-active state, MAC access from slave, return to initial state	Rel-15	C106	
6.5.5/1	Verification of Slave states - 4 signals SPI - Busy state	Rel-15	C106	
7.2.1/1	5 signal SPI - Slave behaviour at initial MAC activation	Rel-15	C105	
7.2.2/1	5 signal SPI - Slave behaviour during data transfer initiation - nominal test	Rel-15	C105	
7.2.3/1	5 signal SPI - Slave behaviour during data transfer initiation by the slave	Rel-15	C105	
7.4.1/1	4 signal SPI - Slave behaviour during data transfer initiation	Rel-15	C106	
7.4.2/1	4 signal SPI - Slave behaviour during simultaneous data transfer initiation	Rel-15	C106	
8.2.1/1	Slave frame generation - SPI access longer than frame	Rel-15	M	
8.2.2 - Link layer - Slave frame retrieval by SPI Master				
8.2.2/1	Slave frame retrieval by Master - NSD bytes not appended	Rel-15	M	
8.2.2/2	Slave frame retrieval by Master - NSD bytes not appended	Rel-15	M	
8.4.1/1	Case 1 - Master sends the frame, no data from the slave	Rel-15	C110	
8.4.2/1	Case 5 - Simultaneous MAC phase initiation, remaining bytes of the slave frame -FFS			
8.4.3/1	Case 6 - Simultaneous MAC phase initiation, short slave frame - FFS			
8.4.4/1	Case 7 - Simultaneous MAC phase initiation, single access - FFS			
9.1.1/1	SHDLC Layer support	Rel-15	M	
9.1.2/1	CLT Layer support	Rel-15	C116	
9.1.3/1	MCT Layer support	Rel-15	M	
11.2.1/1	MCT activation - corrupted MCT_MASTER_REQ response	Rel-15	M	
11.2.2/1	MCT_READY values	Rel-15	M	
13.2.1 - SPI Slave entering power saving mode				
13.2.1/1	Pending slave MAC access request	Rel-15	C115	
13.2.1/2	No assertion of SPI_NSS by the master for a time > T4	Rel-15	C115	
13.2.1/3	Power saving mode declined by the master (T4 = "FFFF")	Rel-15	C115	
13.2.1/4	SHDLC link layer acknowledgement for EVT_LINK_END_OF_OPERATION	Rel-15	C115	
13.2.1/5	Inactivity period ≥ MCT_MASTER_TIMEOUT following the power-up	Rel-15	C115	
13.2.2/1	SPI Slave resuming from power saving mode	Rel-15	C115	

Table 4.3b: Execution clauses for applicability Table 4.3a

C101	IF O_PI_5_SIGNAL AND O_V_CLASS_B THEN M ELSE N/A
C102	IF O_PI_5_SIGNAL AND O_V_CLASS_C THEN M ELSE N/A
C103	IF O_PI_4_SIGNAL AND O_V_CLASS_B THEN M ELSE N/A
C104	IF O_PI_4_SIGNAL AND O_V_CLASS_C THEN M ELSE N/A
C105	IF O_PI_5_SIGNAL
C106	IF O_PI_4_SIGNAL
C107	IF O_MTU_64 THEN M ELSE N/A
C108	IF O_MTU_128 THEN M ELSE N/A
C109	IF O_MTU_256 THEN M ELSE N/A
C110	IF O_CASE_1 THEN M ELSE N/A
C111	IF O_CASE_5 THEN M ELSE N/A
C112	IF O_CASE_6 THEN M ELSE N/A
C113	IF O_CASE_7 THEN M ELSE N/A
C114	IF O_SLAVE_FLOW_CONTROL
C115	IF O_SLAVE_PSM THEN O ELSE N/A

C116	IF O_CLT THEM M ELSE N/A

4.4.3 Applicability table - Higher level protocol testing

Table 4.4 specifies the applicability for higher level protocol testing (e.g. SHDLC). For those test cases no differentiation between SPI master and SPI slave is made as the communication can be started at either endpoint. These test cases use endpoints named 'SUT' for the System Under Test and 'TT' for the Test Tool where the test tool is generating the frames required for testing.

The applicability table in this clause is formatted as described in clause 3.4.2.

Table 4.4: Applicability table - Higher level protocol testing

Test/Seq.	Description	Release	Rel-15	Support
12.1.1/1	Error management - corrupted I-frame	Rel-15	M	
12.1.2/1	Error management - corrupted RR frame	Rel-15	M	
12.2.1/1	Initial state at link reset - reset by the SUT	Rel-15	M	
12.3.1/1	Link establishment by the SUT with default sliding window size	Rel-15	M	
12.3.2/1	Link establishment and connection time out	Rel-15	M	
12.3.3/1	Requesting unsupported window size and/or SREJ support - link establishment by SUT	Rel-15	M	
12.3.4/1	Discard buffered frames on link re-establishment	Rel-15	M	
12.4.1/1	I-frame transmission	Rel-15	M	
12.4.2/1	I-frame reception - single I-Frame reception	Rel-15	M	
12.4.3/1	I-frame reception - multiple I-Frame reception	Rel-15	M	
12.5.1/1	REJ transmission - multiple I-frames received	Rel-15	M	
12.5.2/1	REJ reception	Rel-15	M	
12.6.1/1	Retransmission of multiple frames	Rel-15	M	
12.7.1/1	RNR reception	Rel-15	M	
12.7.2/1	Empty I-frame transmission	Rel-15	M	
12.8.1/1	SREJ transmission	Rel-15	M	
12.8.2/1	SREJ transmission - multiple I-frames received - FFS	Rel-15	M	
12.8.3/1	SREJ reception	Rel-15	M	

4.5 Test cases and test case description

4.5.1 Common conditions and test steps

4.5.1.1 Group specific initial conditions

To execute a test case defined in the present document, predefined initial conditions for the test case and the test case group may need to be fulfilled. Group specific initial conditions may contain descriptions of specific states, execution instructions and/or a reference to group common test steps that need to be executed. If such group specific initial conditions are defined, they will apply to all test cases within the respective test case group.

4.5.1.2 Group common test steps

To avoid the repetition of test steps that are common to test cases of a particular group, those test steps are defined in a separate clause. Group common test steps may need to be executed before the test steps listed in the test procedure, or afterwards.

Test steps to be executed before Step 1 are numbered: 0.1 to 0.n. E.g.:

Step	Direction	Action/Task	Description/Expectation	REQ
0.1	SPI_M	Set SPI_MOSI to high impedance Set SPI_CLK to low state level Set SPI_NSS to high impedance		
0.2	SPI_M	Activate VDD	SPI_S is starting up	
1	...	...	...	...

Test steps n.1 to n.n are to be executed after Step n of the test steps listed in the test procedure of the test case description. E.g.:

Step	Direction	Action/Task	Description/Expectation	REQ
n	...	...	...	...
n.1	SPI_M	Deactivate VDD	SPI_S is going to de-selected mode	

4.5.2 Test case description

4.5.2.0 Structure and description of a test case

Test cases defined in the present document are written in prose. Test case descriptions are available and specific for each test case. The test case description as used for the test cases defined within the present document consist of:

- Test purpose;
- Initial conditions;
- Test procedure;
- Post conditions (optional).

4.5.2.1 Test purpose

The test purpose is a concise and unambiguous description of what is to be tested. The test purpose description should be based on testable requirements identified in the specification named in the related conformance requirement section. If no directly testable requirement could be identified, requirements implicitly tested (see clause 4.6) can be used to describe the test purposes.

To avoid ambiguities, the description given in the test purpose may include a reference to the clause in the specification named in the related conformance requirement section (e.g.: ETSI TS 103 713 [1]) where the tested requirement is specified.

If a single test case is used to verify more than just one requirement, the test purpose description shall be specified as a list of numbered items. E.g.:

1) <Test purpose #1>
2) <Test purpose #2>

4.5.2.2 Initial conditions

In addition to a reference to the common initial conditions and/or the group specific initial conditions the test case description may contain specific initial conditions that apply to the current test case only. The test case specific initial conditions in this sub-clause may contain descriptions of specific states and/or execution instructions that need to be executed.

4.5.2.3 Test procedure

Following the common initial, the group test initial and the test case specific initial conditions, the steps listed in the test procedure define the specific part of the test execution. The test procedure is given in a table format. Each table row consists of the columns:

- Step
- Direction
- Action/Task
- Description/Expectation
- REQ

EXAMPLE:

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_S → SPI_M	Send interrupt on SPI_INT	The SPI_M activates the clock and ...	RQ0601_001 RQ0604_001 RQ0701_001

4.5.2.4 Post conditions

The post conditions, if present, describe actions to be carried out and/or the status of the TT and/or the SUT after the test procedure.

4.6 Implicit testing

For some requirements identified in ETSI TS 103 713 [1], conformance is not verified explicitly in the present document. This does not imply that correct functioning of the described procedures and related features is not essential, but that these are implicitly tested to a sufficient degree.

Descriptive requirements identified in ETSI TS 103 713 [1] that can be implicitly tested, i.e. where the described function or process or the result of an executed function or process can be observed, got assigned an RQ number.

4.7 Pass criterion

Pass criterions are identified in accordance to ISO/IEC 9646-7 [2]:

- A test execution is considered as successful only if the test procedure was fully carried out successfully.
- A test execution is considered as failed if the tested feature provides an unexpected behaviour.
- A test execution is considered as inconclusive if the pass criterion cannot be evaluated due to issues which occur during the setup of the initial conditions or during the execution of a test step to which no conformance requirement is assigned.

5 Conformance requirements

5.0 Conformance requirement references

The conformance requirements that apply to the test cases within the present document are derived from the specification named in the reference text preceding each conformance requirement listing.

5.1 Electrical interfaces

5.1.0 Electrical interfaces - common requirements

Reference: ETSI TS 103 713 [1], clause 6.1.

Req.ID	Clause	Description
RQ0601_001	6.1	An SPI interface implementation shall allow bi-directional communication.
RQ0601_002	6.1	The slave shall be able to initiate communication with the master when it has data available without a prior command from master.

5.1.1 Physical interface with 5 signals

Reference: ETSI TS 103 713 [1], clause 6.2.

Req.ID	Clause	Description
RQ0602_003	6.2	SPI_NSS signal used for the selection of a Slave Endpoint among N slaves sharing the same bus.
RQ0602_004	6.2	SPI_NSS is considered active or asserted at low voltage level.
RQ0602_005	6.2	SPI_MISO, SPI_MOSI and SPI_CLK can be shared between several SPI slaves present on the same SPI bus.
RQ0602_006	6.2	The SPI_INT signal allows the slave to initiate a MAC access request in order to notify the master to start a data transfer.
RQ0602_007	6.2	SPI_INT is defined as an edge-triggered interrupt. It is asserted on the rising edge of the signal.

5.1.2 Physical interface with 4 signals

Reference: ETSI TS 103 713 [1], clause 6.3.

Req.ID	Clause	Description
RQ0603_001	6.3	SPI_NSS is considered active or asserted at low voltage level. SPI_NSS requires a bidirectional IO implementing an Open Drain (OD) interface for both master and slave. This configuration allows driving the SPI_NSS signal to low voltage level by both master and slave without electrical conflict.
RQ0603_002	6.3	A pull-up resistor keeps SPI_NSS at high state level (i.e. idle state) when SS_MO and SS_SO are not asserted.
RQ0603_003	6.3	The SPI_NSS signal is at low state when either SS_MO or SS_SO are asserted.

5.1.3 Electrical characteristics

Reference: ETSI TS 103 713 [1], clause 6.4.

Req.ID	Clause	Description
6.4.1 - DC characteristics		
RQ0604_001	6.4.1	The SPI Electrical specification interface shall be defined for VDD operational voltage classes B and C as defined in ETSI TS 103 666-1 [4], clause 6.2.2.3.
RQ0604_002	6.4.1	In voltage class B the VIH shall be minimum 0,7 × VDD and maximum VDD + 0,5 V.
RQ0604_003	6.4.1	In voltage class B the VIL shall be minimum -0,5 V and maximum 0,3 × VDD.
RQ0604_004	6.4.1	In voltage class B the VOH shall be minimum 0,9 × VDD.
RQ0604_005	6.4.1	In voltage class B the VOL shall be maximum 0,1 × VDD.
RQ0604_006	6.4.1	In voltage class B in case of a 4 signals interface the IOL shall be minimum -1 mA.
RQ0604_007	6.4.1	In voltage class B in case of a 4 signals interface the CI shall be maximum 20 pF
RQ0604_008	6.4.1	In voltage class C the VIH shall be minimum 0,7 × VDD and maximum VDD + 0,3 V.
RQ0604_009	6.4.1	In voltage class C the VIL shall be minimum -0,3 V and maximum 0,3 × VDD.
RQ0604_010	6.4.1	In voltage class C the VOH shall be minimum 0,9 × VDD.
RQ0604_011	6.4.1	In voltage class C the VOL shall be maximum 0,1 × VDD.
RQ0604_012	6.4.1	In voltage class C in case of a 4 signals interface the IOL shall be minimum -1 mA.
RQ0604_013	6.4.1	In voltage class C in case of a 4 signals interface the CI shall be maximum 20 pF.

Req.ID	Clause	Description
RQ0604_014	6.4.1	The value of the resistor R in ETSI TS 103 713 [1], figure 6.2 shall be selected for a resultant maximum current lower than or equal to the minimum between the absolute IOL values of the master and the slave.
6.4.2 - Data transfer mode, AC characteristics		
RQ0604_015	6.4.2	The SPI interface shall implement the SPI mode 0 according to the industry de-facto SPI specification. SPI mode 0 is determined by CPOL = 0 and CPHA = 0.
RQ0604_016	6.4.2	The reference maximum value of SPI_CLK frequency (fCLK) for generic SPI slaves is specified by the slave at initialization in MCT_READY for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_017	6.4.2	The reference minimum value of SPI_CLK low time (tCLKL) for generic SPI slaves is specified as 0,45 × tCLK for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_018	6.4.2	The reference minimum value of SPI_CLK high time (tCLKLH) for generic SPI slaves is specified as 0,45 × tCLK for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_019	6.4.2	The reference minimum value of data setup time to clock rising edge (tSU) for generic SPI slaves is specified as 5 ns for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_020	6.4.2	The reference minimum value of SPI_MOSI hold time/data hold time to clock rising edge (tH) for generic SPI slaves is specified as 3 ns for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_021	6.4.2	The reference minimum value of SPI_MISO hold time/output hold time to clock falling edge (tHO) for generic SPI slaves is specified as 0 ns for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_022	6.4.2	The reference minimum value of SPI_NSS setup time (tCSS 1,8 V) for generic SPI slaves is specified as 63 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_023	6.4.2	The reference minimum value of SPI_NSS setup time (tCSS 3,0 V) for generic SPI slaves is specified as 33 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_024	6.4.2	The reference minimum value of hold time clock falling edge to SPI_NSS inactive (tCSH) for generic SPI slaves is specified as 0,5 × tCLK for 1,8 V and 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_025	6.4.2	The reference minimum value of SPI_NSS inactive time (tCS 1,8 V) for generic SPI slaves is specified as 60 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_026	6.4.2	The reference minimum value of SPI_NSS inactive time (tCS 3,0 V) for generic SPI slaves is specified as 30 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_027	6.4.2	The reference maximum value of SPI_MISO valid delay time from SPI_NSS active (tCSV 1,8 V) for generic SPI slaves is specified as 58 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_028	6.4.2	The reference maximum value of SPI_MISO valid delay time from SPI_NSS active (tCSV 3,0 V) for generic SPI slaves is specified as 28 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_029	6.4.2	The reference minimum value of SPI_MISO valid delay time from clock falling edge (tV) for generic SPI slaves is specified as 0 ns and maximum as 0,7 × tCLKL for 1,8 V to 3,0 V (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_030	6.4.2	The reference minimum value of SPI_MISO output disable time from SPI_NSS inactive (tCSDO 1,8 V) for generic SPI slaves is specified as 0 ns and maximum as 60 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).
RQ0604_031	6.4.2	The reference minimum value of SPI_MISO output disable time from SPI_NSS inactive (tCSDO 3,0 V) for generic SPI slaves is specified as 0 ns and maximum as 30 ns (SPI Slave, Mode 0: CPOL = 0 and CPHA = 0).

5.1.4 Slave state

Reference: ETSI TS 103 713 [1], clause 6.5.

Req.ID	Clause	Description
6.5.1 - Slave state definitions		
RQ0605_001	6.5.1	As soon as the slave is powered on and VDD is valid it enters the Initial state.
RQ0605_002	6.5.1	After a reset the slave enters the Initial state.
RQ0605_003	6.5.1	After the POT time, following VDD valid or a reset, the slave shall transition to the state configured/de-selected.
RQ0605_004	6.5.1	In configured/de-selected state the slave shall not ignore SPI_NSS assertion by master and shall be ready for an SPI access.
RQ0605_005	6.5.1	In configured/de-selected state the slave shall have SPI_MISO in high impedance.
RQ0605_006	6.5.1	In configured/de-selected state the slave shall ignore both SPI_CLK and SPI_MOSI.
RQ0605_007	6.5.1	The slave will be in configured/de-selected state after issuing a MAC access request, until the master generates an access for the data transfer.

Req.ID	Clause	Description
RQ0605_008	6.5.1	In case of a 4 signals interface, SPI_NSS is not asserted by the slave in configured/de-selected state.
RQ0605_009	6.5.1	From configured/de-selected state, the slave shall enter into one of the states: configured/selected, pro-active or power saving mode; or initial in case of a reset.
RQ0605_010	6.5.1	In configured/selected state the slave shall not ignore SPI_NSS assertion by master.
RQ0605_011	6.5.1	In configured/selected state the SPI_MISO is not in high impedance.
RQ0605_012	6.5.1	In configured/selected state the slave shall not ignore SPI_CLK and SPI_MOSI.
RQ0605_013	6.5.1	From configured/selected state, the slave shall enter into one of the states: configured/de-selected, or pro-active; or initial in case of a reset.
RQ0605_014	6.5.1	The slave enters in pro-active state when data need to be sent.
RQ0605_015	6.5.1	The slave exits the pro-active state on its own after T2, regardless of the input signals states driven by the master.
RQ0605_016	6.5.1	In case of a 4 signals interface all input signals shall be ignored in pro-active state.
RQ0605_017	6.5.1	In case of a 4 signals interface the SPI_MISO shall be in high impedance in pro-active state.
RQ0605_018	6.5.1	In case of a 4 signals interface the slave asserts the SPI_NSS in pro-active state.
RQ0605_019	6.5.1	From pro-active state the slave shall transit to configured/de-selected state only if the slave generated a MAC access request and the master did not initiate any MAC phase simultaneously.
RQ0605_020	6.5.1	From pro-active state the slave shall transit to configured/selected state when both the master and the slave generate a MAC phase simultaneously.
RQ0605_021	6.5.1	From pro-active state the slave shall transit to initial state in case of a reset.
RQ0605_022	6.5.1	In the optional busy state the slave shall keep SPI_NSS asserted.
RQ0605_023	6.5.1	In the optional busy state the slave may keep its SPI module enabled.
RQ0605_024	6.5.1	From the optional busy state the slave shall enter the configured/de-selected state.
RQ0605_025	6.5.1	From power-saving-mode state the slave shall transition to the state configured/selected after the master has resumed the slave.
RQ0605_026	6.5.1	From power-saving-mode state the slave shall transition to initial state in case of a reset.

5.2 Data link layer

5.2.0 Overview

Reference: ETSI TS 103 713 [1], clause 7.

Req.ID	Clause	Description
7.1 - Overview		
RQ0701_001	7.1	Clause 9.1 in ETSI TS 102 613 [5] shall apply.

5.2.1 MAC layer

Reference: ETSI TS 103 713 [1], clause 7.2.

Req.ID	Clause	Description
7.2.2 - Timing		
7.2.2.2 - T1 = Slave Ready Time		
RQ0702_001	7.2.2.2	T1 is the MAC phase time required by the slave to get configured and enabled at the end of the MAC phase. The slave shall be ready for the data transfer phase after T1 (i.e. when the SPI_CLK can be started by master).
RQ0702_002	7.2.2.2	Master shall allow at least the time T1 requested by the slave between the start of the MAC phase initiated by either master or slave and the point in time when the data transfer phase starts.
7.2.2.3 - T2 = Slave Request Time		
RQ0702_003	7.2.2.3	T2 is the duration of an SPI_NSS or SPI_INT pulse generated by the slave for a MAC access request. The minimum value of T2 is 1 µs.
RQ0702_004	7.2.2.3	In order to sense the interrupts originating either from slave SPI_NSS or SPI_INT assertion, the master shall be configured for edge-triggered interrupts.
7.2.2.4 - T3 = Slave resume time from power saving mode		
RQ0702_005	7.2.2.4	T3 is the slave resume time from the power saving mode. The slave shall be ready for the SPI access after T3.

Req.ID	Clause	Description
7.2.3 - 5 signals MAC layer		
7.2.3.1 - Initiation of the data transfer from the master		
RQ0702_006	7.2.3.1	In case of a 5 signals interface the master asserts the SPI_NSS at the start of a MAC phase.
RQ0702_007	7.2.3.1	In case of a 5 signals interface after waiting minimum T1 the master starts the bidirectional data transfer by toggling the SPI_CLK signal. In use-cases when it is certain that the slave is not expected to initiate a MAC access request by SPI_INT assertion (e.g. at first access during SPI initialization) master may skip waiting for T1.
RQ0702_008	7.2.3.1	In case of 5 signals interface the master shall de-assert SPI_NSS after data transfer completion.
7.2.3.2 - Initiation of the data transfer from the slave		
RQ0702_009	7.2.3.2	In case of a 5 signals interface before the data transfer is initiated by the slave, the slave shall determine that the SPI_NSS signal is de-asserted (i.e. at the high level state).
RQ0702_010	7.2.3.2	In case of a 5 signals interface when the data transfer is initiated by the slave the, slave shall assert SPI_INT by generating an SPI_INT pulse with a minimum width of T2 seconds.
RQ0702_011	7.2.3.2	In case of a 5 signals interface the data transfer is initiated by the slave, the slave shall configure its SPI module and shall be ready for data transfer from the master after T2.
RQ0702_012	7.2.3.2	The master starts data transfer at a time greater than T1 following the leading edge of SPI_INT, by asserting SPI_NSS
RQ0702_013	7.2.3.2	SPI_CLK starts after the SPI_NSS assertion by the master
RQ0702_014	7.2.3.2	After data transfer completion and SPI_CLK stop, the master de-asserts SPI_NSS
7.2.3.3 - Simultaneous initiation of a data transfer from both master and slave		
RQ0702_015	7.2.3.3	In case of a 5 signals interface when the data transfer is initiated simultaneously by the slave and the master, the resulting procedure from the master perspective is equivalent to the initiation from the master.
RQ0702_016	7.2.3.3	In case of a 5 signals interface when the data transfer is initiated simultaneously by the slave and the master, the slave makes a MAC access request by asserting SPI_INT and waits for the master to generate the access for data transfer.
RQ0702_017	7.2.3.3	In case of a 5 signals interface when the data transfer is initiated simultaneously by the slave and the master, the gap time T1 - T2 shall be long enough for the slave to prepare the SPI module for the data transfer.
7.2.3.4 - MAC activation		
RQ0702_018	7.2.3.4	In case of a 5 signals interface the MAC activation procedure at power on shall be the following: • Master shall set the SPI_NSS output to the de-asserted state. • The master shall drive the VDD power line on. When the power supply of the slave needs to be toggled independently from the power of other devices sharing the SPI bus and while VDD is off or not valid, the slave shall keep SPI_NSS, SPI_CLK, SPI_MOSI and SPI_MISO lines as inputs or in high impedance.
7.2.3.5 - MAC deactivation		
RQ0702_019	7.2.3.5	In case of a 5 signals interface the MAC deactivation procedure for power off shall be the following: • Master shall set the SPI_NSS output to the de-asserted state. • The master shall drive the VDD power line off.
7.2.4 - 4 signals MAC layer		
7.2.4.1 - Initiation of the data transfer from the master		
RQ0702_020	7.2.4.2	In case of a 4 signals interface the master checks the state of the SPI_NSS signal (by reading SS_MI) and if it is de-asserted (i.e. at the high state) the master asserts SS_MO to drive SPI_NSS signal to the asserted state.
RQ0702_021	7.2.4.2	In case of a 4 signals interface after waiting minimum T1 the master starts the bidirectional data transfer by toggling the SPI_CLK signal. If the master is certain that the slave will not initiate a MAC access request by asserting SPI_NSS then the master may skip waiting T1.
RQ0702_022	7.2.4.2	In case of a 4 signals interface the master shall de-assert SPI_NSS after data transfer completion.
7.2.4.2 - Initiation of the data transfer from the slave		
RQ0702_023	7.2.4.3	In case of a 4 signals interface before the data transfer is initiated by the slave, the slave shall determine that the SPI_NSS signal is de-asserted (i.e. at the high level state).
RQ0702_024	7.2.4.3	In case of a 4 signals interface before the data transfer is initiated by the slave the slave shall disable its SPI module.
RQ0702_025	7.2.4.3	In case of a 4 signals interface when the data transfer is initiated by the slave the slave shall assert SS_SO to drive SPI_NSS to the asserted state (i.e. low level) for at least T2 seconds.
RQ0702_026	7.2.4.3	In case of a 4 signals interface after the slave asserted the SS_SO the slave shall enable its SPI module and wait for data transfer from the master.

Req.ID	Clause	Description
RQ0702_027	7.2.4.3	Following the SPI_NSS assertion by the slave the master asserts SS_MO after at least T1.
RQ0702_028	7.2.4.3	SPI_CLK starts after the SPI_NSS assertion by the master.
RQ0702_029	7.2.4.3	After data transfer completion i.e., SPI_CLK stop, the master de-asserts SPI_NSS.
7.2.4.4 - Simultaneous initiation of a data transfer from both master and slave		
RQ0702_030	7.2.4.4	In case of a 4 signals interface when the data transfer is initiated simultaneously by the slave and the master, the resulting procedure from the master perspective is equivalent to the initiation from the master.
RQ0702_031	7.2.4.4	In case of a 4 signals interface when the data transfer is initiated simultaneously by the slave and the master, the slave makes a MAC access request and waits for the master to generate the access for data transfer.
RQ0702_032	7.2.4.4	In case of a 4 signals interface when the data transfer is initiated simultaneously by the slave and the master, the gap time T1 - T2 shall be long enough for the slave to enable the SPI module.
7.2.4.5 - Slave-driven Flow Control		
RQ0702_033	7.2.4.5	In case of a 4 signals interface the slave may assert SS_SO for flow control at any time between the start of the data transfer and SPI_NSS de-assertion by master at end of the data transfer.
RQ0702_034	7.2.4.5	In case of a 4 signals interface if the slave asserts the SPI_NSS (via its SS_SO) during a data transfer in progress for flow control the slave shall not start to send a new frame.
RQ0702_035	7.2.4.5	As long as the SPI_NSS signal is asserted and even if the data transfer is completed, the master shall not run the MAC initiation.
RQ0702_036	7.2.4.5	If the SPI bus is shared among multiple slaves and the slave performing flow control has its SPI module enabled during this time, the master shall not initiate a data transfer to any other slaves on the shared SPI bus as long as the slave keeps the SPI_NSS asserted.
RQ0702_037	7.2.4.5	The slave should not use the flow control mechanism longer than 500 µs.
7.2.4.6 - MAC activation		
RQ0702_038	7.2.4.6	In case of a 4 signals interface the MAC activation procedure at power on shall be the following: • The SS_MO shall be de-asserted (i.e.at low level) setting the SPI_NSS output to high impedance. • The master shall drive the VDD power line on. When the power supply of the slave needs to be toggled independently from the power of the other devices sharing the same SPI bus and while VDD is off or not valid, the slave shall keep SS_SO de-asserted and SPI_CLK, SPI_MOSI and SPI_MISO as inputs or in high impedance.
7.2.4.7 - MAC de-activation		
RQ0702_039	7.2.4.7	In case of a 4 signals interface the MAC deactivation procedure for power off shall be the following: • The SS_MO shall be de-asserted (i.e. at low level) setting the SPI_NSS output to high impedance. • The master shall drive the VDD power line off.

5.2.2 Link layer frame

Reference: ETSI TS 103 713 [1], clause 7.3.

Req.ID	Clause	Description
7.3.1 - Overview		
RQ0703_001	7.3.1	The format of the frames generated by master and slave is determined by the link layer and it is shown in ETSI TS 103 713 [1], figure 7.8. All bytes shall be transmitted with MSB first (most significant bit first).
RQ0703_002	7.3.1	The format of the frames generated by master and slave is determined by the link layer and it is shown in ETSI TS 103 713 [1], figure 7.8. All frames are transferred with the bytes in the order shown in figure 7.8, starting with the LPDU Length.
RQ0703_003	7.3.1	The format of the frames generated by master and slave is determined by the link layer and it is shown in ETSI TS 103 713 [1], figure 7.8. The LPDU field is transferred starting with the LLC Control byte followed by the data generated by the upper OSI layer.

Req.ID	Clause	Description
RQ0703_004	7.3.1	The link layer frame shall contain the following fields: • LPDU length: length of the LPDU, 1 byte. • LPDU (Link Protocol Data Unit): LPDU includes the LLC control byte as defined in ETSI TS 103 713 [1], clause 7.4. • CRC informs about the integrity of the whole frame i.e. Length and LPDU. Detection of errors in a frame shall be based on the 16-bit frame checking sequence as given in ISO/IEC 13239 [6]. The CRC polynomial is: X16 + X12 + X5 + 1. Its initial value is 'FFFF'.
RQ0703_005	7.3.1	Link Layer frames (including header, LPDU and trailer) shall always be prepared with length less than or equal to MTU.
RQ0703_006	7.3.1	MTU values are negotiated at SPI interface initialization as described in clause 7.6 of ETSI TS 103 713 [1]. The resultant MTU shall be the smallest MTU value between the MTU of the master and the MTU of the slave. The MTU shall be the same irrespective of the transfer direction.
RQ0703_007	7.3.1	Non-significant data (NSD) may be appended at the end of a master or slave link layer frame until the end of the SPI access according to the rules described in clause 7.3.1 of ETSI TS 103 713 [1], considering LPDU Length + 3 + NSD length ≤ MTU.
RQ0703_008	7.3.1	"NSD" shall consist of idle bytes set to the value 'FF' sent by: • A master while retrieving a slave frame and not sending any frame.
RQ0703_009	7.3.1	"NSD" shall consist of idle bytes set to the value 'FF' sent by: • A slave while retrieving a frame from master and not sending any frame.
RQ0703_010	7.3.1	The LPDU Length value shall be compliant with the values indicated in ETSI TS 103 713 [1], table 7.2.
RQ0703_011	7.3.1	Master frames shall always start aligned on the first bytes transmitted on SPI_MOSI and SPI_MISO at SPI_CLK start.
RQ0703_012	7.3.1	Slave frames or remaining bytes of a slave frame (i.e. in a second SPI access for retrieving a slave frame) shall always start aligned on the first bytes transmitted on SPI_MOSI and SPI_MISO at SPI_CLK start.
7.3.2 - Frames generation and transfer rules		
RQ0703_013	7.3.2	The SPI master initiates an SPI access either to send a frame, retrieve a frame from the slave after a MAC access request or both.
RQ0703_014	7.3.2	If the SPI master has a frame to send, the SPI master shall send that frame in a single SPI access, however the SPI master may initiate a SPI access with a length higher than the length of the frame to send.
RQ0703_015	7.3.2	In case the SPI access is longer than the length of the frame being send, SPI master and/or slave shall add Non-Significant Data (NSD) bytes following the CRC until the end of the SPI access.
RQ0703_016	7.3.2	A slave frame shall be retrieved in at most two SPI accesses if the number of bytes of the first SPI access is shorter than the slave frame.
RQ0703_017	7.3.2	If the SPI master did not receive the entire slave frame in one SPI access, the master shall initiate a second SPI access with a length equal or greater than the number of remaining bytes of the slave frame to be retrieved from the slave.
RQ0703_018	7.3.2	In the second SPI access, the slave shall continue to send the same frame from the point where the previous SPI access stopped. The remaining part of a slave frame retrieved in a second access shall start on the first byte of the second access with the byte following the last byte retrieved in the prior access.
RQ0703_019	7.3.2	Master shall send only NSD bytes (i.e. bytes set to the value 'FF') during the second SPI access for retrieving the remaining bytes of a slave frame.
RQ0703_020	7.3.2	The length byte of any frame shall always be the first byte sent in an SPI access, i.e. a new frame shall not be started in the same SPI access.
7.3.3 - Data transfer cases		
RQ0703_021	7.3.3	Any frame sent by master or slave shall be preceded by a MAC phase issued respectively by the master or slave.
RQ0703_022	7.3.3	When two accesses are required for transferring a slave frame, master shall generate the second access at any time greater than or equal to the tCS value in ETSI TS 103 713 [1], clause 6.4.2.
RQ0703_023	7.3.3	Case 1: master initiates the MAC phase and then sends a frame. SPI access length is determined by the master frame length. No data is received from the slave.
RQ0703_024	7.3.3	Case 2: slave initiates a slave MAC access request to transfer a frame. Master performs a first access to retrieve the slave frame length followed by a second access to retrieve the remaining bytes of the slave frame considering the length information from the first access.
RQ0703_025	7.3.3	Case 3: slave initiates a MAC access request for sending a frame. Master generates an access with length equal to MTU to make sure the slave frame is transferred in a single access. Slave frame is shorter than the access length and slave appends NSD bytes after the end of its frame until the end of the access.

Req.ID	Clause	Description
RQ0703_026	7.3.3	Case 4: slave initiates a slave MAC access request for sending a frame. Consequently, master generates an access with length based on a best estimate. Master retrieves only part of the frame during the first access and will generate a second access to retrieve the remaining bytes of the slave frame. The length of the second access is based on the frame length information retrieved in the prior access. The total number of bytes transferred on MISO over both accesses is less than or equal to the MTU.
RQ0703_027	7.3.3	Case 5: both master and slave have frames to transfer and MAC phase is initiated by both simultaneously. Master generates an access with length determined by its frame length. Master finds out that only a part of a slave frame was received and generates a second access to retrieve the remaining bytes of the slave frame. The total number of bytes transferred on MISO over both accesses is less than or equal to the MTU.
RQ0703_028	7.3.3	Case 6: both master and slave have frames to transfer and both initiate the MAC phase simultaneously. Master generates an access with length determined by its frame length. As the slave frame is shorter than the master frame, slave adds NSD bytes after the end of its frame up to access end. The SPI access length is more than or equal to the MTU.
RQ0703_029	7.3.3	Case 7: both master and slave have frames to transfer and both initiate the MAC phase simultaneously. Master generates an access with the length equal to MTU to receive any slave frame occurring at the same time within a single access. Both master and slave may append NSD bytes after the end of their frames, up to access completion.

5.2.3 LLC layers

Reference: ETSI TS 103 713 [1], clause 7.4.

Req.ID	Clause	Description
RQ0704_001	7.4	Support of SHDLC layer as defined in ETSI TS 102 613 [5], clause 10, is mandatory for the master and the slave. The first byte of the LPDU is defined in table 7.3 of ETSI TS 103 713 [1].
RQ0704_002	7.4	Support of CLT layer as defined in ETSI TS 102 613 [5], clause 11, is optional for the master and the slave. The first byte of the LPDU is defined in table 7.3 of ETSI TS 103 713 [1].
RQ0704_003	7.4	Support of MCT layer is mandatory for the master and the slave. The first byte of the LPDU is defined in table 7.3 of ETSI TS 103 713 [1].
RQ0704_004	7.4	The LPDUs shall be structured according to ETSI TS 103 713 [1], figures 7.9, 7.10 or 7.11, depending on the frame type.

5.2.4 Interworking of the LLC layers

Reference: ETSI TS 103 713 [1], clause 7.5.

Req.ID	Clause	Description
RQ0705_001	7.5	After MAC activation, the SHDLC link shall not be established and no CLT session shall be open. Only the MCT LLC shall be used by the master and by the slave for the SPI interface initialization.
RQ0705_002	7.5	The master shall take the following action after a successful MCT LLC phase: • If the master has data to be sent to the slave (e.g. due to a contactless transaction) that requires the use of the CLT LLC, it shall initiate a CLT LLC session.
RQ0705_003	7.5	The master shall take the following action after a successful MCT LLC phase: • If the master has data to be sent to the slave, it shall start the establishment of an SHDLC link as soon as possible.
RQ0705_004	7.5	After the slave and the master have established the SHDLC link or opened the CLT session, the slave and the master shall not send MCT LLC frames; received MCT LLC frames shall be ignored.
RQ0705_005	7.5	To enter the SHDLC LLC for the first time after MCT LLC, the link establishment procedure as described in ETSI TS 103 713 [1], clause 7.7.1 shall apply.
RQ0705_006	7.5	Once the SHDLC link is established, a CLT session shall not invalidate the SHDLC context and the endpoint capabilities negotiated during the SHDLC link establishment.
RQ0705_007	7.5	To enter the CLT LLC from MCT LLC or SHDLC LLC, the CLT session shall be opened as described in clause 11.6 of ETSI TS 102 613 [5].
RQ0705_008	7.5	The master shall open a CLT session only when all SHDLC I-Frames are acknowledged. SHDLC LLC frames received by the slave or by the master during a CLT session close the CLT session.

Req.ID	Clause	Description
RQ0705_009	7.5	SHDLC LLC frames received by the slave or by the master during a CLT session close the CLT session.
RQ0705_010	7.5	In case the slave or the master receives a corrupted frame, then the receiving entity shall use the error recovery procedure defined for the LLC of the last correctly received frame. Immediately after MAC activation, the error handling of the MCT LLC shall apply.
RQ0705_004	7.5	After the slave and the master have established the SHDLC link or opened the CLT session, the slave and the master shall not send MCT LLC frames; received MCT LLC frames shall be ignored.

5.2.5 MCT LLC definition

Reference: ETSI TS 103 713 [1], clause 7.6.

Req.ID	Clause	Description
7.6.1 - MCT LPDU structure		
RQ0706_001	7.6.1	The MCT LPDU shall be structured according to ETSI TS 103 713 [1], figure 7.12. The meaning of MCT_CTRL and MCT_DATA is given in ETSI TS 103 713 [1], table 7.4.
RQ0706_002	7.6.1	The MCT LPDU length shall be lower than or equal to 29 bytes.
7.6.2 - MCT_DATA from master		
RQ0706_003	7.6.2	Master-specific MCT_DATA field shall be defined according to ETSI TS 103 713 [1], table 7.5.
RQ0706_004	7.6.2	After the SPI activation as defined in ETSI TS 103 713 [1] clause 7.2.3.4 or in clause 7.2.4.6, the master shall send the MCT_MASTER_REQ frame and the slave shall respond with the MCT_READY frame.
RQ0706_005	7.6.2	The MCT phase shall be performed with default SPI_CLK = 1 MHz and T1 ≥ 255 µs.
RQ0706_006	7.6.2	Slave shall start the MCT phase in Low Power Mode following VDD ON.
RQ0706_007	7.6.2	The MTU negotiation between the master and slave shall be between the MTU sent by the master in the MCT_MASTER_REQ frame and the MTU sent by the slave in the MCT_READY. The lower of the MTU values will be used by both master and slave for all frames.
RQ0706_008	7.6.2	Master indicates in MCT_MASTER_REQ its power source availability (i.e. Low Power, Full Power Mode 1, Full Power Mode 2 or Full Power Mode 3).
RQ0706_009	7.6.2	Master indicates in MCT_MASTER_REQ its power source availability (i.e. Low Power, Full Power Mode 1, Full Power Mode 2 or Full Power Mode 3). Slave shall support at least Low Power Mode.
RQ0706_010	7.6.2	Master indicates in MCT_MASTER_REQ its power source availability (i.e. Low Power, Full Power Mode 1, Full Power Mode 2 or Full Power Mode 3). Slave shall support at least Full Power Mode 1.
RQ0706_011	7.6.2	Master indicates in MCT_MASTER_REQ its power source availability (i.e. Low Power, Full Power Mode 1, Full Power Mode 2 or Full Power Mode 3). Slave shall be able to limit its maximum current according to power source capabilities (for Low Power Mode and Full Power Mode 1).
7.6.3 - MCT_DATA from slave		
RQ0706_012	7.6.3	Slave-specific MCT_DATA field (bytes 0 … 7) shall be configured as defined in ETSI TS 103 713 [1], table 7.8.
RQ0706_013	7.6.3	In case a slave may "self-resume" (e.g. due to activities on another interface) and it has T3 < T1 slave shall report T1 value for T3, defined in ETSI TS 103 713 [1], table 7.8.
RQ0706_014	7.6.3	Slave capabilities indication in MCT_DATA field byte 0 shall be configured as defined in ETSI TS 103 713 [1], table 7.8.
RQ0706_015	7.6.3	The master shall postpone any transfers to other slaves on the shared bus as long as the slave is performing slave-driven flow control.
7.6.4 - MCT activation procedure		
RQ0706_016	7.6.4	Slave start-up time following power-on is defined as POT and has an initial value of 1 s for the first power on, when the slave reported MCT_READY parameters are not yet available. After POT time (from the time VDD is valid after power-on) slave shall be ready to receive the MCT_MASTER_REQ from master.
RQ0706_017	7.6.4	By MCT activation procedure, master shall use the POT value reported by slave or a higher value in subsequent power-up sequences.
RQ0706_018	7.6.4	Master shall wait for MCT_READY from slave after sending MCT_MASTER_REQ.
RQ0706_019	7.6.4	In case slave did not send MCT_READY response (no MAC access request) within MCT_SLAVE_TIMEOUT or if MCT_READY is corrupted, master shall retry the MCT activation by sending another MCT_MASTER_REQ frame.
RQ0706_020	7.6.4	Master shall retry at least two times i.e. shall re-send MCT_MASTER_REQ at least twice without power toggle.

Req.ID	Clause	Description
RQ0706_021	7.6.4	After power-up, if slave gets a corrupted frame or any other frame instead of the MCT_MASTER_REQ, slave shall discard the data and remain in receive state.
RQ0706_022	7.6.4	MCT_MASTER_TIMEOUT is the maximum time within which the master shall send the MCT_MASTER_REQ after power-on or for the retries in case of errors. The value defined is 1 s.
RQ0706_023	7.6.4	MCT_SLAVE_TIMEOUT is the maximum time within which the slave shall send the MCT_READY response to MCT_MASTER_REQ. The default value defined is 200 ms.

5.2.6 SHDLC LLC definition

5.2.6.1 General SHDLC LLC requirements for SPI implementations

Reference: ETSI TS 103 713 [1], clause 7.7.

Req.ID	Clause	Description
7.7.1 - SHDLC overview		
RQ0707_001	7.7.1	The provisions of ETSI TS 102 613 [5], clause 10.1 shall apply. The SWP SHDLC layer is replaced by the SPI SHDLC layer defined in ETSI TS 103 713 [1]. The SHDLC layer shall ensure that data passed up to the next layer has been received exactly as transmitted i.e. error free, without loss and in the correct order. Also, the SHDLC layer manages the flow control, which ensures that data is transmitted only as fast as the receiver may receive it.
RQ0707_002	7.7.1	The provisions of ETSI TS 102 613 [5] clauses 10.3 to 10.8 shall apply. Additional SHDLC rules are defined in ETSI TS 103 713 [1].
7.7.3 - Flow control		
RQ0707_003	7.7.3.1	Flow control is performed by a transmitter in order to avoid corruption or loss of data. It consists of methods applied by the transmitter and receiver in order to send a maximum number of SHDLC frames that can be accepted by the receiver, after which it shall stop sending data until the receiver sends at least an acknowledgement (e.g. SHDLC I frame or SHDLC S-frame) for one of the received SHDLC frames.
RQ0707_004	7.7.3.2	In addition to the provisions of in ETSI TS 103 713 [1], clause 7.3.2, the total number of bytes transferred on SPI_MOSI while retrieving a slave frame over 2 accesses shall be less than the maximum slave frame length i.e. MTU or a window size slot depth.

5.2.6.2 Specific SHDLC LLC requirements for SPI implementations

Reference: ETSI TS 102 613 [5], clause 10 and subclauses.

Req.ID	Clause	Description
10 - SHDLC LLC definition		
10.1 - SHDLC overview		
RX1001_001	10.1	The SHDLC layer in an endpoint shall ensure that data passed up to the next layer has been received exactly as transmitted (i.e. error free, without loss and in the correct order).
RX1001_002	10.1	If an endpoint receives a corrupted frame, it shall discard the frame.
10.4 - Control Field		
RX1004_001	10.4	An endpoint's default size of sliding window shall be four frames.
10.4.2 - S-Frames coding		
RX1004_002	10.4.2	An endpoint shall not send a S-frame with an information field.
RX1004_003	10.4.2	An SREJ shall be transmitted for each erroneous frame; each frame is treated as a separate error.
RX1004_004	10.4.2	Only one SREJ shall remain outstanding on each link direction at any one time.
RX1004_005	10.4.2	Optional type of frame shall not be used before capability negotiation is defined during initialization.
10.4.2 - U-Frames coding		
RX1004_006	10.4.3	An endpoint shall only send U-Frames using modifiers specified in table 10.3 of ETSI TS 102 613 [5].
10.5 - Changing sliding window size and endpoint capabilities		
10.5.0 - Capabilities negotiation		
RX1005_001	10.5.0	If the initial sliding window size is too large or SREJ support is requested and the receiving endpoint cannot handle (at least one) of those features, it shall not acknowledge the RSET frame. Instead, the receiver shall send a RSET frame with an appropriate sliding window size and/or SREJ frame support bit.

Req.ID	Clause	Description
RX1005_002	10.5.0	An endpoint shall obey to window size reconfiguration and/or SREJ support if the requested window size is lower than its default configuration or the peer endpoint does not support SREJ frames.
RX1005_003	10.5.0	If one or more of the indicated endpoint capabilities are not supported by the receiving endpoint, it shall answer with a RSET frame indicating only the supported endpoint capabilities. In this case the RSET response may contain the same window size.
RX1005_004	10.5.0	An endpoint shall not send RSET frames with upper layer protocols bits set (i.e. bit 2 and bit 3 according to the table 10.4 of ETSI TS 102 613 [5] for upper layer protocol negotiation, if the other endpoint has not indicated support of the SHDLC upper layer protocol negotiation in the ACT_INFORMATION field
RX1005_005	10.5.0	If several upper layer protocols are indicated as supported, the receiving endpoint shall not acknowledge the RSET frame and instead it shall send a RSET frame with only one protocol from the received list of supported protocols. Otherwise, the receiving endpoint shall accept the selected protocol with a UA frame.
RX1005_006	10.5.0	If both endpoints exchange simultaneously RSET frames with multiple supported upper layer protocols, the common supported protocol with the lowest bit number indicated in table 10.4 of ETSI TS 102 613 [5] shall be selected.
RX1005_007	10.5.0	If only one indicated upper layer protocol is indicated and supported by the receiving endpoint, the receiving endpoint shall accept the selection of the upper layer protocol and it shall acknowledge the selected upper layer protocol with a UA frame.
RX1005_008	10.5.0	An endpoint shall not acknowledge a RSET frame with RFU bits set to 1. Instead the receiver shall send a RSET with appropriate parameters e.g. sliding window size, endpoint capabilities and upper layer supported protocols with RFU bits unset.
RX1005_009	10.5.0	During SHDLC link establishment, the window size, selective reject support and the upper layer protocol shall be negotiated at the same time, i.e. an endpoint negotiating the SHDLC link shall provide in its RSET answer (if any) all requested modifications including window size and endpoint capabilities.
10.5.1 - RSET frame payload		
RX1005_010	10.5.1	The number provided for the endpoint sliding window size shall be between 2 to 4 inclusive.
RX1005_011	10.5.1	In case this RSET frame is sent in response to a received RSET frame, the window size value shall be equal or lower than the previously provided value.
RX1005_012	10.5.1	If an RSET frame is received without the second optional byte the default value of SREJ not supported should be used.
RX1005_013	10.5.1	A RSET frame response shall not indicate the same window size and the same endpoint capabilities as the received RSET frame; in such a case a UA frame shall be sent.
10.5.2 - UA frame payload		
RX1005_014	10.5.2	The endpoint shall not include a payload in UA frames.
10.6 - SHDLC context		
10.6.1 - Constants		
RX1006_001	10.6.1	An endpoint shall retry to setup link if the targeted endpoint did not answer with a UA or a RSET frame to a RSET frame within T3 (5 ms).
RX1006_002	10.6.1	I-frames shall be acknowledged within T1.
RX1006_003	10.6.1	If the I-frames are not acknowledged, an endpoint shall retransmit these frames not sooner than T2.
10.6.2 - Variables		
RX1006_004	10.6.2	An endpoint shall increment its value of the N(S) field after emission of an I-Frame.
RX1006_005	10.6.2	N(R) shall be set as described in ETSI TS 102 613 [5].
RX1006_006	10.6.2	During full duplex data transmission or by emission of a S type frame, all received frames with a sequence number lower than N(R) are acknowledged.
10.6.3 - Initial Reset State		
RX1006_007	10.6.3	The following initial states shall apply in every endpoint after successful link establishment: N(S) = N(R) = DN(R) = 0.
10.7 - SHDLC sequence of frames		
10.7.2 - Link establishment with default sliding window size		
RX1007_001	10.7.2	An endpoint establishing an SHDLC link shall initiate link establishment by sending a RSET frame.
RX1007_002	10.7.2	If an endpoint supports the sliding window size and SREJ value in the RSET frame, it shall acknowledge that frame with a UA frame.
RX1007_003	10.7.2	An endpoint receiving a RSET frame without window size and/or endpoint capabilities field shall interpret the RSET frame as if it contained the default values.
RX1007_004	10.7.2	Before link establishment, all SHDLC frames except RSET from other endpoint shall be discarded.
RX1007_005	10.7.2	If the link is re-established, all buffered frames (received out of order or stored in the retransmission queue) shall be discarded.
RX1007_006	10.7.2	If the link is re-established, an endpoint shall inform the upper layer of a link reset.
RX1007_007	10.7.2	An endpoint shall support a link re-establishment which is initiated by the peer endpoint.

Req.ID	Clause	Description
10.7.4 - Data flow		
RX1007_008	10.7.4	Once the link is established, an endpoint shall be able to receive data.
RX1007_009	10.7.4	An endpoint shall acknowledge frame reception regularly.
RX1007_010	10.7.4	The acknowledgement timeout shall not be too long.
RX1007_011	10.7.4	If the number of unacknowledged I-frames on the link equals the negotiated window size, then the endpoint shall not transmit any further I-frames until reception of an acknowledgement.
10.7.5 - Reject (go N back)		
RX1007_012	10.7.5	If an endpoint detects missing I-frame sequence numbers and if SREJ is not supported or if several frames got lost, the endpoint shall send a REJ frame as soon as possible.
RX1007_013	10.7.5	When an endpoint receives a REJ frame with a sequence number which identifies an unacknowledged I-frame previously sent within the sliding window size it shall restart the stream at the first missing frame.
RX1007_014	10.7.5	After sending REJ, an endpoint shall accept the peer endpoint restarting the stream at the first missing frame.
10.7.6 - Last Frame loss		
RX1007_015	10.7.6	Each frame shall have a guarding/transmit timeout in order to retransmit frames if the destination does not notice a loss.
10.7.7 - Receive and not ready		
RX1007_016	10.7.7	When an endpoint transmits a RNR and is now ready to receive an I-Frame, it shall send a RR frame every 5 ms to 20 ms until it receives a new I-frame.
RX1007_017	10.7.7	If an endpoint receives a RR in a context described in RX1007_0016 and has no data to send, it shall send an I-Frame with empty information field to signal the proper reception of the RR frame.
RX1007_018	10.7.7	If an endpoint receives RNR frame then it shall suspend transmission of I-frames within the negotiated WS.
RX1007_019	10.7.7	If an endpoint receives a RR in a context described in RX1007_0016 and still has data to send, it shall resume the I-Frame(s) transmission.
10.7.8 - Selective reject		
RX1007_020	10.7.8	If an endpoint receives a SREJ frame and supports for SREJ was agreed at link establishment, it shall retransmit the corresponding I-Frame.
10.8 - Implementation model		
10.8.2 - Information Frame reception		
RX1008_001	10.8.2	If an I-frame (x,y) is received by an endpoint and support for Selective Reject S frames was negotiated for the link and X is exactly one higher than N(R), a SREJn(r) shall be sent instead of the REJn(r). The received I-frame shall be buffered.
RX1008_002	10.8.2	Once the retransmitted I-frame with X = N(R) is received in the context of RX1007_0020, the buffered I-frame shall also be processed.
RX1006_003	10.8.2	N(R) shall be set as described in ETSI TS 102 613 [5].

5.2.7 Power management

Reference: ETSI TS 103 713 [1], clause 7.8.

Req.ID	Clause	Description
7.8.1 - Power saving mode		
RQ0708_001	7.8.1	The master and the slave shall both resume from power saving mode in the same LLC context following the master or the slave resumption.
7.8.2 - Conditions for entering power saving mode		
RQ0708_002	7.8.2.1	The slave shall not enter into power saving mode, if the slave has issued a MAC access request and is waiting for data transfer from the master.
RQ0708_003	7.8.2.1	The inactivity period T4 is negotiated by the master and the slave at interface initialization as described in clause 7.6. The master shall provide an inactivity period in MCT_MASTER_REQ and the slave shall send back an MCT_READY with the same T4 value for acceptance, or a different value if it cannot support the value received from the master.
RQ0708_004	7.8.2.1	If the master sends T4 = 'FFFF' in MCT_MASTER_REQ, the slave shall disable the entering power saving mode on detection of an inactivity period and the slave shall send an MCT_READY with the same T4 value.
RQ0708_005	7.8.2.1	A slave which supports a resume time T3 lower than T1 and is either not able to systematically enter into power saving mode (i.e. it may remain active) or may resume independently of the activity on the SPI interface, shall report in MCT_READY a T3 value equal to T1 (or higher).

Req.ID	Clause	Description
RQ0708_006	7.8.2.1	In power saving mode, the slave shall maintain its SPI interface as for the case when SPI_NSS is de-asserted with the slave not in power saving mode.
RQ0708_007	7.8.2.2	The slave power source status and capabilities indicated by the master at interface initialization shall not change when the master enters into power saving mode.
RQ0708_008	7.8.2.2	The slave power source status and capabilities indicated by the master at interface initialization shall not change when the master is in power saving mode.
RQ0708_009	7.8.2.2	The slave power source status and capabilities indicated by the master at interface initialization shall not change when the master is resuming from power saving mode.
RQ0708_010	7.8.2.2	The slave power source status and capabilities indicated by the master at interface initialization shall not change after the master has resumed from the power saving mode.
RQ0708_011	7.8.2.2	SPI_NSS shall be maintained de-asserted by the master when the master is in power saving mode and when the master is resuming.
7.8.3 - Resuming from power saving mode		
RQ0708_012	7.8.3.1	Resuming the slave from power saving mode shall be performed by the master when any of the conditions above for the slave to enter into power saving mode has been previously met. The leading edge of the SPI_NSS assertion shall trigger the slave to resume.
RQ0708_013	7.8.3.1	The master shall ensure that all signals it drives are in the idle state corresponding to SPI mode 0 before initiating the resuming of the slave.
RQ0708_014	7.8.3.1	"The master shall perform the following procedure to resume the slave: 1) In the case of a 4 signals SPI interface, the master checks the state of the SPI_NSS signal and if it is de asserted, it goes to the step 2, otherwise loops on step 1. In the case of a 5 signals SPI interface, the master starts with step 2. 2) The master asserts SPI_NSS and waits for at least T3, then goes to step 3. 3) The master considers the slave as resumed from the power saving mode and starts the data transfer. The slave shall support the resumption up to the data transfer phase with SPI_NSS continuously asserted by the master during T3."
RQ0708_015	7.8.3.2	If the master has entered into power saving mode, it shall resume when the slave initiates a MAC access request.
RQ0708_016	7.8.3.2	The slave initiates the MAC access request with the procedures as described in ETSI TS 103 713 [1], clause 7, regardless the power management status of the master.
RQ0708_017	7.8.3.2	Following a resume by a MAC access request during T2 as described in ETSI TS 103 713 [1], clause 7.2.2.3, the master shall start an SPI access after T1 or later, from the leading edge of the slave MAC access request pulse, i.e. ETSI TS 103 713 [1], clauses 7.2.3.2 and 7.2.4.3 apply.

6 Test cases for electrical interfaces

6.0 Initial conditions for tests of the electrical interfaces

6.0.1 Common initial conditions for tests of the electrical interfaces

The Test Tool (TT) is connected to the signal lines of the SPI.

6.0.2 Pre-conditions for the measurement of DC characteristics

For the measurement of the DC characteristics:

- when measuring the Output high voltage, the Output high current (IOH) is limited to -100 µA;
- when measuring the Output low voltage, the Output low current (IOL) is limited to 1 mA;
- the test environment shown in Figure 4.1 is used, where the SPI master is the SUT;
- the TT Connector for a 5 signal SPI supports the testing architecture shown in Figure 4.3;
- the TT Connector for a 4 signal SPI supports the testing architecture shown in Figure 4.4.

NOTE 1: Measurements of signal voltage levels can be done at any time VDD has breached the minimum voltage threshold for voltage class to be measured in and whenever VDD is deactivated by the master.

NOTE 2: The ground reference for all measurements is VSS of the power supply.

NOTE 3: Currents flowing into the slave are considered positive.

6.0.3 Pre-conditions for the measurement of AC characteristics

For the measurement of the AC characteristics:

- The SPI mode 0 has to be supported and used for all measurements of AC characteristics (CPOL = 0 and CPHA = 0).
- As for testing purposes control and signalling on the SPI bus has to be initiated by a specified entity, the related AC testing has to be executed using the appropriate test environment shown in clause 4:
  - Testing of SPI_CLK, SPI_MOSI and SPI_NSS for either 4 or 5 signals SPI is done using the test environment shown in Figure 4.1.
  - Testing of SPI_MISO in either 4 or 5 signals SPI is done using the test environment shown in Figure 4.2.
  - Testing of SPI_NSS, slave driven in a 4 signals SPI is done using the test environment shown in Figure 4.2.
  - Testing SPI_INT in a 5 signals SPI is done using the test environment shown in Figure 4.2.
- The TT Connector for a 5 signals SPI supports the testing architecture shown in Figure 4.3.
- The TT Connector for a 4 signals SPI supports the testing architecture shown in Figure 4.4.

The test cases are defined to reflect these requirements.

To get the SPI to operational state using timings and parameters as negotiated during initial MAC activation the following procedure shall be executed using the default values in accordance to ETSI TS 103 713 [1]:

POT: 1 s

SPI_CLK: 1 MHz

T1: 255 µs

T2: 1 µs

6.0.4 Preparation procedure - SPI master AC testing

For SPI master testing, a test environment where the SPI master is the SUT and the TT is emulating an SPI slave is used (see Figure 4.1).

Step	Direction	Action/Task	Description/Expectation
0.1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on
0.2	SPI_M → TT(SPI_S)	Run MAC activation	A 5 signals SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 A 4 signals SPI_M runs a MAC activation as defined in Annex C, clause C.2.1
0.3	SPI_M → TT(SPI_S)	Initiate a data transfer for the MAC phase and send MCT_MASTER_REQ	The MCT_MASTER_REQ frame is received by the SPI_S after the initial POT (1 s)
	TT	Verification of MCT_MASTER_REQ data	The MCT_MASTER_REQ data is verified and stored in the TT
0.4	TT(SPI_S) → SPI_M	Return MCT_READY_DEF	The TT sends an MCT_READY with the data defined in MCT_READY_DEF
0.5	SPI_M	Run MAC deactivation	A 5 signals SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 A 4 signals SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2

6.0.5 Preparation procedure - SPI slave AC testing

For SPI slave testing, a test environment where the SPI slave is the SUT and the TT is emulating an SPI master is used (see Figure 4.2).

Step	Direction	Action/Task	Description/Expectation
0.1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on
0.2	TT(SPI_M) → SPI_S	Run MAC activation	The TT(SPI_M) of a 5 signal SPI runs a MAC activation as defined in Annex C, clause C.1.1 The TT(SPI_M) for a 4 signal SPI runs a MAC activation as defined in Annex C, clause C.2.1
0.3	TT(SPI_M) → SPI_S	Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after initial POT (1 s)	The MCT_MASTER_REQ_DEF frame prepared by the TT(SPI_M) is received by the SPI_S
0.4	SPI_S → TT(SPI_M)	Return MCT_READY	The SPI_S sends an MCT_READY
	TT	Verification of MCT_READY data	The MCT_READY data is verified and stored in the TT
0.5	TT(SPI_M) → SPI_S	Run MAC deactivation	The TT(SPI_M) for a 5 signal SPI runs a MAC deactivation as defined in Annex C, clause C.1.2 The TT(SPI_M) for a 4 signal SPI runs a MAC deactivation as defined in Annex C, clause C.2.2

6.0.6 Post-processing procedure

The post-processing procedure for electrical testing shall ensure that all measurements performed by the TT can be completed and that the associated SPI master slave connection under test is securely deactivated and powered down.

Step	Direction	Action/Task	Description/Expectation
n.1	TT	Wait for 10 s	TT and power supply for the SPI is kept active to allow contingently running processes and measurements to stop
n.2	SPI_M → SPI_S	Run MAC deactivation	The SPI_M for a 5 signal SPI runs a MAC deactivation as defined in Annex C, clause C.1.2 The SPI_M for a 4 signal SPI runs a MAC deactivation as defined in Annex C, clause C.2.2
n.3	TT/Tester	Power off	The SPI is powered off, i.e. the power supply of the PCB or chip hosting the SPI is switched off

6.1 Electrical characteristics - 5 signals SPI - SPI Master testing

6.1.1 5 signals SPI - DC characteristics for operational voltage class B

6.1.1.1 Test purpose

The SPI electrical specification interface shall be defined for VDD operational voltage classes B and C as defined in ETSI TS 103 666-1 [4], clause 6.2.2.3. For the SPI physical interface with 5 signals operating in voltage class B (3,0 V) the DC characteristics defined in ETSI TS 103 713 [1], table 6.2 apply, except for the content from rows related to the SPI_NSS signal.

6.1.1.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.2 apply.

2) The technology dependent high impedance values for the input and output buffers connected to the SPI are used to adjust the slave emulation characteristics of the TT.

6.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_018
2	TT	Run voltage measurement on VDD	VDD*1 is within the limits defined for voltage class B (2,7 V - 3,3 V)	RQ0604_001
3	TT	Run voltage measurement on SPI_CLK to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) the voltage measured on the SPI_CLK*2 signal in low state stays within the specified limits (-0,5 V - 0,1 × VDD)	RQ0604_003 RQ0604_005
4	TT	Run voltage measurement on SPI_NSS to determine VOH/VIH	Up to the maximum current defined for a signal in high state (-100 µA) the voltage measured on the SPI_NSS*2 signal in high state stays within the specified limits (0,7 × VDD - VDD + 0,5 V)	RQ0604_002 RQ0604_004
5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	
6	TT	Run voltage measurement on VDD	VDD is deactivated	RQ0604_001

NOTE 1: The voltage on VDD is measured as a reference from power on to power off (for sample rate and measurement uncertainties see Annex A).

NOTE 2: A differentiation between voltages on connected input and output buffers may not be possible (see clause 4.2.1). The lower thresholds given in the reference table apply.

6.1.1.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

6.1.2 5 signals SPI - DC characteristics for operational voltage class C

6.1.2.1 Test purpose

The SPI Electrical specification interface shall be defined for VDD operational voltage classes B and C as defined in ETSI TS 103 666-1 [4], clause 6.2.2.3. For the SPI physical interface with 5 signals operating in voltage class C (1,8 V) the DC characteristics defined in ETSI TS 103 713 [1], table 6.3 apply, except for the content from rows related to the SPI_NSS signal.

6.1.2.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.2 apply.

2) The technology dependent high impedance values for the input and output buffers connected to the SPI are used to adjust the slave emulation characteristics of the TT.

6.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_018
2	TT	Run voltage measurement on VDD	VDD*1 is within the limits defined for voltage class C (1,68 V - 1,92 V)	RQ0604_001

Step	Direction	Action/Task	Description/Expectation	REQ
3	TT	Run voltage measurement on SPI_CLK to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) the voltage measured on the SPI_CLK*2 signal in low state stays within the specified limits (-0,3 V - 0,1 × VDD)	RQ0604_009 RQ0604_011
4	TT	Run voltage measurement on SPI_NSS to determine VOH/VIH	Up to the maximum current defined for a signal in high state (-100 µA) the voltage measured on the SPI_NSS*2 signal in high state stays within the specified limits (0,7 × VDD - VDD + 0,3 V)	RQ0604_008 RQ0604_010
5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	
6	TT	Run voltage measurement on VDD	VDD is deactivated	RQ0604_001

NOTE 1: The voltage on VDD is measured as a reference from power on to power off (for sample rate and measurement uncertainties see Annex A).

NOTE 2: A differentiation between voltages on connected input and output buffers may not be possible (see clause 4.2.1). The lower thresholds given in the reference table apply.

6.1.2.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

6.1.3 5 signals SPI - AC characteristics for operational voltage class B

6.1.3.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI master.

To allow determination of AC characteristics for master driven signals the SPI timing diagram from ETSI TS 103 713 [1], Figure 6.3 is adapted as shown in figure 6.1.

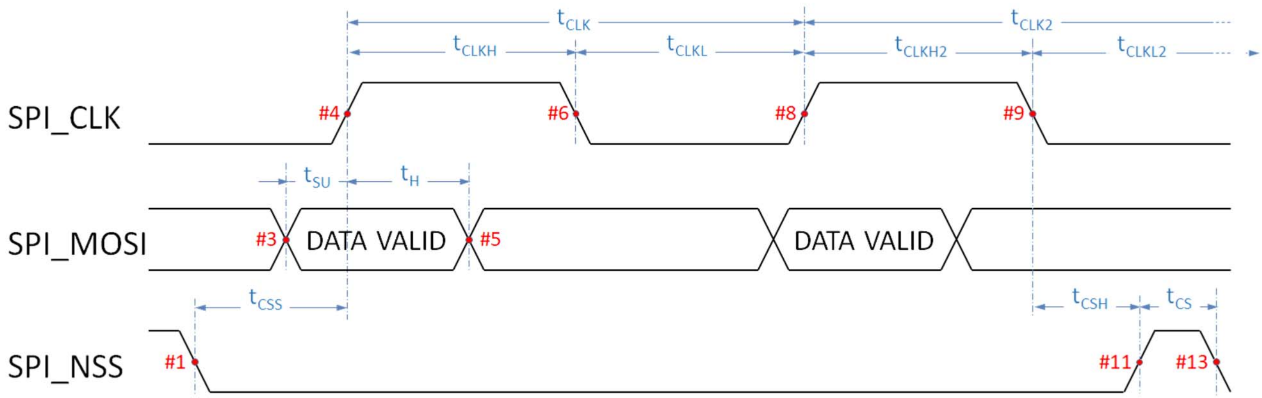

Figure 6.1: SPI timing diagram for master driven signals

This test focuses on AC characteristics for master driven signals operated in voltage class B (3,0 V).

6.1.3.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI master AC testing from clause 6.0.4 is executed.

3) The SPI slave asserts SPI_INT for a 1 µs pulse (or known duration T2) for a MAC access request.

6.1.3.3 Test procedure

Sequence 1 - Determination of clock specific AC characteristics

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS	
3	SPI_M → TT(SPI_S)	Initiate data transfer	The SPI_M provides valid data on SPI_MOSI	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at the first rising edge on SPI_CLK	
5	SPI_M → TT(SPI_S)	End providing valid data	The SPI_M stops providing valid data on SPI_MOSI	
6	SPI_M → TT(SPI_S)	Send falling edge on SPI_CLK	SPI_CLK is switched to low level The TT stores time stamp #6 for the falling edge on SPI_CLK	
	TT	Determine*[1] tCLKH	The TT determines and stores tCLKH tCLKH is larger or equal to 0,45 × tCLK	RQ0604_017
7	SPI_M → TT(SPI_S)	Continue data transfer	The SPI_M provides valid data on SPI_MOSI	
8	SPI_M → TT(SPI_S)	Send rising edge on SPI_CLK	SPI_CLK is switched to high level The TT stores time stamp #8 for the rising edge on SPI_CLK	
	TT	Determine*[1] tCLKL	The TT determines and stores tCLKL tCLKL is larger or equal to 0,45 × tCLK	RQ0604_018
	TT	Determine*[1] tCLK	The TT determines and stores tCLK The SPI_CLK frequency is lower or equal to the maximum frequent specified by the SPI_S in MCT_READY	RQ0604_016
9	SPI_M → TT(SPI_S)	Repeat step 5 to step 8 as long as data is available on SPI_M	tCLK, tCLKH and tCLKL are determined for the complete period the SPI_M is providing a clock signal	
10	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK	

NOTE 1: The measurement of tCLKL and tCLKH on SPI_CLK is continued for the time a clock signal is provided on SPI_ClK. For step 5 to step 8 the time stamps generated on rising and falling edges of SPI_CLK are not explicitly numbered, but are handled similar to the time stamps generated for the first clock period.

Sequence 2 - Determination of assertion related AC characteristics

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS The TT stores time stamp #1	
3	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at the time of the first rising edge on SPI_CLK	

Step	Direction	Action/Task	Description/Expectation	REQ
	TT	Determine tCSS	The time between time stamp #1 and time stamp #4 is determined (tCSS ≥ 33 ns)	RQ0604_023
5	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI	
6	SPI_M → TT(SPI_S)	Continue data provisioning	All data that shall be sent during this MAC activation is sent.	
7	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK The TT stores time stamp #9	
8	SPI_M → TT(SPI_S)	De-assert SPI_NSS	The SPI_M de-asserts SPI_NSS The TT stores time stamp #11	
	TT	Determine tCSH	The time between time stamp #9 and time stamp #11 is determined (tCSH ≥ 0,5 × tCLK)	RQ0604_024
9	TT	Measurement of SPI_NSS for at least 30 ns	SPI_NSS is not asserted within tCS (tCS ≥ 30 ns)	RQ0604_026

Sequence 3 - Determination of AC characteristic related to data transfer

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS	
3	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI The TT stores time stamp #3	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at the time of the first rising edge on SPI_CLK	
	TT	Determine*[2] tSU	The time between time stamp #3 and time stamp #4 is determined (tSU ≥ 5 ns)	RQ0604_019
5	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI The TT stores time stamp #5	
	TT	Determine*[2] tH	The time between time stamp #4 and time stamp #5 is determined (tH ≥ 3 ns)	RQ0604_020
6	SPI_M → TT(SPI_S)	Send falling edge on SPI_CLK		
7	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI The TT stores a time stamp	
8	SPI_M → TT(SPI_S)	Send rising edge on SPI_CLK	The TT stores a time stamp	
	TT	Determine*[2] tSU	The time between the last two stored time stamps is determined (tSU ≥ 5 ns)	RQ0604_019
9	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI The TT stores a time stamp	
	TT	Determine*[2] tH	The time between the last two stored time stamps is determined (tH ≥ 3 ns)	RQ0604_020
10	SPI_M → TT(SPI_S)	Repeat step 6 to step 9 as long as data is available on SPI_M	tSU and tH are determined for the complete period the SPI_M is providing data and a clock signal	
11	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK	

NOTE 2: The measurement of tSU and tH is continued for the time data and a clock signal are provided. For step 6 to step 9 the time stamps generated on rising edges of SPI_CLK and when data provisioning starts and ends on SPI_MOSI are not explicitly numbered, but are handled similar to the time stamps generated for the first data transfer.

6.1.3.4 Post condition

All sequences of this test case require the post-processing procedure defined in clause 6.0.6 to be executed.

6.1.4 5 signals SPI - AC characteristics for operational voltage class C

6.1.4.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI master.

For determination of AC characteristics of master driven signals, the specific SPI timing diagram shown in Figure 6.1 is used.

This test focuses on AC characteristics for master driven signals operated in voltage class C (1,8 V).

6.1.4.2 Initial conditions

The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply:

1) The preparation procedure for SPI master AC testing from clause 6.0.4 is executed.

2) The SPI slave asserts SPI_INT for a 1 µs pulse (or known duration T2) for a MAC access request.

6.1.4.3 Test procedure

Sequence 1 - Determination of clock specific AC characteristics

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS	
3	SPI_M → TT(SPI_S)	Initiate data transfer	The SPI_M provides valid data on SPI_MOSI	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at the time of the first rising edge on SPI_CLK	
5	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI	
6	SPI_M → TT(SPI_S)	Send falling edge on SPI_CLK	SPI_CLK is switched to low level The TT stores time stamp #6 for the falling edge on SPI_CLK	
	TT	Determine*[1] tCLKH	The TT determines and stores tCLKH tCLKH is larger or equal to 0,45 × tCLK	RQ0604_017
7	SPI_M → TT(SPI_S)	Continue data transfer	The SPI_M provides valid data on SPI_MOSI	

Step	Direction	Action/Task	Description/Expectation	REQ
8	SPI_M → TT(SPI_S)	Send rising edge on SPI_CLK	SPI_CLK is switched to high level The TT stores time stamp #8 at the time of the rising edge on SPI_CLK	
	TT	Determine*[1] tCLKL	The TT determines and stores tCLKL tCLKL is larger or equal to 0,45 × tCLK	RQ0604_018
	TT	Determine*[1] tCLK	The TT determines and stores tCLK The SPI_CLK frequency is lower or equal to the maximum frequency specified by the SPI_S in MCT_READY	RQ0604_016
9	SPI_M → TT(SPI_S)	Repeat step 5 to step 8 as long as data is available on SPI_M	tCLK, tCLKH and tCLKL are determined for the complete period the SPI_M is providing a clock signal	
10	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK	

NOTE 1: The measurement of tCLKL and tCLKH on SPI_CLK is continued for the time a clock signal is provided on SPI_ClK. For step 5 to step 8 the time stamps generated on rising and falling edges of SPI_CLK are not explicitly numbered, but are handled in a similar way to the time stamps generated for the first clock period.

Sequence 2 - Determination of assertion related AC characteristics

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS The TT stores time stamp #1	
3	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at thetime of the first rising edge on SPI_CLK is stored	
	TT	Determine tCSS	The time between time stamp #1 and time stamp #4 is determined (tCSS ≥ 63 ns)	RQ0604_022
5	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI	
6	SPI_M → TT(SPI_S)	Continue data provisioning	All data that shall be sent during this MAC activation is sent.	
7	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK The TT stores time stamp #9	
8	SPI_M → TT(SPI_S)	De-assert SPI_NSS	The SPI_M de-asserts SPI_NSS The TT stores time stamp #11	
	TT	Determine tCSH	The time between time stamp #9 and time stamp #11 is determined (tCSH ≥ 0,5 × tCLK)	RQ0604_024
9	TT	Measurement of SPI_NSS for at least 60 ns	SPI_NSS is not asserted within tCS (tCS ≥ 60 ns)	RQ0604_025

Sequence 3 - Determination of AC characteristic related to data transfer

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	SPI_M → TT(SPI_S)	Assert SPI_NSS	The SPI_M asserts SPI_NSS	

Step	Direction	Action/Task	Description/Expectation	REQ
3	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI The TT stores time stamp #3	
4	SPI_M → TT(SPI_S)	Activation of SPI_CLK	SPI_CLK is activated after at least POT The TT stores time stamp #4 at the time of the first rising edge on SPI_CLK	
	TT	Determine*2 tSU	The time between time stamp #3 and time stamp #4 is determined (tSU ≥ 5 ns)	RQ0604_019
5	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI The TT stores time stamp #5	
	TT	Determine*2 tH	The time between time stamp #4 and time stamp #5 is determined (tH ≥ 3 ns)	RQ0604_020
6	SPI_M → TT(SPI_S)	Send falling edge on SPI_CLK		
7	SPI_M → TT(SPI_S)	Provide valid data on SPI_MOSI	The SPI_M provides valid data on SPI_MOSI The TT stores a time stamp	
8	SPI_M → TT(SPI_S)	Send rising edge on SPI_CLK	The TT stores a time stamp	
	TT	Determine*2 tSU	The time between the last two stored time stamps is determined (tSU ≥ 5 ns)	RQ0604_019
9	SPI_M → TT(SPI_S)	End providing valid data on SPI_MOSI	The SPI_M stops providing valid data on SPI_MOSI The TT stores a time stamp	
	TT	Determine*2 tH	The time between the last two stored time stamps is determined (tH ≥ 3 ns)	RQ0604_020
10	SPI_M → TT(SPI_S)	Repeat step 6 to step 9 as long as data is available on SPI_M	tSU and tH are determined for the complete period the SPI_M is providing data and a clock signal	
11	SPI_M → TT(SPI_S)	Stop SPI_CLK	The SPI_M stops SPI_CLK	

NOTE 2: The measurement of tSU and tH is continued for the time data and a clock signal are provided. For step 6 to step 9 the time stamps generated on rising edges of SPI_CLK and when data provisioning starts and ends on SPI_MOSI are not explicitly numbered, but are handled in a similar way to the time stamps generated for the first data transfer.

6.1.4.4 Post condition

All sequences of this test case require the post-processing procedure defined in clause 6.0.6 to be executed.

6.2 Electrical characteristics - 5 signals SPI - SPI Slave testing

6.2.1 5 signals SPI - Class B, AC characteristics for slave driven signals

6.2.1.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI slave.

To allow determination of AC characteristics for slave driven signals the SPI timing diagram from ETSI TS 103 713 [1], Figure 6.3 is adapted as shown in the figure 6.2.

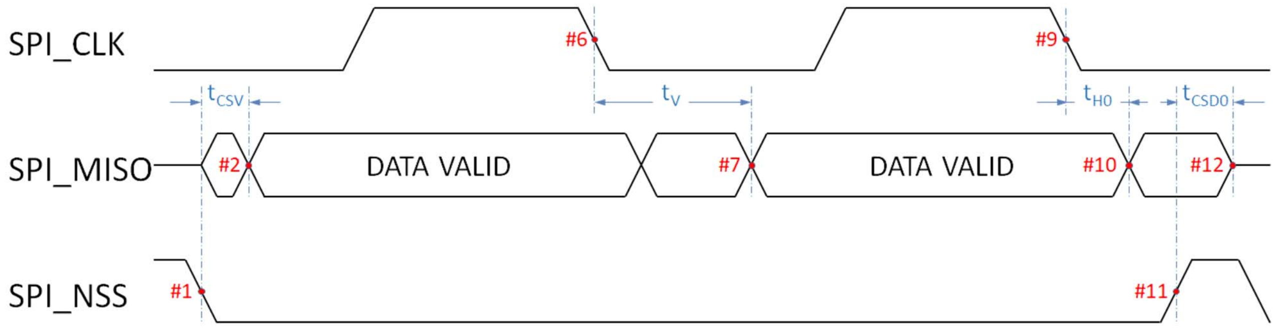

Figure 6.2: SPI timing diagram for slave driven signals

This test focuses on AC characteristics for slave driven signals operated in voltage class B

6.2.1.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI slave AC testing from clause 6.0.5 is executed.

6.2.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_M asserts SPI_NSS The TT stores time stamp #1	
3	SPI_S → TT(SPI_M)	Provide valid data on SPI_MISO	The SPI_S provides valid data on SPI_MISO The TT stores time stamp #2	
	TT	Determine tCSV	The time between time stamp #1 and time stamp #2 is determined (tCSV ≥ 28 ns)	RQ0604_028
5	TT(SPI_M) → SPI_S	Start SPI_CLK	The SPI_M starts SPI_CLK by sending a rising edge on SPI_CLK	
6	TT(SPI_M) → SPI_S	Send falling edge on SPI_CLK	The TT stores time stamp #6	
7	SPI_S → TT(SPI_M)	End providing valid data on SPI_MISO	The TT stores time stamp (equivalent to time stamp #10)	
	TT	Determine tHO	The time between time stamp #6 and time stamp equivalent to #10 is determined (tHO ≥ 0 ns)	RQ0604_021
8	SPI_S → TT(SPI_M)	Provide valid data on SPI_MISO	The SPI_S provides valid data on SPI_MISO The TT stores time stamp #7	
	TT	Determine tV	The time between time stamp #6 and time stamp #7 is determined (tV ≥ 0 ns)	RQ0604_029
9	TT(SPI_M) → SPI_S	Repeat step 6. to step 8. as long as data is available on SPI_S	tHO and tV are determined for the complete period the SPI_S is providing data and SPI_M is providing a clock signal	
10	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_M de-asserts SPI_NSS The TT stores time stamp #11	
11	TT	Measurement of SPI_MISO for at least 30 ns	The SPI_S disables the output on SPI_MISO within tCSDO (tCSDO: 0 ns to ≥ 30 ns)	RQ0604_031

6.2.1.4 Post condition

The post-processing procedure defined in clause 6.0.6 shall be executed.

6.2.2 5 signals SPI - Class C, AC characteristics for slave driven signals

6.2.2.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI slave.

For determination of AC characteristics of slave driven signals, the specific SPI timing diagram shown in Figure 6.2 is used.

This test focuses on AC characteristics for slave driven signals operated in voltage class C (1,8 V).

6.2.2.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI slave AC testing from clause 6.0.5 is executed.

6.2.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 with the values provided by the SPI_S in MCT_READY	RQ0604_015
2	TT(SPI_M)	Assert SPI_NSS	The SPI_M asserts SPI_NSS The TT stores time stamp #1	
3	SPI_S → TT(SPI_M)	Provide valid data on SPI_MISO	The SPI_S provides valid data on SPI_MISO The TT stores time stamp #2	
	TT	Determine tCSV	The time between time stamp #1 and time stamp #2 is determined (tCSV ≥ 58 ns)	RQ0604_027
5	TT(SPI_M) → SPI_S	Start SPI_CLK	The SPI_M starts SPI_CLK by sending a rising edge on SPI_CLK	
6	TT(SPI_M) → SPI_S	Send falling edge on SPI_CLK	The TT stores time stamp #6	
7	SPI_S → TT(SPI_M)	End providing valid data on SPI_MISO	The TT stores time stamp (equivalent to time stamp #10)	
	TT	Determine tHO	The time between time stamp #6 and time stamp equivalent to #10 is determined (tHO ≥ 0 ns)	RQ0604_021
8	SPI_S → TT(SPI_M)	Provide valid data on SPI_MISO	The SPI_S provides valid data on SPI_MISO The TT stores time stamp #7	
	TT	Determine tV	The time between time stamp #6 and time stamp #7 is determined (tV ≥ 0 ns)	RQ0604_029
9	TT(SPI_M) → SPI_S	Repeat step 6. to step 8. as long as data is available on SPI_S	tHO and tV are determined for the complete period the SPI_S is providing data and SPI_M is providing a clock signal	
10	TT(SPI_M)	De-assert SPI_NSS	The SPI_M de-asserts SPI_NSS The TT stores time stamp #11	
11	SPI_S	Measurement of SPI_MISO for at least 60 ns	The SPI_S is disabling the output on SPI_MISO within tCSDO (tCSDO: 0 ns to ≥ 60 ns)	RQ0604_030

6.2.2.4 Post condition

The post-processing procedure defined in clause 6.0.6 shall be executed.

6.3 Electrical characteristics - 4 signals SPI - SPI Master testing

6.3.1 4 signals SPI - DC characteristics for operational voltage class B

6.3.1.1 Test purpose

The SPI Electrical specification interface shall be defined for VDD operational voltage classes B and C as defined in ETSI TS 103 666-1 [4], clause 6.2.2.3. For the SPI physical interface with 4 signals operating in voltage class B the DC characteristics defined in ETSI TS 103 713 [1], table 6.2 apply.

6.3.1.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.2 apply.

2) The technology dependent high impedance values for the input and output buffers connected to the SPI are used to adjust the slave emulation characteristics of the TT.

6.3.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_038
	TT	Run voltage measurement on VDD	VDD*1 is within the limits defined for voltage class B (2,7 V - 3,3 V)	RQ0604_001
2	TT	Run voltage measurement on SPI_CLK to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) *3 the voltage measured on the SPI_CLK*2 signal in low state stays within the specified limits (-0,5 V - 0,1 × VDD)	RQ0604_003 RQ0604_005 RQ0604_006
3	TT	Run voltage measurement on SPI_NSS to determine VOH/VIH	Up to the maximum current defined for a signal in high state (-100 µA) the voltage measured on the SPI_NSS*2 signal in high state stays within the specified limits (0,7 × VDD - VDD + 0,5 V)	RQ0604_002 RQ0604_004
4	SPI_M → TT(SPI_S)	Assert SPI_NSS		
	TT	Run voltage measurement on SPI_NSS to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) the voltage measured on the SPI_NSS*2 signal in low state stays within the specified V limits (-0,5 V - 0,1 × VDD)	RQ0604_005 RQ0604_006
		Run current measurement on SPI_NSS to determine IOL	Measure IOL while VOL is set to 0,3 V (IOLmin: -1 mA)	RQ0604_006
		Determine SPI_NSS related parameters	Determine if the pull-up resistor is correctly dimensioned Determine CI	RQ0604_014*4 RQ0604_007*4
5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	
6	TT	Measurement of VDD	VDD is deactivated	RQ0604_001

NOTE 1: The voltage on VDD is measured as a reference from power on to power off (for sample rate and measurement uncertainties see Annex A).

NOTE 2: A differentiation between voltages on connected input and output buffers may not be possible (see clause 4.2.1). The lower thresholds given in the reference table apply. Thus, requirements from RQ0604_002 and RQ0604_004 cannot be fully reflected.

NOTE 3: The current IOL on SPI_CLK and an asserted SPI_NSS are not verified by the TT (RQ0604_006). Here the TT acts as power sink and adjusts the current to 1 mA.

NOTE 4: A verification of these parameters based on quasi static measurement may be possible. Assumptions and calculations made to gain a result are made visible in the report.

6.3.1.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

6.3.2 4 signals SPI - DC characteristics for operational voltage class C

6.3.2.1 Test purpose

The SPI Electrical specification interface shall be defined for VDD operational voltage classes B and C as defined in ETSI TS 103 666-1 [4], clause 6.2.2.3. For the SPI physical interface with 4 signals operating in voltage class C the DC characteristics defined in ETSI TS 103 713 [1], table 6.3 apply.

6.3.2.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.2 apply.

2) The technology dependent high impedance values for the input and output buffers connected to the SPI are used to adjust the slave emulation characteristics of the TT.

6.3.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_038
	TT	Run voltage measurement on VDD	VDD*[1] is within the limits defined for voltage class C (1,62 V - 1,98 V)	RQ0604_001
2	TT	Run voltage measurement on SPI_CLK to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) *[3] the voltage measured on the SPI_CLK*[2] signal in low state stays within the specified limits (-0,3 V - 0,1 × VDD)	RQ0604_009 RQ0604_011 RQ0604_012
3	TT	Run voltage measurement on SPI_NSS to determine VOH/VIH	Up to the maximum current defined for a signal in high state (-100 µA) the voltage measured on the SPI_NSS*[2] signal in high state stays within the specified limits (0,7 × VDD - VDD + 0,3 V)	RQ0604_008 RQ0604_010

Step	Direction	Action/Task	Description/Expectation	REQ
4	SPI_M → TT(SPI_S)	Assert SPI_NSS		
	TT	Run voltage measurement on SPI_NSS to determine VOL/VIL	Up to the maximum current defined for a signal in low state (1 mA) the voltage measured on the SPI_NSS*[2] signal in low state stays within the specified V limits (-0,3 V - 0,1 × VDD)	RQ0604_009 RQ0604_011 RQ0604_012
		Run current measurement on SPI_NSS to determine IOL	Measure IOL while VOL is set to 0,3 V (IOLmin: -1 mA)	RQ0604_012
		Determine SPI_NSS related parameters	Determine if the pull-up resistor is correctly dimensioned Determine CI	RQ0604_014*[4] RQ0604_013*[4]
5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	
6	TT	Measurement of VDD	VDD is deactivated	RQ0604_001

NOTE 1: The voltage on VDD is measured as a reference from power on to power off (for sample rate and measurement uncertainties see Annex A).

NOTE 2: A differentiation between voltages on connected input and output buffers may not be possible (see clause 4.2.1). The lower thresholds given in the reference table apply. Thus, requirements from RQ0604_008 and RQ0604_010 cannot be fully reflected.

NOTE 3: The current IOL on SPI_CLK and an asserted SPI_NSS are not verified by the TT (RQ0604_012). Here the TT acts as power sink and adjusts the current to 1 mA.

NOTE 4: A verification of these parameters based on quasi static measurement may be possible. Assumptions and calculations made to gain a result are made visible in the report.

6.3.2.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

6.3.3 4 signals SPI - AC characteristics for operational voltage class B

6.3.3.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], Table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI master.

For determination of AC characteristics of master driven signals, the SPI timing diagram shown in Figure 6.1: SPI timing diagram for master driven signals is used.

This test focuses on AC characteristics for master driven signals operated in voltage class B (3,0 V).

6.3.3.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI master AC testing from clause 6.0.4 is executed.

3) The SPI slave asserts SPI_NSS for a 1 µs pulse (or known duration T2) for a MAC access request.

6.3.3.3 Test procedure

Sequence 1 - Determination of clock specific AC characteristics

See clause 6.1.3.3 - Sequence 1. The test procedure is identical for 4 signals and for 5 signals SPI.

Sequence 2 - Determination of assertion related AC characteristics

See clause 6.1.3.3 - Sequence 2. The test procedure is identical for 4 signals and for 5 signals SPI.

Sequence 3 - Determination of AC characteristic related to data transfer

See clause 6.1.3.3 - Sequence 3. The test procedure is identical for 4 signals and for 5 signals SPI.

6.3.3.4 Post condition

All sequences of this test case require the post-processing procedure defined in clause 6.0.6 to be executed.

6.3.4 4 signals SPI - AC characteristics for operational voltage class C

6.3.4.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], Table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI master.

For determination of AC characteristics of master driven signals, the SPI timing diagram shown in Figure 6.1: SPI timing diagram for master driven signals is used.

This test focuses on AC characteristics for master driven signals operated in Class C (1,8 V).

6.3.4.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI master AC testing from clause 6.0.4 is executed.

3) The SPI slave asserts SPI_NSS for a 1 µs pulse (or known duration T2) for a MAC access request.

6.3.4.3 Test procedure

Sequence 1 - Determination of clock specific AC characteristics

See clause 6.1.4.3 - Sequence 1. The test procedure is identical for 4 signals and for 5 signals SPI.

Sequence 2 - Determination of assertion related AC characteristics

See clause 6.1.4.3 - Sequence 2. The test procedure is identical for 4 signals and for 5 signals SPI.

Sequence 3 - Determination of AC characteristic related to data transfer

See clause 6.1.4.3 - Sequence 3. The test procedure is identical for 4 signals and for 5 signals SPI.

6.3.4.4 Post condition

The post-processing procedure defined in clause 6.0.6 shall be executed.

6.4 Electrical characteristics - 4 signals SPI - SPI Slave testing

6.4.1 4 signals SPI - Class B, AC characteristics for slave driven signals

6.4.1.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI slave.

For determination of AC characteristics of slave driven signals, the SPI timing diagram shown in Figure 6.2 is used.

This test focuses on AC characteristics for slave driven signals operated in voltage class B.

6.4.1.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI slave AC testing from clause 6.0.5 is executed.

6.4.1.3 Test procedure

Sequence 1

See clause 6.2.1.3. The test procedure is identical for 4 signals and for 5 signals SPI.

6.4.1.4 Post condition

The post-processing procedure defined in clause 6.0.6 shall be executed.

6.4.2 4 signals SPI - Class C, AC characteristics for slave driven signals

6.4.2.1 Test purpose

To comply to ETSI TS 103 713 [1] an SPI bus shall have implemented the SPI mode 0 according to the industry de-facto SPI specification. Timing parameters indicated in ETSI TS 103 713 [1], table 6.4 are reference values for generic SPI slaves and therefore have to be supported by the SPI slave.

For determination of AC characteristics of slave driven signals, the SPI timing diagram shown in Figure 6.2 is used.

This test focuses on AC characteristics for slave driven signals operated in voltage class C (1,8 V).

6.4.2.2 Initial conditions

1) The initial conditions listed in clause 6.0.1 and clause 6.0.3 apply.

2) The preparation procedure for SPI slave AC testing from clause 6.0.5 is executed.

6.4.2.3 Test procedure

Sequence 1

See clause 6.2.2.3. The test procedure is identical for 4 signals and for 5 signals SPI.

6.4.2.4 Post condition

The post-processing procedure defined in clause 6.0.6 shall be executed.

6.5 Verification of slave states - SPI Slave testing

6.5.0 Explanation of slave states

The slave allows SPI shared bus and states. Thus, the signals SPI_MISO, SPI_MOSI and SPI_CLK may be shared between multiple slaves. However, each slave has a dedicated SPI_NSS.

To allow proper operation the slave shall be in one of the following states:

- Initial state
- Configured state
- Pro-active state
- Busy state
- Power saving mode

6.5.1 Initial state

6.5.1.1 Test purpose

The slave enters the initial state as soon as it is powered on and VDD is valid or after a reset. In this state, the slave is not initialized and the SPI module is not ready to send or receive any data. The master shall not perform any SPI access while the slave is in this state.

The initial state is tested implicitly as no slave activity is expected during this phase.

6.5.1.2 Initial condition

1) The test environment shown in Figure 4.2 is used.
2) Depending on the architecture of the SPI the TT Connector either supports the testing architecture shown in Figure 4.4 for a 4 signals SPI or the testing architecture shown in Figure 4.3 for a 5 signals SPI.

6.5.1.3 Test procedure

Sequence 1 - Initial state after VDD valid

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available) for at least 1 s	The SPI contacts are traced for a time > POT (1 s) The SPI_S shall not initiate any action on the SPI within POT	RQ0605_001
2	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

Sequence 2 - Initial state after reset

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run the preparation procedure - SPI slave AC testing	The preparation procedure - SPI slave AC testing, defined in clause 6.0.5 is executed	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Run MAC activation	SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available) for at least 1 s	The SPI contacts are traced for a time > POT (defined in MCT_READY) The SPI_S shall not initiate any action on the SPI within POT	RQ0605_002
4	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

6.5.2 Configured state

6.5.2.1 Test purpose

The slave in configured state has two sub states.

In de-selected sub-state, the slave SPI module is enabled, i.e. it shall not ignore SPI_NSS assertion by master and shall be ready for an SPI access. The slave is waiting for a master access. As a consequence, the slave shall have SPI_MISO in high impedance and shall ignore both SPI_CLK and SPI_MOSI. In an implementation with 4 signals interface, SPI_NSS is not asserted by the slave.

The slave will also be in this state after issuing a MAC access request, until the master allows access for the data transfer.

In selected sub-state, the master has asserted the SPI_NSS during the MAC phase and data transfer phase. The slave SPI module is enabled, i.e. it shall not ignore SPI_NSS assertion by the master and shall not ignore SPI_CLK and SPI_MOSI. SPI_MISO is not in high impedance.

6.5.2.2 Initial condition

1) The test environment shown in Figure 4.2 is used.

2) Depending on the architecture of the SPI the TT Connector either supports the testing architecture shown in Figure 4.4 for a 4 signals SPI or the testing architecture shown in Figure 4.3 for a 5 signals SPI.

6.5.2.3 Test procedure

Sequence 1 - Configured state, de-selected sub-state during activation

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
2	SPI_S	Switch to state configured/de-selected	After POT time the SPI_S switches to configured/de-selected state	RQ0605_003
	TT	Trace activities on SPI_MISO	SPI_MISO shall be at high impedance	RQ0605_005
3	TT(SPI_M) → SPI_S	Start SPI_CLK		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The SPI_S does not react to the clock signal	RQ0605_006

Step	Direction	Action/Task	Description/Expectation	REQ
4	TT(SPI_M) → SPI_S	Provide valid data on SPI_MOSI		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The SPI_S does not react to the provisioning of data on SPI_MOSI	RQ0605_006 RQ0605_008
5	TT(SPI_M) → SPI_S	Assert SPI_NSS		
6	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available	The reaction to the assertion can be approved if the SPI_S responds to data on SPI_MISO	RQ0605_004
7	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

Sequence 2 - Configured state, de-selected sub-state after issuing a MAC access

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
2	TT(SPI_M) → SPI_S	Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after initial POT (1 s)	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
4	SPI_S → TT(SPI_M)	Assert SPI_NSS (4 signal SPI) or Assert SPI_INT (5 signal SPI)	The SPI_S is issuing a MAC access request	
5	SPI_S	Switch to state configured/de-selected	While waiting for the SPI_M to generate an access for the data transfer the SPI_M switches to configured, de-selected state	RQ0605_007
	TT	Trace activities on SPI_MISO	SPI_MISO shall be at high impedance	RQ0605_005
6	TT(SPI_M) → SPI_S	Start SPI_CLK		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The SPI_S does not react to the clock signal	RQ0605_006
7	TT(SPI_M) → SPI_S	Provide valid data on SPI_MOSI		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The SPI_S does not react to the provisioning of data on SPI_MOSI	RQ0605_006 RQ0605_008
8	TT(SPI_M) → SPI_S	Assert SPI_NSS		
9	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available	The reaction to the assertion can be approved if the SPI_S responds to data on SPI_MISO	RQ0605_004
10	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

Sequence 3 - Configured state, selected sub-state, return to de-selected sub-state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
2	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after initial POT (1 s)	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
4	SPI_S → TT(SPI_M)	Assert SPI_NSS (4 signal SPI) or Assert SPI_INT (5 signal SPI)	The SPI_S is issuing a MAC access request	
5	SPI_S	Switch to state configured/de-selected	While waiting for the SPI_M to generate an access for the data transfer the SPI_M switches to configured, de-selected state	RQ0605_007
6	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_S is selected	
	SPI_S	Switch to state configured/selected		RQ0605_010
7	TT	Trace activities on SPI_MISO	SPI_MISO shall not be at high impedance	RQ0605_011
8	TT(SPI_M) → SPI_S	Activate SPI_CLK		
9	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available	The reaction to the activities on SPI_CLK and SPI_MOSI can be approved if the SPI_S responds to data on SPI_MISO	RQ0605_012
10	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S is de-selected	
	SPI_S	Switch to state configured/de-selected		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The switch back to configured/de-selected state can be approved if the SPI_S is not initiating any action on the SPI	RQ0605_013
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

Sequence 4 - Configured state, selected sub-state, return to initial state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)	
2	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after initial POT (1 s)	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
4	SPI_S → TT(SPI_M)	Assert SPI_NSS (4 signal SPI) or Assert SPI_INT (5 signal SPI)	The SPI_S is issuing a MAC access request	
5	SPI_S	Switch to state configured/de-selected	While waiting for the SPI_M to generate an access for the data transfer the SPI_M switches to configured, de-selected state	RQ0605_007

Step	Direction	Action/Task	Description/Expectation	REQ
6	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_S is selected	
	SPI_S	Switch to state configured/selected		RQ0605_010
7	TT	Trace activities on SPI_MISO	SPI_MISO shall not be at high impedance	RQ0605_011
8	TT(SPI_M) → SPI_S	Activate SPI_CLK		
9	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available	The reaction to the activities on SPI_CLK and SPI_MOSI can be approved if the SPI_S responds to data on SPI_MISO	RQ0605_012
10	TT/Tester	Reset the SPI		
	SPI_S	Switch to initial state		
11	TT(SPI_M) → SPI_S	Assert SPI_NSS, activate SPI_CLK and provide valid data on SPI_MOSI in a time << POT		
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (if available)	The switch back to initial state can be approved if the SPI_S is not reacting to action on the SPI	RQ0605_013
12	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

NOTE: The option where the slave switches from configured/de-selected mode to power saving mode is implicitly tested in clause 13.2.1.

6.5.3 5 signals SPI - Pro-active state

6.5.3.1 Test purpose

The Pro-active state is entered by the slave when data need to be sent. The slave in a 5 signal SPI issues a MAC access request by asserting SPI_INT. The slave enters this state for the duration of T2, as described in ETSI TS 103 713 [1] clause 7.2.3.2. The slave exits this state on its own after T2, regardless of the input signals states driven by the master.

6.5.3.2 Initial condition

1) The test environment shown in Figure 4.2 is used.

2) The TT Connector supports the testing architectures shown in Figure 4.3 for a 5 signals SPI.

6.5.3.3 Test procedure

Sequence 1 - Pro-active state, MAC access from slave, return to configured/de-selected state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	

Step	Direction	Action/Task	Description/Expectation	REQ
6	SPI_S → TT(SPI_M)	Assert SPI_INT for the T2 duration	The SPI_S is issuing a MAC access request by asserting SPI_INT while switching to the pro-active state	RQ0605_014
	TT	Trace activities on SPI_INT	After T2 SPI_INT is de-asserted	RQ0605_015
7	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_M asserts SPI_NSS when detecting the assertion of SPI_INT	
	TT	Trace activities on SPI_MISO	No activities shall be seen on SPI_MISO	
8	TT(SPI_M) → SPI_S	After T1 activate SPI_CLK and request data from SPI_S	The SPI_S receives the data request from SPI_M	
9	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives the requested data from SPI_M	
10	TT(SPI_M) → SPI_S	Stop SPI_CLK and de-assert SPI_NSS	The SPI_M switches to the state configured/de-selected	RQ0605_019
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

Sequence 2 - Pro-active state, simultaneous initiation, return to configured/selected state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
6	SPI_S → TT(SPI_M)	Assert SPI_INT for the T2 duration	The SPI_S is issuing a MAC access request by asserting SPI_INT while switching to the pro-active state	RQ0605_014
	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK	The SPI_M asserts the SPI_NSS similar to a regular activation	
	TT	Trace activities on SPI_INT	After T2 SPI_INT is de-asserted while the SPI_S is switching to configured/selected state	RQ0605_015 RQ0605_020
7	TT(SPI_M) → SPI_S	Send data on SPI_MOSI after T1	The SPI_S receives the data from SPI_S	
8	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives data from SPI_M	
10	TT(SPI_M) → SPI_S	Stop SPI_CLK and de-assert SPI_NSS	The SPI_M switches to the state configured/de-selected	RQ0605_019
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

Sequence 3 - Pro-active state, MAC access from slave, return to initial state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF prepared by the TT	

Step	Direction	Action/Task	Description/Expectation	REQ
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
6	SPI_S → TT(SPI_M)	Assert SPI_INT for the T2 duration	The SPI_S is issuing a MAC access request by asserting SPI_INT while switching to the pro-active state	RQ0605_014
	TT	Trace activities on SPI_INT	After T2 SPI_INT is de-asserted	RQ0605_015
7	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_M asserts SPI_NSS when detecting the assertion of SPI_INT	
	TT	Trace activities on SPI_MISO	No activities shall be seen on SPI_MISO	
8	TT(SPI_M) → SPI_S	After T1 activate SPI_CLK and request data from SPI_S	The SPI_S receives the data request from SPI_M	
9	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives the requested data from SPI_M	
10	TT/Tester	Reset the SPI	The SPI_M switches to initial state	RQ0605_021
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

6.5.4 4 signals SPI - Pro-active state

6.5.4.1 Test purpose

The Pro-active state is entered by the slave when data need to be sent. The slave in a 4 signals SPI issues a MAC access request by asserting the SPI_NSS. The slave enters this state for the duration of T2, as described in ETSI TS 103 713 [1] clause 7.2.4.3. The slave exits this state on its own after T2, regardless of the input signals states driven by the master.

6.5.4.2 Initial condition

1) The test environment shown in Figure 4.2 is used.

2) The TT Connector supports the testing architectures shown in Figure 4.4 for a 4 signals SPI.

6.5.4.3 Test procedure

Sequence 1 - Pro-active state, MAC access from slave, return to configured/de-selected state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
6	SPI_S → TT(SPI_M)	Assert SS_SO to drive SPI_NSS to low level for T2	The SPI_S is issuing a MAC access request by asserting SPI_NSS while switching to the pro-active state	RQ0605_014 RQ0605_018
	TT(SPI_M) → SPI_S	Activate SPI_CLK and provide valid data on SPI_MOSI	No activities shall be seen on SPI_MISO while in pro-active state	RQ0605_016
	TT	Trace activities on SPI_MISO	SPI_MISO shall be in high impedance	RQ0605_017
	TT	Trace activities on SPI_NSS	After T2 SPI_NSS is de-asserted	
7	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_M asserts SPI_NSS at T2 (when SPI_S is no longer asserting SPI_NSS)	RQ0605_015

Step	Direction	Action/Task	Description/Expectation	REQ
8	TT(SPI_M) → SPI_S	After T1 request data from SPI_S	The SPI_S receives the data request from SPI_M	
9	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives the requested data from SPI_M	
10	TT(SPI_M) → SPI_S	Stop SPI_CLK and de-assert SPI_NSS	The SPI_M switches to the state configured/de-selected	RQ0605_019
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	

Sequence 2 - Pro-active state, simultaneous initiation, return to configured/selected state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF frame prepared by the TT	
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	
	SPI_S → TT(SPI_M)	Assert SS_SO to drive SPI_NSS to low level for T2	The SPI_S is issuing a MAC access request by asserting SPI_NSS while switching to the pro-active state	RQ0605_014 RQ0605_018
	TT(SPI_M) → SPI_S	Assert SPI_NSS (in parallel to the assertion by SPI_S) Activate SPI_CLK and provide valid data on SPI_MOSI	No activities shall be seen on SPI_MISO while in pro-active state	RQ0605_016
	TT	Trace activities on SPI_MISO	SPI_MISO shall be in high impedance	RQ0605_017
7	SPI_S → TT(SPI_M)	After T1 receives data from SPI_M	The SPI_S switches to configured/selected state before it receives the data provided by the SPI_M	RQ0605_020
8	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives data from SPI_M	
9	TT/Tester	Reset the SPI		
10	TT(SPI_M) → SPI_S	Deactivate VDD	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	

Sequence 3 - Pro-active state, MAC access from slave, return to initial state

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT/Tester	Reset the SPI		
3	TT(SPI_M) → SPI_S	Activate VDD	The emulated SPI_M activates VDD but is not providing any signal on SPI_CLK or asserting SPI_NSS	
4	TT(SPI_M) → SPI_S	Assert SPI_NSS Activate SPI_CLK (1 MHz) and send MCT_MASTER_REQ_DEF frame after POT	The SPI slave receives the MCT_MASTER_REQ_DEF prepared by the TT	
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S intends to send the MCT_READY but got de-asserted	

Step	Direction	Action/Task	Description/Expectation	REQ
6	SPI_S → TT(SPI_M)	Assert SS_SO to drive SPI_NSS to low level for T2	The SPI_S is issuing a MAC access request by asserting SPI_NSS while switching to the pro-active state	RQ0605_014 RQ0605_018
	TT(SPI_M) → SPI_S	Activate SPI_CLK and provide valid data on SPI_MOSI	No activities shall be seen on SPI_MISO while in pro-active state	RQ0605_016
	TT	Trace activities on SPI_MISO	SPI_MISO shall be in high impedance	RQ0605_017
	TT	Trace activities on SPI_NSS	After T2 SPI_NSS is de-asserted	
7	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_M asserts SPI_NSS at T2 (when SPI_S is no longer asserting SPI_NSS)	RQ0605_015
8	TT(SPI_M) → SPI_S	After T1 request data from SPI_S	The SPI_S receives the data request from SPI_M	
9	SPI_S → TT(SPI_M)	Send data on SPI_MISO	The SPI_M receives the requested data from SPI_M	
10	TT/Tester	Reset the SPI	The SPI_M switches to initial state	RQ0605_021
11	TT(SPI_M) → SPI_S	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	

6.5.5 4 signals SPI - Busy state

6.5.5.1 Test purpose

The busy state is an optional state applicable for the 4 signal SPI only when the slave performs the optional slave driven flow control by keeping SPI_NSS asserted following the configured/selected state.

6.5.5.2 Initial condition

1) The test environment shown in Figure 4.2 is used.

2) The TT Connector supports the testing architectures shown in Figure 4.4 for a 4 signals SPI.

6.5.5.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT(SPI_M) → SPI_S	Assert SPI_NSS provide a clock signal on SPI_CLK and valid data on SPI_MOSI	At assertion of SPI_NSS the SPI_S is switched to configured/selected state. Then the SPI_M starts a data transfer	
3	SPI_S	Assert SPI_NSS	The SPI_S asserts SPI_NSS (via SS_SO) whilst the data transfer from SPI_M is in progress The SPI_S shall not start to send any frame in the middle of the current access The SPI_S is switched to busy state Set timer for the assertion of SPI_NSS	RQ0605_022 RQ0605_023 (see note)
	TT	Trace activities on SPI_NSS and SPI_MISO	The SPI_M shall not handle the assertion of SPI_NSS by the SPI_S as a MAC access request	

Step	Direction	Action/Task	Description/Expectation	REQ
4	SPI_M → TT(SPI_M)	De-assert SPI_NSS	Set timer for the de-assertion of SPI_NSS The slave should not keep SPI_NSS asserted longer than 500 µs The SPI_S enters the configured/de-selected state	 RQ0605_024
	TT	Determine the time for the SPI_NSS assertion by SPI_S	The slave should not keep SPI_NSS asserted longer than 500 µs	RQ0702_037
5	TT(SPI_M)	Assert SPI_NSS	Any time after the SPI_S de-asserts SPI_NSS the SPI_M may assert SPI_NSS	
6	TT(SPI_M) → SPI_S	Continue sending data		
7	TT(SPI_M)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	
NOTE: RQ0605_023 (In the optional busy state the slave may keep its SPI module enabled) cannot be verified.				

6.5.6 Power saving mode state

6.5.6.1 Test purpose

The Power saving mode state is entered by the slave to reduce the power consumption. Entry and exit conditions are described in ETSI TS 103 713 [1] clause 7.8.

Tests appropriate for the power saving mode and power saving mode state are defined in clause 13 - Power management.

7 Test cases for data link layer- MAC Layer

7.0 Common conditions for data link layer test cases

7.0.1 Pre-condition for data link layer test cases

A master is connected to a single slave in accordance to the definitions given in ETSI TS 103 713 [1] for 4 signals or 5 signals SPI. The SPI test tool (TT) is connected as defined in the testing architectures in clause 4.1.1 or 4.1.2.

7.0.2 MAC parameter determination procedure - SPI Master testing

Step	Direction	Action/Task	Description/Expectation
0.1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on
0.2	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)
0.3	SPI_M → TT(SPI_S)	Activate SPI_CLK and send MCT_MASTER_REQ frame after initial POT	
	TT	Analyse and store the MCT_MASTER_REQ	Evaluate the supported power mode, the suggested MTU length, the T4 time for the master and the support of SHDLC based flow control
0.4	TT(SPI_S) → SPI_M	Return MCT_READY_CONF	The TT(SPI_S) sends MCT_READY_CONF confirming the SPI_M values for the supported power mode, the MTU length, and the SHDLC based flow control. SPI_S specific values are set as defined in MCT_READY_CONF

Step	Direction	Action/Task	Description/Expectation
0.5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)

Test cases for SPI master testing without an explicitly defined initial MAC activation or a reset of the SPI shall operate with the power mode, flow control, power saving mode, MTU length and timing parameters derived from this initial handshake and data transfer phase as long as the same SUT is used.

7.0.3 MAC parameter determination procedure - SPI Slave testing

Step	Direction	Action/Task	Description/Expectation
0.1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on
0.2	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 (5 signals SPI) or C.2.1 (4 signals SPI)
0.3	TT(SPI_M) → SPI_S	Activate SPI_CLK and send MCT_MASTER_REQ_CONF frame after initial POT	
0.4	SPI_S → TT(SPI_M)	Return MCT_READY	
	TT	Analyse and store the MCT_READY	The SPI_S sends an MCT_READY with the parameters supported
0.5	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)

Test cases for SPI slave testing without an explicitly defined initial MAC activation or a reset of the SPI shall operate with the power mode, flow control, power saving mode, MTU length and timing parameters derived from this initial handshake and data transfer phase as long as the same SUT is used.

7.0.4 Post-condition for data link layer test cases

Step	Direction	Action/Task	Description/Expectation	REQ
n.1	SPI_M → SPI_S	Run MAC deactivation SPI	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 (5 signals SPI) or C.2.2 (4 signals SPI)	

7.1 MAC Layer - 5 signals SPI - SPI Master testing

7.1.1 5 signals SPI - Master behaviour during initial data transfer initiation

7.1.1.1 Test purpose

After powering on the SPI the MCT is executed. Timing values and the behaviour of the master during initial MCT and the following data transfer phase is checked.

7.1.1.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY_CONF frame are send during MCT LLC setup.

2) The testing architecture described in Figure 4.3 is used.

3) Initial parameters shall be considered:

- POT: 1 s
- SPI_CLK: 1 MHz
- T1: 255 µs

7.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on	
2	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_001 RQ0702_018
	TT	Generate time stamp for switching on VDD		
3	SPI_M → TT(SPI_S)	Assert SPI_NSS to initiate MAC phase	SPI_NSS is asserted	RQ0702_006 RQ0706_016 RQ0706_022
	TT	Generate a time stamp for the assertion of SPI_NSS	The TT calculates the elapsed time between VDD switched on and SPI_NSS assertion The TT verifies if the initial POT is in accordance to its definition (1 s ± 10 %)	
4	SPI_M → (TT)SPI_S	Start toggling SPI_CLK		
	TT	Generate time stamp for the start of SPI_CLK toggling		
		Start measuring the clock frequency		
5	SPI_M → (TT)SPI_S	Send an MCT_MASTER_REQ		RQ0706_004 RQ0706_005
	TT	Generate time stamp for the start of the transfer of the MCT_MASTER_REQ	The TT calculates the elapsed time between activation of SPI_CLK and sending the MCT_MASTER_REQ (T1 ≥ 255 µs ± 10 %)	RQ0706_003
6	TT(SPI_S) → SPI_M	Return MCT_READY_CONF	The TT(SPI_S) sends MCT_READY_CONF confirming the SPI_M values for the supported power mode, the MTU length, and the SHDLC based flow control in low power mode	RQ0706_004 RQ0706_007
	TT	Stop measuring the clock frequency	The TT calculates the frequency of SPI_CLK (1 MHz ± 10 %)	RQ0706_005
7	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

7.1.1.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 shall be executed at the end the sequence.

7.1.2 5 signals SPI - Master behaviour during data transfer initiation

7.1.2.1 Test purpose

After powering on the SPI the MCT is executed. Timing values and the behaviour of the master during MCT and the following data transfer phase is checked.

7.1.2.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

    - The slave is emulated to ensure that the specific parameters given in the MCT_READY_CONF frame are sent during MCT LLC setup.

2) The testing architecture described in Figure 4.3 is used.

3) Parameters negotiated in the initial MCT phase shall be considered, where the SPI slave parameters are set to:

    - POT: ≥ 10 ms
    - SPI_CLK: 10 MHz
    - T1: ≥ 100 µs

4) To properly handle the SPI master parameters either test case 7.1.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.1.2.3 Test procedure

Sequence 1 - Data transfer initiation with formerly negotiated MCT_DATA

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_001 RQ0702_018
	TT	Generate time stamp for switching on VDD		
2	SPI_M → TT(SPI_S)	Assert SPI_NSS to initiate MAC phase	SPI_NSS is asserted	RQ0702_006 RQ0706_016 RQ0706_022
	TT	Generate a time stamp for the assertion of SPI_NSS	The TT calculates the elapsed time between VDD switch on and SPI_NSS assertion The TT verifies if the POT is not shorter than defined in MCT_READY_CONF POT ≥10 ms, - 10 %	RQ0706_017
3	SPI_M → (TT)SPI_S	Start toggling SPI_CLK		
	TT	Generate time stamp for the start of SPI_CLK toggling		
		Start measuring the clock frequency		
4	SPI_M → (TT)SPI_S	Send an MCT_MASTER_REQ		RQ0706_004 RQ0706_005
	TT	Generate time stamp for the start of the transfer of the MCT_MASTER_REQ	The TT calculates the elapsed time between activation of SPI_CLK and sending the MCT_MASTER_REQ T1 is not shorter than defined in MCT_READY_CONF T1 ≥ 100 µs, - 10 %	RQ0702_001 RQ0706_003
5	TT(SPI_S) → SPI_M	Return MCT_READY_CONF	The TT(SPI_S) sends MCT_READY_CONF confirming the SPI_M values for the supported power mode, the MTU length, and the SHDLC based flow control in the power mode supported by the SPI_M	RQ0706_004 RQ0706_007 RQ0706_018
	TT	Stop measuring the clock frequency	The TT calculates the frequency of SPI_CLK In accordance to MCT_READY_CONF SPI_CLK ≥ 10 MHz ± 10 %	RQ0706_005
6	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

7.1.2.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 shall be executed at the end the sequence.

7.1.3 5 signals SPI - Master behaviour during simultaneous data transfer initiation

7.1.3.1 Test purpose

After SPI activation the MCT is executed. SPI master behaviour is tested during the simultaneous initiations from the master and the slave.

7.1.3.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.
   - The slave is emulated to ensure that the specific parameters given in the MCT_READY frame can be fulfilled.
2) To properly handle the SPI master parameters either test case 7.1.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.1.3.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_018 RQ0702_001
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	RQ0706_004
3	TT(SPI_S)	Wait for MCT_SLAVE_TIMEOUT + 10 ms		
4	SPI_M → TT(SPI_S)	Assert SPI_NSS	Simultaneous initiation from SPI_M and SPI_S	
	TT(SPI_S)	Assert SPI_INT at high state		
5	SPI_M → TT(SPI_S)	Resend MCT_MASTER_REQ at a time greater than T1	The SPI_S receives the MCT_MASTER_REQ frame	RQ0706_018 RQ0706_019 RQ0702_015
6	TT(SPI_S) → SPI_M	Send MCT_READY_DEF frame		
7	SPI_M → TT(SPI_S)	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	RQ0702_008 RQ0702_014

7.1.3.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.2 is executed at the end of the test case.

7.1.4 5 signals SPI - MAC deactivation

7.1.4.1 Test purpose

The SPI master behaviour during power off is tested.

7.1.4.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that no signal line is asserted by the slave during testing.

   - The SPI master is the SUT.

2) To properly handle the SPI master parameters either test case 7.1.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.1.4.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	RQ0702_018 RQ0702_001
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	RQ0706_004
3	TT(SPI_S) → SPI_M	Send MCT_READY_DEF frame		
4	SPI_M → TT(SPI_S)	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	RQ0702_008 RQ0702_014
5	SPI_M → TT(SPI_S)	Run MAC deactivation	SPI_NSS is set to high impedance VDD power line is set to OFF	RQ0702_020

7.1.4.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.2 MAC Layer - 5 signals SPI - SPI Slave testing

7.2.1 5 signals SPI - Slave behaviour at initial MAC activation

7.2.1.1 Test purpose

The initial MAC activation is executed. The slave behaviour is tested for an initial MCT phase performed with SPI_CLK = 1 MHz and POT = 1 s.

7.2.1.2 Initial conditions

1) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that default parameters are used during initial MAC activation and data transfer.

   - The SPI slave is the SUT.

2) To properly handle the SPI slave parameters the MAC parameter determination procedure from clause 7.0.3 shall be executed.

7.2.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT(SPI_M) → SPI_S	Initiate a data transfer for the MAC phase Toggle SPI_CLK with 1 MHz after a delay of 1 s (POT) after switching on VDD		
3	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_CONF frame	The SPI_S receives the pre-defined MCT_MASTER_REQ	RQ0706_016
4	SPI_S → TT(SPI_M)	Assert SPI_INT on the rising edge of the SPI_INT pulse	SPI_INT is asserted for a minimum duration of T2	
	TT	Measurement of SPI_INT	Verify that SPI_INT is asserted for a time ≥T2 (1 µs)	RQ0702_003 RQ0702_010
5	TT(SPI_M) → SPI_S	Assert SPI_NSS and start data transfer	SPI_M starts data transfer at a time greater than T1 measured from the leading edge of SPI_INT	RQ0702_002 RQ0702_011
6	SPI_S → TT(SPI_M)	Send MCT_READY frame	The MCT_READY frame with an MCT LPDU length lower than or equal to 29 bytes containing MCT_DATA is received by the TT(SPI_M)	RQ0706_001 RQ0706_002 RQ0706_004
7	SPI_S → TT(SPI_M)	De-assert SPI_NSS	After data transfer completion and SPI_CLK stop, the master de-asserts SPI_NSS	RQ0702_015

7.2.1.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.2.2 5 signals SPI - Slave behaviour during data transfer initiation - nominal test

7.2.2.1 Test purpose

The MAC activation is executed. The slave behaviour during data transfer initiation with negotiated MCT timing is tested.

7.2.2.2 Initial conditions

1) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that the parameters provided in the initial MCT_READY are used during MAC activation and data transfer.

   - The SPI slave is the SUT.

2) To properly handle the SPI slave parameters the MAC parameter determination procedure from clause 7.0.3 shall be executed.

7.2.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_DEF	The SPI_S receives the predefined MCT_MASTER_REQ	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	
4	SPI_S → TT(SPI_M)	Assert SPI_INT on the rising edge of the SPI_INT pulse	SPI_INT is asserted for a minimum duration of T2	
	TT	Measurement of SPI_INT	Verify that SPI_INT is asserted for a time ≥T2 (1 µs)	RQ0702_003 RQ0702_010
5	TT(SPI_M) → SPI_S	Assert SPI_NSS and starts data transfer	SPI_M starts data transfer at a time greater than T1	RQ0702_002 RQ0702_011
6	SPI_S → TT(SPI_M)	Send MCT_READY frame	The MCT_READY frame with an MCT LPDU length lower than or equal to 29 bytes containing MCT_DATA as defined in table 7.5 of ETSI TS 103 713 [1] is sent to the SPI_M within MCT_SLAVE_TIMEOUT	RQ0706_001 RQ0706_002 RQ0706_004 RQ0706_023
7	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	

7.2.2.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.2.3 5 signals SPI - Slave behaviour during data transfer initiation by the slave

7.2.3.1 Test purpose

After MCT_MASTER_REQ is sent the master keeps the NSS line asserted. The slave shall not assert the SPI_INT while the NSS is asserted.

7.2.3.2 Initial conditions

1) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that default parameters are used during initial MAC activation and data transfer.

   - The SPI slave is the SUT.

2) To properly handle the SPI slave parameters the MAC parameter determination procedure from clause 7.0.3 shall be executed.

7.2.3.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_DEF frame	The SPI_S receives the pre-defined MCT_MASTER_REQ frame	
3	TT(SPI_M) → SPI_S	Keep SPI_NSS asserted for 250 µs after the MCT_MASTER_REQ is sent	While the NSS is asserted the slave shall not assert the SPI_INT	RQ0702_009
4	TT(SPI_M) → SPI_S	De-assert SPI_NSS		
5	SPI_S → TT(SPI_M)	Assert SPI_INT on the rising edge of the SPI_INT pulse	SPI_INT is asserted for a minimum duration of T2	
	TT	Measurement of SPI_INT	Verify that SPI_INT is asserted for a time ≥T2 (1 µs)	RQ0702_003 RQ0702_010
6	TT(SPI_M) → SPI_S	Assert SPI_NSS and start data transfer	SPI_M starts data transfer at a time greater than T1	RQ0702_002 RQ0702_011
7	SPI_S → TT(SPI_M)	Send MCT_ READY frame	The MCT_ READY frame with an MCT LPDU length lower than or equal to 29 bytes containing MCT_DATA as defined in table 7.5 of ETSI TS 103 713 [1] is received by the SPI_M	RQ0706_001 RQ0706_002 RQ0706_004
8	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	

7.2.3.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.2 is executed at the end of the test case.

7.3 MAC Layer - 4 signals SPI - SPI Master testing

7.3.1 4 signals SPI - Master behaviour during initial data transfer initiation

7.3.1.1 Test purpose

After powering on the SPI the MCT is executed. Timing values and the behaviour of the master during initial MCT and the following data transfer phase is checked.

7.3.1.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY_CONF frame are sent during MCT LLC setup.

2) The testing architecture described in Figure 4.4 is used.

3) Initial parameters shall be considered:

   - POT: 1 s
   - SPI_CLK: 1 MHz
   - T1: 255 µs

7.3.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT/Tester	Power on	The power supply of the PCB hosting the SPI is switched on	
2	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_019 RQ0702_038
	TT	Generate time stamp for switching on VDD		
3	SPI_M → TT(SPI_S)	Assert SPI_NSS to initiate MAC phase	SPI_NSS is asserted	RQ0702_020 RQ0706_016
	TT	Generate a time stamp for the assertion of SPI_NSS	The TT calculates the elapsed time between VDD switched on and SPI_NSS assertion The TT verifies if the initial POT is in accordance to its definition (1 s ± 10 %)	
4	SPI_M → (TT)SPI_S	Start toggling SPI_CLK		
	TT	Generate time stamp for the start of SPI_CLK toggling		
		Start measuring the clock frequency		
5	SPI_M → (TT)SPI_S	Send an MCT_MASTER_REQ		RQ0706_004 RQ0706_005
	TT	Generate time stamp for the start of the transfer of the MCT_MASTER_REQ	The TT calculates the elapsed time between activation of SPI_CLK and sending the MCT_MASTER_REQ (T1 ≥ 255 µs ± 10 %)	RQ0706_003
6	TT(SPI_S) → SPI_M	Return MCT_READY_CONF	The TT(SPI_S) sends MCT_READY_CONF confirming the SPI_M values for the supported power mode, the MTU length, and the SHDLC based flow control in low power mode	RQ0706_004 RQ0706_007
	TT	Stop measuring the clock frequency	The TT calculates the frequency of SPI_CLK (1 MHz ± 10 %)	RQ0706_005
7	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2	

7.3.1.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 shall be executed at the end of the sequence.

7.3.2 4 signal SPI - Master behaviour during data transfer initiation

7.3.2.1 Test purpose

After powering on the SPI the MCT is executed. Timing values and the behaviour of the master during MCT and the following data transfer phase is checked.

7.3.2.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY_CONF frame are send during MCT LLC setup.

2) The testing architectures described in Figure 4.4 are used.

3) Parameters negotiated in the initial MCT phase shall be considered, where the SPI slave parameters are set to:

- POT: ≥ 10 ms
- SPI_CLK: 10 MHz
- T1: ≥ 100 µs

4) To properly handle the SPI master parameters either test case 7.3.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.3.2.3 Test procedure

Sequence 1 - Data transfer initiation with formerly negotiated MCT_DATA

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_038
	TT	Generate time stamp for switching on VDD		
2	SPI_M → TT(SPI_S)	Assert SPI_NSS to initiate MAC phase	SPI_NSS is asserted	RQ0706_020
	TT	Generate a time stamp for the assertion of SPI_NSS	The TT calculates the elapsed time between VDD switched on and SPI_NSS assertion The TT verifies if the POT is not shorter than defined in MCT_READY_CONF POT ≥10 ms, - 10 %	RQ0702_001 RQ0702_002 RQ0706_017
3	SPI_M → (TT)SPI_S	Start toggling SPI_CLK		
	TT	Generate time stamp for the start of SPI_CLK toggling		
		Start measuring the clock frequency		
4	SPI_M → (TT)SPI_S	Send an MCT_MASTER_REQ		RQ0706_004
	TT	Generate time stamp for the start of the transfer of the MCT_MASTER_REQ	The TT calculates the elapsed time between activation of SPI_CLK and sending the MCT_MASTER_REQ T1 is not shorter than defined in MCT_READY_CONF T1 ≥ 100 µs, - 10 %	RQ0702_001 RQ0706_003
5	TT(SPI_S) → SPI_M	Return MCT_READY_CONF	The TT(SPI_S) sends MCT_READY_CONF confirming the SPI_M values for the supported power mode, the MTU length, and the SHDLC based flow control in the power mode supported by the SPI_M	RQ0706_004 RQ0706_007 RQ0706_018
	TT	Stop measuring the clock frequency	The TT calculates the frequency of SPI_CLK In accordance to MCT_READY_CONF SPI_CLK ≥ 10 MHz ± 10 %	RQ0706_005
6	SPI_M → TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	RQ0702_039

7.3.2.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 shall be executed at the end of the sequence.

7.3.3 4 signal SPI - Master behaviour during simultaneous data transfer initiation

7.3.3.1 Test purpose

After SPI activation the MCT is executed. SPI master behaviour is tested during the simultaneous initiations from the master and the slave.

7.3.3.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY frame can be fulfilled.

2) To properly handle the SPI master parameters either test case 7.3.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.3.3.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M→ TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_018 RQ0702_038
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	RQ0706_004
3	TT(SPI_S)	Wait for MCT_SLAVE_TIMEOUT + 10 ms		
4	SPI_M	Assert SPI_NSS	Simultaneous initiation from SPI_M and SPI_S	
	TT(SPI_S)	Assert SS_SO		
5	SPI_M → TT(SPI_S)	Resend MCT_MASTER_REQ at a time greater than T1	The SPI_S receives the MCT_MASTER_REQ frame	RQ0706_018 RQ0706_019 RQ0702_030
6	TT(SPI_S) → SPI_M	Send MCT_READY_DEF frame		
7	SPI_M→ TT(SPI_S)	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	RQ0702_022 RQ0702_029

7.3.3.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.3.4 4 signal SPI - Master behaviour during data flow control

7.3.4.1 Test purpose

After SPI activation the MCT is executed. SPI master behaviour is tested during the data flow control from the slave.

7.3.4.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY frame can be fulfilled.

2) To properly handle the SPI master parameters either test case 7.3.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.3.4.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M→ TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_018 RQ0702_038
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	SPI_CLK is toggled	RQ0706_004
3	TT(SPI_S)	While the SPI_CLK is toggled the slave asserts SS_SO for 450 µs to drive SPI_NSS to the low state after the detection of the first transferred data.	After data transfer is completed SPI_M shall not assert SS_MO and shall not toggle SPI_CLK until SS_SO is asserted	RQ0702_035
4	TT(SPI_S) → SPI_M	Send MCT_READY_DEF frame		
5	SPI_M→ TT(SPI_S)	De-assert SPI_NSS		

7.3.4.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.3.5 4 signal SPI - MAC deactivation

7.3.5.1 Test purpose

SPI master behaviour is tested during power off.

7.3.5.2 Initial conditions

1) The test environment shown in Figure 4.1 is used.
   - The slave is emulated to ensure that no signal line is asserted by the slave during testing.
   - The SPI master is the SUT.
2) To properly handle the SPI master parameters either test case 7.3.1 or the MAC parameter determination procedure from clause 7.0.2 shall be executed.

7.3.5.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M→ TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	RQ0702_018 RQ0702_001
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	RQ0706_004
3	TT(SPI_S) → SPI_M	Send MCT_READY_DEF frame		
4	SPI_M → TT(SPI_S)	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	RQ0702_023
5	SPI_M → TT(SPI_S)	De-activate SPI	SS_MO is set to low level VDD power line is set to OFF	RQ0702_040

7.3.5.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

7.4 MAC Layer - 4 signals SPI - SPI Slave testing

7.4.1 4 signal SPI - Slave behaviour during data transfer initiation

7.4.1.1 Test purpose

After SPI activation the MCT is executed. MCT timing values and the SPI Slave behaviour is tested.

7.4.1.2 Initial conditions

1) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that default parameters are used during initial MAC activation and data transfer.

   - The SPI slave is the SUT.

2) To properly handle the SPI slave parameters the MAC parameter determination procedure from clause 7.0.3 shall be executed.

7.4.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_DEF frame	The SPI_S receives the pre-configured MCT_MASTER_REQ frame	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	
4	SPI_S	Assert SS_SO to drive SPI_NSS to the low state to initiate a MAC access request to transfer MCT_READY	SS_SO is asserted for a minimum of T2	RQ0702_025
5	TT(SPI_M) → SPI_S	Assert the SPI_NSS and starts data transfer at a time greater than T1	SPI_M starts data transfer at a time greater than T1	RQ0702_002 RQ0702_026
6	SPI_S → TT(SPI_M)	Send MCT_READY frame	The MCT_READY frame with an MCT LPDU as defined in ETSI TS 103 713 [1] is received by the SPI_M	RQ0702_031 RQ0702_032 RQ0706_001 RQ0706_002 RQ0706_004 RQ0706_023
7	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	

7.4.1.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.2 is executed at the end of the test case.

7.4.2 4 signal SPI - Slave behaviour during simultaneous data transfer initiation

7.4.2.1 Test purpose

After SPI activation the MCT is executed. MCT timing values and the SPI Slave behaviour is tested during the simultaneous initiations from the master and the slave.

7.4.2.2 Initial conditions

1) The test environment shown in Figure 4.2 is used:

   - The master is emulated to ensure that the parameters provided in the initial MCT_READY are used during MAC activation and data transfer.

   - The SPI slave is the SUT.

2) To properly handle the SPI slave parameters the MAC parameter determination procedure from clause 7.0.3 shall be executed.

7.4.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.2.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_DEF frame	The SPI_S receives the pre-configured MCT_MASTER_REQ frame	
3	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	
4	SPI_S	Assert SS_SO to drive SPI_NSS to the low state	SS_SO is asserted for a minimum of T2	RQ0702_025
	TT(SPI_M)	Assert SS_MO	Simultaneous initiation from SPI_M and SPI_S	
5	TT(SPI_M) → SPI_S	Start data transfer at a time greater than T1		
6	SPI_S → TT(SPI_M)	Send MCT_READY frame	The MCT_READY frame with an MCT LPDU length lower than or equal to 29 bytes is received by the SPI_M	RQ0706_001 RQ0706_002 RQ0706_004 RQ0706_023
7	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	

7.4.2.4 Post-condition

The general post-condition for data link layer test cases as defined in clause 7.0.4 is executed at the end of the test case.

8 Test cases for data link layer- Link layer

8.1 Link layer - SPI Master testing

8.1.1 Link layer - Master frame generation

8.1.1.1 Test purpose

The SPI master shall send a frame in a single SPI access, even if the SPI access initiated by the master has a length higher than the length of the frame to be send, up to MTU.

8.1.1.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.1 is used.

   - The slave is emulated to ensure that the specific parameters given in the MCT_READY_CONF frame are send during MCT LLC setup.

3) To properly handle the SPI master parameters the MAC parameter determination procedure from clause 7.0.2 shall be executed.

8.1.1.3 Test procedure

Sequence 1 - Link layer - Master frame generation - MTU = 32 bytes

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ with MTU = 32 bytes (as supported by the SPI_M) after VDD is switched on	The SPI_S confirms the MTU length suggested in the MCT_MASTER_REQ	
2	SPI_M → TT(SPI_S)	Assert SPI_NSS, then wait for minimum T1 time	SPI_NSS signal is in asserted state	
3	SPI_M → TT(SPI_S)	Initiate SPI access and start data transfer	The data sent from the SPI_M shall be received by the SPI_S in a single SPI access NSD bytes have been added if the SPI access has been longer than the frame being send	RQ0703_014 RQ0703_015

Sequence 2 - Link layer - Master frame generation - MTU = 64 bytes

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ with MTU = 64 bytes (as supported by the SPI_M) after VDD is switched on	The SPI_S confirms the MTU length suggested in the MCT_MASTER_REQ	
2	SPI_M → TT(SPI_S)	Assert SPI_NSS, then wait for minimum T1 time	SPI_NSS signal is in asserted state	
3	SPI_M → TT(SPI_S)	Initiate SPI access and start data transfer	The data sent from the SPI_M shall be received by the SPI_S in a single SPI access NSD bytes have been added if the SPI access has been longer than the frame being send	RQ0703_014 RQ0703_015

Sequence 3 - Link layer - Master frame generation - MTU = 128 bytes

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ with MTU = 128 bytes (as supported by the SPI_M) after VDD is switched on	The SPI_S confirms the MTU length suggested in the MCT_MASTER_REQ	
2	SPI_M → TT(SPI_S)	Assert SPI_NSS, then wait for minimum T1 time	SPI_NSS signal is in asserted state	
3	SPI_M → TT(SPI_S)	Initiate SPI access and start data transfer	The data sent from the SPI_M shall be received by the SPI_S in a single SPI access NSD bytes have been added if the SPI access has been longer than the frame being send	RQ0703_014 RQ0703_015

Sequence 4 - Link layer - Master frame generation - MTU = 256 bytes

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ with MTU = 256 bytes (as supported by the SPI_M) after VDD is switched on	The SPI_S confirms the MTU length suggested in the MCT_MASTER_REQ	
2	SPI_M → TT(SPI_S)	Assert SPI_NSS, then wait for minimum T1 time	SPI_NSS signal is in asserted state	
3	SPI_M → TT(SPI_S)	Initiate SPI access and start data transfer	The data sent from the SPI_M shall be received by the SPI_S in a single SPI access NSD bytes have been added if the SPI access has been longer than the frame being send	RQ0703_014 RQ0703_015

8.1.1.4 Post Condition

General post-conditions defined in clause 7.0.4 is executed at the end of each sequence.

8.1.2 Link layer - Master frame generation - SPI access longer than frame

8.1.2.1 Test purpose

If the SPI access is longer than the length of the frame being sent, the SPI master shall add Non-Significant Data (NSD) bytes following the CRC until the end of the SPI access.

8.1.2.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.1 is used:

   - The slave is emulated to ensure that the specific parameters given in the SLAVE_EMULATE_STATE are fulfilled.

3) Ensure that the SPI master has no data to send, respectively wait for 1 second after the last data is sent from the SPI master.

4) The SPI slave requests a MAC access with an MTU of 32 bytes.

8.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Assert SPI_NSS, then wait for minimum T1 time	SPI_NSS signal is in asserted state	
2	SPI_M → TT(SPI_S)	Initiate SPI access and send a frame of minimum 1 byte	Non-Significant Data (NSD) bytes have been added following the CRC until the given MTU length is reached	RQ0703_015

8.1.2.4 Post Condition

General post-conditions defined in clause 7.0.4 apply.

8.1.3 Slave frame retrieval by Master

8.1.3.1 Test purpose

A slave frame shall be retrieved in at most two SPI accesses if the number of bytes of the first SPI access is shorter than the slave frame.

8.1.3.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.1 is used:

   - The slave is emulated to ensure that the specific parameters given in the SLAVE_EMULATE_STATE are fulfilled.

3) The slave shall send 250 bytes of data to the master in the MAC phase following next.

8.1.3.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_S)	Assert SPI_NSS for T2 time.		RQ0605_015
2	SPI_M	Assert SPI_NSS after T1 - T2 time	SPI_NSS signal is in asserted state	
3	SPI_M → TT(SPI_S) TT(SPI_S) → SPI_M	SPI_M initiates SPI access and start data transfer of 20 bytes Transfers 10 bytes of data	A master shall receive all the data at most two SPI access without and with NSD bytes. The received frame shall be the same as sent frame	RQ0703_015

8.1.3.4 Post Condition

General post-conditions defined in clause 7.0.4 apply.

8.1.4 Bytes set to 'FF' in second SPI access

FFS.

8.2 Link layer - SPI Slave testing

8.2.1 Slave frame generation - SPI access longer than frame

8.2.1.1 Test Purpose

If the SPI access is longer than the length of the frame being sent, the SPI slave shall add Non-Significant Data (NSD) bytes following the CRC until the end of the SPI access.

8.2.1.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.2 is used:

   - The master is emulated to ensure that the specific parameters given in the MASTER_EMULATE_STATE can be fulfilled.

   - The SPI slave is the SUT.

3) The master shall send 20 bytes of data to slave in the MAC phase following next.

8.2.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_S	Assert SPI_NSS for T2		
2	TT(SPI_M)	Assert SPI_NSS after T1 - T2	SPI_NSS signal is in asserted state	
3	SPI_S → TT(SPI_M)	Master initiate SPI access and start data transfer of 20 bytes Slave transfer 10 bytes of data.	Complete data shall receive in one SPI access and SPI slave shall add Non-Significant Data (NSD) bytes following the CRC until the end of the SPI access.	RQ0703_015

8.2.1.4 Post Condition

General post-conditions defined in clause 7.0.4 apply.

8.2.2 Slave frame retrieval by SPI Master

8.2.2.1 Test purpose

A slave frame shall be retrieved in at most two SPI accesses if the number of bytes of the first SPI access is shorter than the slave frame.

8.2.2.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.2 is used:

   - The master is emulated to ensure that the specific parameters given in the MASTER_EMULATE_STATE can be fulfilled.

   - The SPI slave is the SUT.

3) The slave shall send 250 bytes of data to slave in the MAC phase following next.

8.2.2.3 Test procedure

Sequence 1 - Slave frame retrieval by Master - NSD bytes not appended

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_S	Assert SPI_NSS for T2		RQ0605_015
2	TT(SPI_M)	Asserts SPI_NSS after T1 - T2	SPI_NSS signal is in asserted state	
3	SPI_S → TT(SPI_M)	The SPI_M shall initiate a first SPI access for two bytes and a second SPI access with a length equal to the remaining frame	The SPI_S shall send the complete frame in two SPI accesses, The NSD byte shall not be appended to the second SPI access frame. The received frame shall be the same as the sent frame (identical byte order)	RQ0703_016 RQ0703_018

Sequence 2 - Slave frame retrieval by Master - NSD bytes appended

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_S	Assert SPI_NSS for T2		RQ0605_015
2	TT(SPI_M)	Asserts SPI_NSS after T1 - T2	SPI_NSS signal is in asserted state	
3	SPI_S → TT(SPI_M)	The SPI_M shall initiate a first SPI access for two bytes and a second SPI access shall greater than remaining frame	The slave shall send the complete frame in two SPI accesses The NSD bytes shall be appended to the second SPI access frame. The received frame shall be the same as the sent frame (identical byte order)	RQ0703_016 RQ0703_018

8.2.2.4 Post Condition

General post-conditions defined in clause 7.0.4 apply.

8.3 Link layer data transfer cases - SPI Master testing

8.3.1 Case 2 - Slave MAC access request, retrieve the slave frame

8.3.1.1 Test purpose

The SPI master behaviour is verified. The SPI slave initiates a MAC access request to transfer a frame. The SPI master performs a first access to retrieve the SPI slave frame length followed by a second access to retrieve the remaining bytes of the SPI slave frame considering the length information from the first access.

8.3.1.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.1 is used:

   - The slave is emulated to ensure that specific parameters will be send in the MCT_READY.

   - The SPI master is the SUT.

3) The MAC parameter determination procedure as defined in clause 7.0.2 is executed.

8.3.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_S)	Initiate MAC access		
2	SPI_M →TT(SPI_S)	Establish MAC access	The MAC access is established with an MTU set to 32 bytes	
3	SPI_M → TT(SPI_S)	Generate an SPI access of 1 byte to send the SPI_M frame length information	The SPI_S receives no data (NSD only) on SPI_MOSI, SPI access length = 1 byte	RQ0703_008 RQ0703_020
	TT(SPI_S) → SPI_M	send 1 byte to send the SPI_S frame length information		
4	SPI_M → TT(SPI_S)	Generate an SPI access of 31 bytes Send link layer frame on SPI_MOSI	The SPI_S receives no data (NSD only) on SPI_MOSI, SPI access length = 31 bytes	RQ0703_019 RQ0703_017
	TT(SPI_S) → SPI_M	Send the remaining frame of the slave (31 bytes)	The SPI_M receives the frame	RQ0703_016 RQ0703_024

8.3.1.4 Post conditions

The SPI is deactivated.

8.3.2 Case 3 - Slave frame transferred in a single access, NSD attached

8.3.2.1 Test purpose

The SPI master behaviour is verified. The SPI slave initiates a MAC access request for sending a frame. The SPI master generates an access with length equal to MTU to make sure the SPI slave frame is transferred in a single access. The SPI slave frame is shorter than the access length and SPI slave appends NSD bytes after the end of its frame until the end of the access.

8.3.2.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.1 is used:

   - The slave is emulated to ensure that specific parameters will be send in the MCT_READY.

   - The SPI master is the SUT.

3) The MAC parameter determination procedure as defined in clause 7.0.2 is executed.

8.3.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_S)	Initiate MAC access		
2	SPI_M → TT(SPI_S)	Establish MAC access	The MAC access is established with an MTU set to 32 bytes	
3	SPI_M → TT(SPI_S)	Generate an SPI access of 32 bytes length	The SPI_S receives no data (NSD only) on SPI_MOSI, SPI access length = 32 bytes	RQ0703_008
	TT(SPI_S) → SPI_M	Send link layer frame on SPI_MISO with 1 byte length containing DATA_01 || CRC || FF…FF (X bytes)	The SPI_M receives the frame	RQ0703_015 RQ0703_025

8.3.2.4 Post conditions

The SPI is deactivated.

8.3.3 Case 4 - Slave MAC access request, frame partially retrieved

8.3.3.1 Test purpose

The master behaviour is verified. Slave initiates a slave MAC access request for sending a frame. Consequently, master generates an access with length based on a best estimate. Master retrieves only part of the frame during the first access and will generate a second access to retrieve the remaining bytes of the slave frame. The length of the second access is based on the frame length information retrieved in the prior access. The total number of bytes transferred on MISO over both accesses is less than or equal to the MTU.

8.3.3.2 Initial conditions

FFS.

8.3.3.3 Test procedure

FFS.

8.4 Link layer data transfer cases - SPI Slave testing

8.4.1 Case 1 - Master sends the frame, no data from the slave

8.4.1.1 Test purpose

The SPI slave behaviour is verified. The SPI master initiates the MAC phase and then sends a frame. The SPI access length is determined by the master frame length. No data is received from the slave.

8.4.1.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.2 is used:

   - The master is emulated to ensure that the specific parameters given in the MASTER_EMULATE_STATE can be fulfilled.

   - The SPI slave is the SUT.

3) The MAC parameter determination procedure as defined in clause 7.0.3 is executed.

4) MCT LLC phase is successfully performed with negotiated MTU = 32 bytes.

8.4.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	The SPI_M shall initiate an SPI access by sending MCT_MASTER_REQUEST.	Complete data shall be received in the SPI access.	RQ0703_009 RQ0703_023
	SPI_S → TT(SPI_M)	The slave sends NSD	No data (NSD bytes only) are received by SPI_M	

8.4.1.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

8.4.2 Case 5 - Simultaneous MAC phase initiation, remaining bytes of the slave frame

8.4.2.1 Test purpose

The SPI slave behaviour is verified. Both, SPI master and SPI slave have frames to transfer. The MAC phase is initiated by master and slave simultaneously. The SPI master generates an SPI access with the length of frame it intends to send. According to the length byte sent by the SPI slave only a part of the frame provided by the slave was received. The SPI master and generates a second SPI access to retrieve the remaining bytes of the respective frame.

8.4.2.2 Initial conditions

FFS.

8.4.2.3 Test procedure

FFS.

8.4.3 Case 6 - Simultaneous MAC phase initiation, short slave frame

8.4.3.1 Test purpose

The SPI slave behaviour is verified. Both, SPI master and SPI slave have frames to transfer. The MAC phase is initiated by master and slave simultaneously. The SPI master generates an SPI access with the length of the frame it intends to send. According to the length byte sent by the SPI slave, the slave frame is shorter than the master frame. The SPI slave adds NSD bytes after the end of its link layer frame up to length defined for the SPI access.

8.4.3.2 Initial conditions

FFS.

8.4.3.3 Test procedure

FFS.

8.4.4 Case 7 - Simultaneous MAC phase initiation, single access

8.4.4.1 Test purpose

The SPI slave behaviour is verified. Both, SPI master and SPI slave have frames to transfer. The MAC phase is initiated by master and slave simultaneously. The SPI master generates an SPI access with the length equal to MTU to receive any slave frame occurring at the same time within a single access. Both, master and slave may append NSD bytes after the end of their frames, up to SPI access completion.

8.4.4.2 Initial conditions

FFS.

8.4.4.3 Test procedure

FFS.

9 Test cases for supported LLC layers

9.1 SPI slave supported LLC layers - SPI Slave testing

9.1.1 SHDLC Layer support

9.1.1.1 Test purpose

To verify that the mandatory SHDLC layer is supported by the slave.

9.1.1.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.
2) The test environment shown in Figure 4.2 is used:
   - The master is emulated to ensure that no data is sent after MCT LLC phase completion.
   - The SPI slave is the SUT.
3) The MAC parameter determination procedure as defined in clause 7.0.2 is executed.
4) A MCT phase followed by a MAC access is initiated.

9.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S SPI_S → TT(SPI_M)	Initiate SHDLC link establishment procedure		
2	TT(SPI_M)→ SPI_S	Send the DEFAULT_SPI frame with LLC control field defined for SHDLC in table 7.3 of ETSI TS 103 713 [1]	The SPI_S receives the LPDU with DATA_1D || CRC packet as defined in Figure 7.11 of ETSI TS 103 713 [1]	RQ0704_001 RQ0704_004

9.1.1.4 Post conditions

SPI is deactivated.

9.1.2 CLT Layer support

9.1.2.1 Test purpose

To verify the that the optional CLT layer is supported by the slave if CLT support is indicated.

9.1.2.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that no data is sent after MCT LLC phase completion.

   - The SPI slave is the SUT.

3) The MAC parameter determination procedure as defined in clause 7.0.2 is executed.

4) A MCT phase followed by a MAC access is initiated. Contactless transaction or an alternative triggering has to be initiated for this.

9.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Initiate CLT LLC session		
2	TT(SPI_M) → SPI_S	Send the representative SPI frame with LLC control field defined for CLT in table 7.3 of ETSI TS 103 713 [1]	The SPI_S receives the LPDU with DATA_1D || CRC CLT payload as defined in Figure 7.10 of ETSI TS 103 713 [1].	RQ0704_002 RQ0704_004

9.1.2.4 Post conditions

SPI is deactivated.

9.1.3 MCT Layer support

9.1.3.1 Test purpose

To verify the that the mandatory MCT layer is supported by the slave.

9.1.3.2 Initial conditions

1) General pre-conditions defined in clause 7.0.1 apply.

2) The test environment shown in Figure 4.2 is used.

   - The master is emulated to ensure that no data is sent after MCT LLC phase completion.

   - The SPI slave is the SUT.

3) The MAC parameter determination procedure as defined in clause 7.0.2 is executed.

9.1.3.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ frame	The SPI_S receives the MCT_MASTER_REQ frame	
2	SPI_S → TT(SPI_M)	Send MCT_READY frame	The SPI_M receives MCT_READY frame	RQ0704_003 RQ0704_004

9.1.3.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

10 Test cases for interworking of the LLC layers

FFS.

11 Test cases for MCT LLC

11.1 MCT LLC - SPI Master testing

11.1.1 Behaviour during MCT activation - no MCT_READY response

11.1.1.1 Test purpose

SPI master behaviour is tested during the MCT activation when the slave does not send MCT_READY response within MCT_SLAVE_TIMEOUT.

11.1.1.2 Initial conditions

The test environment shown in Figure 4.1 is used.

- The slave is emulated to ensure that no MCT_READY frame is sent during MCT LLC setup.

11.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	RQ0702_018 RQ0702_038
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The SPI_S receives the MCT_MASTER_REQ	
3	TT(SPI_S)	Wait for a period longer than MCT_SLAVE_TIMEOUT		
4	SPI_M → TT(SPI_S)	Resend MCT_MASTER_REQ at a time greater than T1 and lower than 1 s	The SPI_S receives the MCT_MASTER_REQ frame	RQ0706_019 RQ0706_020 RQ0706_022 RQ0702_030
5	TT(SPI_S)	Wait for a period longer than MCT_SLAVE_TIMEOUT		

Step	Direction	Action/Task	Description/Expectation	REQ
6	SPI_M → TT(SPI_S)	Resend MCT_ MASTER_REQ at a time greater than T1 and lower than 1 s	The SPI_S receives the MCT_MASTER_REQ frame	RQ0706_019 RQ0706_020 RQ0706_022 RQ0702_030
7	TT(SPI_S) → SPI_M	Send MCT_ READY frame with an MCT LPDU length lower than or equal to 29 bytes		
8	SPI_M → TT(SPI_S)	De-assert SPI_NSS	After data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	RQ0702_022 RQ0702_029
9	SPI_M → TT(SPI_S)	Run MAC deactivation		

11.1.1.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

11.1.2 MCT_MASTER_REQ values

11.1.2.1 Test purpose

Some MCT_MASTER_REQ values are read and compared to the IUT options declared by the vendor.

11.1.2.2 Initial conditions

The test environment shown in Figure 4.1 is used.

- The slave is emulated to ensure that the MCT_READY frame is not limiting the data transfer suggested by the master.

11.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_ M → TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	RQ0702_018 RQ0702_038
2	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ frame	The SPI_S receives the MCT_MASTER_REQ frame with Master-specific MCT_DATA field as defined in ETSI TS 103 713 [1], table 7.7: IF O_LOW_POWER_MODE the value of bit 5 and bit 4 of Byte 1 (Capabilities) in MCT_DATA equals to '00' IF O_FULL_POWER_MODE_1 the value of bit 5 and bit 4 of Byte 1 (Capabilities) in MCT_DATA equals to '01' IF O_FULL_POWER_MODE_2 the value of bit 5 and bit 4 of Byte 1 (Capabilities) in MCT_DATA equals to '10' IF O_FULL_POWER_MODE_3 the value of bit 5 and bit 4 of Byte 1 (Capabilities) in MCT_DATA equals to '11'	RQ0706_003 RQ0706_008

Step	Direction	Action/Task	Description/Expectation	REQ
	TT	Analyse MCT_MASTER_REQ	Evaluate the values related to the supported power mode	
3	TT(SPI_S) → SPI_M	Send MCT_READY frame	Confirm the power mode supported	
4	SPI_M → TT(SPI_S)	De-assert SPI_NSS	After data transfer completion and SPI_CLK stop, the master de-asserts SPI_NSS	RQ0702_022 RQ0702_029
5	SPI_M → TT(SPI_S)	Run MAC deactivation		

11.1.2.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

11.2 MCT LLC - SPI Slave testing

11.2.1 MCT activation - corrupted MCT_MASTER_REQ response

11.2.1.1 Test purpose

SPI slave behaviour is tested during the MCT activation when the master sends a corrupted MCT_MASTER_REQ.

11.2.1.2 Initial conditions

The test environment shown in Figure 4.2 is used.

- The master is emulated to ensure that a corrupted MCT_MASTER_REQ_NC frame and a correct MCT_MASTER_REQ can be send.

11.2.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_NC		
3	SPI_S	Wait	No MCT_READY is received by the master within MCT_SLAVE_TIMEOUT	RQ0706_021
4	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ at a time greater than T1		
5	SPI_S → TT(SPI_M)	Send MCT_READY frame	The MCT_ READY frame with an MCT LPDU length lower than or equal to 29 bytes	RQ0706_001 RQ0706_002
6	TT(SPI_M)	De-assert SPI_NSS	After data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	
7	SPI_M → TT(SPI_S)	Run MAC deactivation		

11.2.1.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

11.2.2 MCT_READY values

11.2.2.1 Test purpose

Some MCT_READY values are read and compared to the SUT options declared by the vendor.

11.2.2.2 Initial conditions

The test environment shown in Figure 4.2 is used.

- The master is emulated to ensure that the MCT_MASTER_CONF frame can be sent to not limit the capabilities that are returned in the MCT_READY.

11.2.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M) → SPI_S	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	
2	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_CONF	The SPI_S receives a predefined MCT_MASTER_REQ	
3	SPI_S → TT(SPI_M)	Send MCT_READY frame	The SPI_M receives the MCT_READY frame with an MCT LPDU length lower than or equal to 29 bytes contains the MCT_DATA as defined in table 7.9 of ETSI TS 103 713 [1]: IF O_SLAVE_FLOW_CONTROL the value of bit 4 of Byte 1 (Capabilities) in MCT_DATA is set to '1' If only the default MTU length of 32 bytes is supported, the value of bit 3 and bit 2 of Byte 1 (Capabilities) in MCT_DATA is set to '00' IF O_MTU_64 the value of bit 3 and bit 2 of Byte 1 (Capabilities) in MCT_DATA is set to '01' IF O_MTU_128 the value of bit 3 and bit 2 of Byte 1 (Capabilities) in MCT_DATA is set to '10' IF O_MTU_256 the value of bit 3 and bit 2 of Byte 1 (Capabilities) in MCT_DATA is set to '11'	RQ0706_001 RQ0706_002 RQ0706_004 RQ0706_012 RQ0706_014
	TT	Analyse the MCT_READY frame	Evaluate the values related to MTU length and slave driven flow control	
4	TT(SPI_M) → SPI_S	De-assert SPI_NSS	After the data transfer is completed and SPI_CLK is stopped, the SPI_M de-asserts SPI_NSS	
5	SPI_M → TT(SPI_S)	Run MAC deactivation		

11.2.2.4 Post conditions

The SPI is powered off, i.e. the power supply for the PCB or chip hosting the SPI is switched off.

12 Test cases for SHDLC LLC

12.0 SHDLC LLC Overview

12.0.1 General considerations

SHDLC LLC testing is using endpoints. For the endpoints, definitions from Annex A, clause A.4 apply:

- The SUT may be either the master endpoint or the slave endpoint.
- The TT may be either the master endpoint or the slave endpoint.

NOTE: The requirements for this section are taken from ETSI TS 102 613 [5].

12.1 SHDLC LLC Data handling

12.1.1 Error management - corrupted I-frame

12.1.1.1 Test purpose

12.1.1.2 Initial Conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send a corrupted I-frame(NS0_S,x)		
2	TT → SUT		The TT does not send an acknowledgment	RX1001_002
3	SUT → TT		The SUT waits 10 ms and sends a correct I-frame(NS0_S,x)	
4	TT → SUT		Acknowledge the received I-frame	

12.1.2 Error management - corrupted RR frame

12.1.2.1 Test purpose

12.1.2.2 Initial Conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT		Trigger the TT to send an I-frame	
2	TT → SUT	Send I(NS0_T,x)		RX1001_002
3	SUT → TT	Send a corrupted RR(NS0_T+1) frame		
4	SUT		Wait T2 time and do not acknowledge the received frame	

Step	Direction	Action/Task	Description/Expectation	REQ
5	TT → SUT	Send I(NS0_T,x)		RX1001_002

12.2 SHDLC context

12.2.1 Initial state at link reset - reset by the SUT

12.2.1.1 Test purpose

To ascertain that the SUT is in the correct initial state when the link is reset by the SUT.

12.2.1.2 Initial conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.2.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send RSET(Ws=2, SREJ=0)		
2	TT → SUT	Send UA		
3	Conditional	TT sends I-frame after link establishment or when triggered.	If the TT does not immediately send I-frames after SHDLC link establishment, trigger the TT to send an I-frame. If the trigger involves sending I-frames to the SUT, only one I-frame shall be sent	
4	TT → SUT	Send I-frame (0, NR).	If the trigger in step 3 involved sending an I-frame to the SUT, NR = 1, else NR = 0	RX1006_005
5	SUT → TT	Send RR(1)		
6	Conditional	TT may send more I-frames.	If the TT continues to send I-frames, acknowledge them	
7	SUT → TT	Send I-frame (NS, NR)		
8	TT → SUT	Acknowledge the previously sent I-frame		RX1006_006

12.3 SHDLC sequence of frames

12.3.1 Link establishment by the SUT with default sliding window size

12.3.1.1 Test purpose

To ensure the SUT establishes a link with default sliding window sizes.

12.3.1.2 Initial conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.3.1.3 Test procedure

The test procedure shall only be executed for RSET values, from the following table, that are supported by the SUT.

RSET()
RSET(2)
RSET(3)
RSET(4)
RSET(2, SREJ=0)
RSET(2, SREJ=1)
RSET(3, SREJ=0)
RSET(3, SREJ=1)
RSET(4, SREJ=0)
RSET(4, SREJ=1)

SREJ should be tested only for the biggest window size supported by the SUT.

Sequence 1

Step	Direction	Action/Task	Description/ Expectation	REQ
1	SUT → TT	Send the RSET frame indicated in the test execution clause		
2	TT → SUT	Send UA		RX1007_002 RX1007_003 RX1007_007 RX1005_012 RX1006_014 RX1005_013
3	SUT → TT	Send an I-frame		
4	TT → SUT	Acknowledges the previously sent I-frame		RX1007_008
NOTE 1: If SUT sends I-frames between steps 2 and 3, they shall be acknowledged by the TT. NOTE 2: RX1007_003 is only validated when RSET() is sent in step 1.				

12.3.2 Link establishment and connection time out

12.3.2.1 Test purpose

To ensure the correct time outs are applied by the SUT after a link is established.

12.3.2.2 Initial conditions

None of the SUT contacts is activated.

12.3.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT	Trigger the TT	Activate SPI interface	
2	SUT ←→ TT		Perform SPI interface activation	
3	TT → SUT	Send RSET		RX1007_001
4	SUT		Do not send a UA frame	
5	TT → SUT	Send RSET	After at least T3 time after execution of step 3	RX1006_004
6	SUT → TT	Send an I-frame(0,0)		
7	TT → SUT	Send RSET	After at least T3 time after execution of step 5	RX1006_004
8	SUT → TT	Send UA		
9	SUT → TT	Send an I-frame		
10	TT → SUT		Acknowledge the previously sent I-frame	RX1007_008

12.3.3 Requesting unsupported window size and/or SREJ support - link establishment by SUT

12.3.3.1 Test purpose

To ensure that the SUT accepts correct window size values when the link is reset.

12.3.3.2 Initial conditions

SHDLC link is established.

12.3.3.3 Test procedure

Run sequence 1 for every row in the table below. If the SUT supports window size 4, the first row (RSET()) shall be skipped.

RSET frame to be sent in step 1	Valid RSET frames which can be received in step 2
RSET()	RSET(3) RSET(2) RSET(3, SREJ=0) RSET(2, SREJ=0)
RSET(4, SREJ=1)	RSET() RSET(4) RSET(4, SREJ=0) RSET(3) RSET(3, SREJ=0) RSET(3, SREJ=1) RSET(2) RSET(2, SREJ=0) RSET(2, SREJ=1)

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send the RSET frame	Indicated in the test execution clause for step 1	
2	TT → SUT	Send RSET frame containing values which are supported by the SUT	Values are indicated as valid in the test execution clause for step 2	RX1007_003 RX1005_001 RX1005_010 RX1005_011 RX1005_003
3	SUT → TT		Respond UA	
NOTE: RX1007_003 is only validated when RSET() is sent in step 1.				

12.3.4 Discard buffered frames on link re-establishment

12.3.4.1 Test purpose

To ensure the SUT discards buffer frames when a link is re-established.

12.3.4.2 Initial conditions

1) The SHDLC link is established with SREJ support.

2) SHDLC link is idle, i.e. no further communication expected.

12.3.4.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send I-frame(NS0_S,x)		
2	TT → SUT	Acknowledges I-frame(NS0_S,x)		
3	SUT → TT	Send I-frame(NS0_S+2,x)		
4	TT → SUT	Send SREJ(NS0_S+1)		
5	SUT ←→ TT		Re-establish SHDLC link	
6	SUT ← → TT	SUT sends I-frame(0,NR) to I-frame(NS0_S+1,NR)	TT acknowledges these I-frames	RX1007_005

12.4 Data flow

12.4.1 I-frame transmission

12.4.1.1 Test purpose

To ensure the SUT correctly handles I-frame transmission.

12.4.1.2 Initial conditions

1) SHDLC link is established with the window size indicated in the test execution clause.

2) SHDLC link is idle, i.e. no further communication is expected.

12.4.1.3 Test procedure

Run sequence 1 for:

- Every supported window size:
  - Every I-frame is acknowledged individually by the ES.

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	User → TT	Trigger the TT	Send 9 I-frames	
2	TT → SUT SUT → TT		TT send I-frames as indicated in step 1 SUT acknowledges these frames using the acknowledgement mechanism indicated in the test execution clause, using RR frames.	RX1006_007 RX1006_008 RX1006_009

12.4.2 I-frame reception - single I-Frame reception

12.4.2.1 Test purpose

To ensure the SUT correctly sends a number of I-frames one at a time.

12.4.2.2 Initial conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.4.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT →TT	Send 10 I-frames	Wait for the acknowledgement of the previously sent I-frame before sending the next I-frame	
2	TT → SUT	TT acknowledges these I-frames		RX1007_009 RX1006_008
3	conditional	If retransmission occurs, perform steps 4 and 5		
4	SUT →TT	Send 10 I-frames	Wait for the acknowledgement of the previously sent I-frame before sending the next I-frame	
5	TT → SUT	TT acknowledges these I-frames	TT acknowledges these I-frames, without requiring retransmission by the SUT	RX1007_009 RX1006_008

12.4.3 I-frame reception - multiple I-Frame reception

12.4.3.1 Test purpose

To ensure the SUT correctly sends multiple I-frames with the correct time span.

12.4.3.2 Initial conditions

1) SHDLC link is established with the window size indicated in the test execution clause.

2) SHDLC link is idle, i.e. no further communication is expected.

12.4.3.3 Test procedure

Run sequence 1 for every supported window size.

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send 10 I-frames	The SUT shall send each I-frame within T1, without waiting for the acknowledgement of the previously sent I-frame, while still complying to the negotiated window size. There shall be at least two occurrences of consecutive I-frames transmitted with a single idle bit between the frames.	
2	TT → SUT		TT acknowledges these frames.	RX1007_009 RX1006_008
3	conditional	If retransmission occurs, perform steps 4. and 5.		
4	SUT → TT	Send 10 I-frames	The SUT shall send each I-frame within T1, without waiting for the acknowledgement of the previously sent I-frame, while still complying to the negotiated window size. There shall be at least two occurrences of consecutive I-frames transmitted with a single idle bit between the frames.	
5	TT → SUT		TT acknowledges these frames, without requiring retransmission by the SUT.	RX1007_009 RX1006_008

12.5 Reject (go N back)

12.5.1 REJ transmission - multiple I-frames received

12.5.1.1 Test purpose

To ensure the SUT correctly handles REJ transmission with multiple I-frames received.

12.5.1.2 Initial conditions

1) SHDLC link is established with WS=3 and without SREJ support.
2) SHDLC link is idle, i.e. no further communication is expected.

12.5.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send I-frame(NS0_S,x)		
2	TT → SUT	Acknowledge I-frame(NS0_S,x)		
3	SUT → TT	Send I-frame(NS0_S+2,x) followed immediately by I-frame(NS0_S+3,x)		
4	TT → SUT	Send REJ(NS0_S+1)	The SUT is allowed to send additional REJ(NS0_S+1), in response to any additional I-frame(NS0_S+x,x)	RX1007_012
5	SUT ←→ TT	SUT sends 10 I-frames starting at I-frame(NS0_S+1,x)	TT acknowledge I-frames	RX1007_014

12.5.2 REJ reception

12.5.2.1 Test purpose

To ensure the SUT correctly handles REJ reception.

12.5.2.2 Initial conditions

1) SHDLC link is established without SREJ support.
2) SHDLC link is idle, i.e. no further communication is expected.

12.5.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT	Trigger the TT to send I-frames	The TT shall be triggered such that it streams I-frames, as described in B_STREAM_IFRAMES	
2	TT → SUT	Send I-frame(NS0_TT, y)		
3	SUT		Do not acknowledge I-frame(NS0_TT,y)	
4	TT → SUT	Send I-frame(NS0_TT+1,y)		
5	SUT → TT	Send REJ(NS0_TT)	The SUT is required to send additional REJ(NS0_TT), in response to any additional I-frame(NS0_TT+x,y)	
6	TT → SUT	Send I-frame(NS0_TT,y)		RX1007_013
7	SUT → TT		Acknowledge I-frame(NS0_TT,y)	

Step	Direction	Action/Task	Description/Expectation	REQ
8	TT → SUT	Send I-frame(NS0_TT+1,y)		RX1007_013
9	SUT → TT		Acknowledge I-frame(NS0_TT+1,y).	

12.6 Last Frame Loss

12.6.1 Retransmission of multiple frames

12.6.1.1 Test purpose

To ensure the SUT correctly manages the retransmission of multiple frames.

12.6.1.2 Initial conditions

1) SHDLC link is established without SREJ support.
2) SHDLC link is idle, i.e. no further communication is expected.

12.6.1.3 Test procedure

Run sequence 1 for:

- Every supported window size:
  - I-frames are acknowledged by the ES just before T1 expires and using the maximum allowed value for NR.

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	User → TT	Trigger the TT to send 9 I-frames		
2	TT → SUT	TT send I-frames	As indicated in step 1, respecting the negotiated window size	RX1007_011
3	SUT		SUT does not acknowledge the I-frame(s) within T1	
4	TT → SUT	The TT retransmits the I-frame(s) respecting the window size	After T2 (calculated from the first non-acknowledge frame)	RX1007_015 RX1006_006 RX1007_011
5	SUT → TT		Acknowledges the received I-frame(s) within T1	

12.7 Receive and not ready

12.7.1 RNR reception

12.7.1.1 Test purpose

To ensure the SUT correctly manages RNR reception.

12.7.1.2 Initial conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.7.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT	Trigger the TT	Send 9 **non-empty** I-frames	
2	TT → SUT	Start sending I-frames		
3	SUT → TT		Acknowledge the first received I-frame(NSa_TT,x) with RNR(NSa_TT+1)	
4	SUT		Wait 100 ms	RX1007_018
	TT → SUT		The TT may send further I-frames within the negotiated WS; in this case the SUT should not acknowledge these I-frames	
5	SUT → TT	Send RR, every 5 ms to 20 ms	Until a new I-frame is received where N(R) = NSa_TT+1	
6	TT → SUT SUT → TT	TT sends remaining I-frames, where N(s) of the first I-frame = NSa_TT+1	All of the I-frames shall be non-empty. SUT acknowledges remaining I-frames	RX1007_018

12.7.2 Empty I-frame transmission

12.7.2.1 Test purpose

To ensure the SUT correctly manages empty I-frame transmission.

12.7.2.2 Initial conditions

SHDLC link is established and idle, i.e. no further communication is expected.

12.7.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT	Trigger the TT	Send 1 I-frame	
2	TT → SUT	Send I-frame(NSa_TT,x)		
3	SUT → TT		Acknowledge I-frames(NSa_TT,x) with RNR(NSa_tT+1)	
4	SUT→ TT	Send RR(NSa_TT+1)		
5	TT → SUT	Send empty I-frame(NSa_TT+1,x)		RX1007_017
6	SUT → TT		Send acknowledgement of I-frame(NSa_TT+1)	

12.8 Selective reject

12.8.1 SREJ transmission

12.8.1.1 Test purpose

To ensure the SUT correctly manages SREJ transmission.

12.8.1.2 Initial conditions

1) The SHDLC link is established with SREJ support.

2) SHDLC link is idle, i.e. no further communication is expected.

12.8.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SUT → TT	Send I-frame(NS0_S,x)		
2	TT → SUT		Acknowledge I-frame(NS0_S,x)	
3	SUT → TT	Send I-frame(NS0_S+2,x)		
4	TT → SUT	Send SREJ(NS0_S+1)		RX1008_001
5	SUT → TT	Sends I-frame(NS0_S+1,x)		
6	TT → SUT		Acknowledges I-frame(NS0_S+1,x) and I-frame(NS0_S+2,x)	
7	SUT → TT	Send I-frame(NS0_S+3, x)		
8	TT → SUT		Acknowledges I-frame(NS0_S+3,x)	RX1008_002

12.8.2 SREJ reception

12.8.2.1 Test purpose

To ensure the SUT correctly manages SREJ reception.

12.8.2.2 Initial conditions

1) SHDLC link is established with SREJ support.

2) SHDLC link is idle, i.e. no further communication is expected.

12.8.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT	Trigger the TT	Send 9 I-frames with as small a delay between subsequent I-frames as possible	
2	TT → SUT	Send I-frame(NS0_TT,x)		
3	SUT → TT		Do not acknowledge the received I-frame	
4	TT → SUT		If the TT retransmits I-frame(NS0_TT,x), then stop the test procedure, as it is not possible for the SUT to send a valid REJ. This is not a failure of the TT If the TT transmits I-frame(NS0_TT+1,x), then continue the test procedure	
5	SUT → TT	Send SREJ(NS0_TT)		
6	TT → SUT		Retransmit only the rejected I-Frame and continue sending remaining I-frames SUT acknowledges remaining I-frames	RX1007_020

13 Test cases for power management

13.1 Power management - SPI Master testing

13.1.1 SPI Master entering power saving mode

13.1.1.1 Test purpose

Verify that the slave power source status and capabilities indicated by the master at interface initialization do not change when the master enters into power saving mode or is in power saving mode.

13.1.1.2 Initial conditions

The test environment shown in Figure 4.1 is used.

13.1.1.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1	SPI_M→ TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	
2	SPI_M→ TT(SPI_S)	Initiate data transfer	The SPI_M initiates a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
3	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The TT(SPI_S) receives the MCT_MASTER_REQ	
	TT	Store and interpret the MCT_MASTER_REQ		
4	TT(SPI_S) → SPI_M	Return MCT_READY	The SPI_S sends an MCT_READY	
5	SPI_M	De-assert SPI_NSS		
6	TT(SPI_S) → SPI_M	Allow SPI_M to switch into implementation-specific power saving mode		
7	TT(SPI_S)	Drain current to the maximum defined for power mode supported by the SPI_M for 10 s	The TT acts as an SPI_S draining the maximum current stated to be supported in the MCT_MASTER_REQ	
8	TT(SPI_S)	Decrease the current load to be within the LOW POWER MODE definition		
9	SPI_M→ TT(SPI_S)	Run MAC deactivation	The SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 or clause C.2.2	
10	SPI_M→ TT(SPI_S)	Run MAC activation	The SPI_M runs a MAC activation as defined in Annex C, clause C.1.1 or clause C.2.1	
11	SPI_M→ TT(SPI_S)	Initiate data transfer	The SPI_M initiates a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
12	SPI_M → TT(SPI_S)	Send MCT_MASTER_REQ	The TT(SPI_S) receives the MCT_MASTER_REQ	
	TT	Store and interpret the MCT_MASTER_REQ	The power mode stated to be supported in the MCT_MASTER_REQ shall be identical with the power mode from MCT_MASTER_REQ in step 3.	RQ0708_007 AND/OR RQ0708_008 see note
NOTE: A verification of the power mode stated to be supported in the MCT_MASTER_REQ is possible in the following MCT_MASTER_REQ only, the capability can be checked by using a controlled power drain.				

A differentiation of results between **entering to** power save mode and **in** power save mode is not possible as the MCT_MASTER_REQ cannot be checked in a transition phase or while not running a MAC activation.

13.1.2 SPI Master resuming from power saving mode

13.1.2.1 Test purpose

Verify that a master that has entered into power saving mode, resume when the slave initiates a MAC access request.

Following a resume by a MAC access request during T2 as described in clause 7.2.2.3 of ETSI TS 103 713 [1], the master shall start an SPI access after T1 or later, from the leading edge of the slave MAC access request pulse, i.e. ETSI TS 103 713 [1], clauses 7.2.3.2 and 7.2.4.3 apply. Where the leading edge of the MAC access request (SPI_NSS or SPI_INT assertion) should trigger the resuming of the master.)

13.1.2.2 Initial conditions

The test environment shown in Figure 4.1 is used.

13.1.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1 to 6	Execute step1. To step 6. of test case 13.1.1/Sequence 1 to ensure that the SPI_M is in power saving mode			
7	SPI_M	Set to all signals driven by SPI_M to idle state corresponding to SPI mode 0		
8	TT(SPI_S) → SPI_M	Assert SPI_NSS (4 signals SPI) Assert SPI_INT (5 signals SPI)	The emulated SPI_S initiates a MAC access request for T2 (1 µs)	RQ0708_016*
9	TT(SPI_S)	De-assert SPI_NSS	The SPI_S switches to configured/de-selected mode	
10	SPI_M	Resume from power saving mode		RQ0708_015
11	SPI_M → TT(SPI_S)	Initiate SPI access after T1	The SPI_S is ready to receive data	RQ0708_017
12	SPI_M → TT(SPI_S)	Transfer data	The SPI_S receives data	
13	TT(SPI_S) → SPI_M	Transfer data	The SPI_S sends data	
14	SPI_M → TT(SPI_S)	Run MAC deactivation	A 5 signals SPI_M runs a MAC deactivation as defined in Annex C, clause C.1.2 A 4 signals SPI_M runs a MAC deactivation as defined in Annex C, clause C.2.2	

NOTE: Requirement RQ0708_016 is implicitly tested if the MAC access request is initiated.

13.2 Power management - SPI Slave testing

13.2.1 SPI Slave entering power saving mode

13.2.1.1 Test purpose

In order to optimize the power consumption, both the master and the slave may enter into power saving mode.

Verify that a slave is entering into power saving mode if there is no pending activity and one of the following conditions is fulfilled:

- All frames have been transmitted by the slave, acknowledged successfully by the master and the slave detects an inactivity time T4 with no assertion of SPI_NSS by the master.
- The slave receives the SHDLC link layer acknowledgement for EVT_LINK_END_OF_OPERATION.
- The slave does not detect any activity for more than the default inactivity period MCT_MASTER_TIMEOUT following the power-up or during MCT activation procedure.

13.2.1.2 Initial conditions

The test environment shown in Figure 4.2 is used.

13.2.1.3 Test procedure

Sequence 1 - Pending slave MAC access request

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Activate VDD	Power-up of the SPI	
2	TT(SPI_M) → SPI_S	Initiate data transfer with SPI_CLK set to 1 MHz	First the SPI_M runs a MAC activation as defined in Annex C, clause C.1.2 or C.2.2 then the initiation of a data transfer as defined in Annex C, clause C.1.3 or C.2.3	
3	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_PSM_Y after initial POT (1 s)	The SPI_S receives the MCT_MASTER_REQ_PSM_Y	RQ0708_003
4	SPI_S → TT(SPI_M)	Return MCT_READY	The SPI_S sends an MCT_READY	RQ0708_003
5	TT(SPI_M) → SPI_S	De-assert SPI_NSS		
6	SPI_S → TT(SPI_M)	Issue MAC access request	Assert SPI_NSS (4 signals SPI) or SPI_INT (5 signals SPI)	
7	TT(SPI_M)	No assertion of SPI_NSS within T4		
	TT	Measure IOH on VDD	The SPI_S is in configured/de-selected mode	
8	SPI_S	No switch to Power saving mode after T4	After T4 the SPI_S shall not switch to power saving mode	RQ0708_002
	TT	Measure IOH on VDD for additional 30 s	IOH shall not decrease	
9	TT(SPI_M) → SPI_S	Deactivate VDD	The emulated SPI_M deactivates VDD	

Sequence 2 - No assertion of SPI_NSS by the master for a time > T4

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Activate VDD	Power-up of the SPI	
2	TT(SPI_M) → SPI_S	Initiate data transfer with SPI_CLK set to 1 MHz	First the SPI_M runs a MAC activation as defined in Annex C, clause C.1.2 or C.2.2 then the initiation of a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
3	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_PSM_Y after initial POT (1 s)	The SPI_S receives the MCT_MASTER_REQ_PSM_Y	RQ0708_003

Step	Direction	Action/Task	Description/Expectation	REQ
4	SPI_S → TT(SPI_M)	Return MCT_READY	The SPI_S sends an MCT_READY	RQ0708_003

	TT	Verification of MCT_READY data	The MCT_READY data is verified:
			IF (T3 ≤ T1) the SPI_S may not enter power saving mode for system related reasons or it may "self-resume"
			IF (T4 in MCT_READY = "FFFF") the SPI_S is not entering power saving mode
			THEN
			the SPI can be deactivated and the test case can be ended
			ELSE
			Assume the SPI_S return value for T4 is set as the inactivity period time
			The time stamp for the last de-assertion of SPI_NSS is stored in the TT
				RQ0708_005

	TT	Provide information about T1, T3 and T4	T1, T3 and T4 values are provided to the user to verify the TT and the SPI_S behaviour	
	TT	Measure IOH on VDD	The SPI_S is in configured/de-selected mode	
5	SPI_S	Switch to Power saving mode after T4	After T4 the SPI_S may switch to power saving mode	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (on a 5 signal SPI)	IF (SPI_S is in power saving mode) neither SPI_MISO is activated nor SPI_NSS (on a 4 signal SPI) or SPI_INT (on a 5 signal SPI) are asserted	
	TT	Measure IOH on VDD for additional 30 s	IF (SPI_S is in power saving mode) IOH shall be lower than IOH measured in configured/deselected mode	
6	TT(SPI_M)	Deactivate VDD	The emulated SPI_M deactivates VDD	

Sequence 3 - Power saving mode declined by the master (T4 = "FFFF")

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Activate VDD	Power-up of the SPI	
2	TT(SPI_M) → SPI_S	Initiate data transfer with SPI_CLK set to 1 MHz	First the SPI_M runs a MAC activation as defined in Annex C, clause C.1.2 or C.2.2 then the initiation of a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
3	TT(SPI_M) → SPI_S	Send MCT_MASTER_REQ_DEF frame after initial POT (1 s)	The SPI_S receives the MCT_MASTER_REQ_DEF frame prepared by the TT	RQ0708_003
4	SPI_S → TT(SPI_M)	Return MCT_READY	The SPI_S sends an MCT_READY confirming T4 = "FFFF"	RQ0708_003
	TT	Verification of MCT_READY data	The MCT_READY data is verified and stored in the TT IF T4 in MCT_READY = "FFFF" the SPI_S is not entering power saving mode; the SPI can be deactivated ELSE T4 returned is incorrect	RQ0708_004
5	TT(SPI_M)	Deactivate VDD	The emulated SPI_M deactivates VDD	

Sequence 4 - SHDLC link layer acknowledgement for EVT_LINK_END_OF_OPERATION

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Activate VDD	Power-up of the SPI	
2	TT(SPI_M) → SPI_S	Initiate data transfer with SPI_CLK set to 1 MHz	First the SPI_M runs a MAC activation as defined in Annex C, clause C.1.2 or clause C.2.2 then the initiation of a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
3	TT(SPI_M) → SPI_S	Initiate MAC access		
4	TT(SPI_M) → SPI_S	Initiate SHDLC date transfer		
5	SPI_S → SPI_M	Send EVT_LINK_END_OF_OPERATION		
	TT	Measure IOH on VDD	The SPI_S is in configured/selected mode	
6	TT(SPI_M) → SPI_S	Acknowledge EVT_LINK_END_OF_OPERATION		
7	SPI_S	Switch to power saving mode	Switching to power saving mode	
	TT	Trace activities on SPI_MISO, SPI_NSS and SPI_INT (on a 5 signal SPI)	IF (SPI_S is in power saving mode) neither SPI_MISO is activated nor SPI_NSS (on a 4 signal SPI) or SPI_INT (on a 5 signal SPI) are asserted	
	TT	Measure IOH on VDD for additional 30 s	IF (SPI_S is in power saving mode) IOH shall be lower than IOH measured in configured/selected mode	
8	TT(SPI_M)	Deactivate VDD	The emulated SPI_M deactivates VDD	

Sequence 5 - Inactivity period ≥ MCT_MASTER_TIMEOUT following the power-up

Step	Direction	Action/Task	Description/Expectation	REQ
1	TT(SPI_M)	Activate VDD	Power-up of the SPI	
2	TT(SPI_M) → SPI_S	Initiate data transfer with SPI_CLK set to 1 MHz	First the SPI_M runs a MAC activation as defined in Annex C, clause C.1.2 or C.2.2 then the initiation of a data transfer as defined in Annex C, clause C.1.3 or clause C.2.3	
	TT	Measure IOH on VDD	The SPI_S is in configured/selected mode	
3	TT(SPI_M) → SPI_S	Do not send any MCT_MASTER_REQ for at least MCT_MASTER_TIMEOUT (1 s) after initial POT (1 s)	The SPI_S does not receive any MCT_MASTER_REQ	(RQ0708_002)
	SPI_S	Switch to power saving mode	The SPI_S may switch to power saving mode	
	TT	Measure IOH on VDD for additional 30 s	If the SPI_S has switched to power saving mode IOH shall have decreased	
4	TT(SPI_M)	Deactivate VDD	The emulated SPI_M deactivates VDD	

13.2.2 SPI Slave resuming from power saving mode

13.2.2.1 Test purpose

Before resuming the slave from power saving mode any of the conditions defined for the slave to enter into power saving mode has been previously met.

The master ensures that all signals it drives are in the idle state corresponding to SPI mode 0. The leading edge of the SPI_NSS assertion shall trigger the slave to resume.

13.2.2.2 Initial conditions

The test environment shown in Figure 4.2 is used.

13.2.2.3 Test procedure

Sequence 1

Step	Direction	Action/Task	Description/Expectation	REQ
1 to 5	Execute step1. To step 5. Of test case 13.1.1/Sequence 2 to ensure that the SPI_M is in power saving mode			RQ0708_012
6	TT(SPI_M)	Set to all signals driven by SPI_M to idle state corresponding to SPI mode 0		RQ0708_013
	TT	Continue measuring IOH on VDD		
7	TT(SPI_M) → SPI_S	Assert SPI_NSS	The SPI_S switches from power saving mode to configured/selected mode	RQ0708_012
	TT	Continue measuring IOH on VDD	With switching to configured/selected mode the IOH is increasing	
8	TT(SPI_M) → SPI_S	Wait for T3 keeping SPI_NSS asserted	The SPI_S gets ready to receive data	RQ0708_003
9	TT(SPI_M) → SPI_S	Operate SPI_CLK with 1 MHz Start data transfer	The SPI_S is ready to receive data	RQ0708_014
10	TT(SPI_M) → SPI_S	De-assert SPI_NSS	The SPI_S switches to configured/de-selected mode	
11	SPI_M	Deactivate VDD	The emulated SPI_M deactivates VDD	

NOTE: Requirement RQ0708_014 is a requirement to the master. When executing this test sequence, the emulated master is showing the described behaviour. A verification of the TT is out of scope of this test.

Annex A (normative):
SPI Test tool functional requirements

A.1 General requirements

A.1.0 Introduction

The SPI test tool shall implement the functions of the SPI interfaces as described in ETSI TS 103 713 [1].

For SHDLC tests the SHDLC LLC definition from ETSI TS 102 613 [5] apply.

A.1.1 VDD, VSS

A.1.1.1 Default measurement uncertainties

Unless stated otherwise below, the following uncertainties apply:

- Voltage measurement uncertainty: < ±100 mV
- Time measurement uncertainty: < ±100 ns
- Current Load Amplitude: 0 mA to 20 mA
- Adjustable Step Size: 100 µA
- Uncertainty: < ±100 µA
- Additional Current Offset: 0 mA to 5 mA
- Adjustable Step Size: 100 µA
- Uncertainty: < ±100 µA

A.2 5 signal SPI

A.2.1 SPI_MOSI, SPI_MISO, SPI_NSS

A.2.1.1 Default measurement/setting uncertainties

A.2.1.1.0 Introduction

Unless stated otherwise below, the following measurement uncertainties apply:

- Voltage measurement uncertainty: < ±25 mV
- Time measurement uncertainty: < ±10 ns

A.2.1.1.1 SPI_MOSI

- Voltage setting uncertainty: < ±25 mV
- Rise and fall Time setting uncertainty: < ±100 ns

A.2.1.1.2 SPI_MISO

- Voltage setting uncertainty: < ±25 mV
- Rise and fall Time setting uncertainty: < ±100 ns

A.2.1.1.3 SPI_NSS

- Voltage setting uncertainty: < ±25 mV
- Rise and fall Time setting uncertainty: < ±100 ns

A.2.2 SPI_CLK

A.2.2.1 Default measurement/setting uncertainties

Unless stated otherwise below, the following measurement uncertainties apply for frequencies up to 10 MHz:

- Voltage measurement uncertainty: < ±25 mV
- Frequency measurement uncertainty: < ±0,5 %
- Rise and fall time measurement uncertainty: < ±2,5 ns

Duty cycle:

- Measurement range: 35 % to 65 %
- Measurement uncertainty: < ±2,5 %

Unless stated otherwise below, the following setting uncertainties apply for frequencies up to 10 MHz:

- Voltage setting uncertainty: < ±25 mV
- Rise and fall Time setting uncertainty: < ±2,5 ns
- Frequency setting uncertainty: < ±1 %

SPI operation with frequencies > 10 MHz are FFS

A.2.3 Contact SPI_INT

A.2.3.1 Default measurement/setting uncertainties

Unless stated otherwise below, the following uncertainties apply:

- Voltage measurement uncertainty: < ±25 mV
- Time measurement uncertainty: < ±10 ns
- Voltage setting uncertainty: < ±25 mV
- Rise and fall Time setting uncertainty: < ±100 ns

A.3 SPI bus with 4 signals

For a 4 signals SPI the measurement and setting uncertainties listed for the 5 signals SPI except for the non-existent SPI_INT.

A.4 Endpoints

SHDLC communication occurs between two endpoints. Those endpoints can be assigned to either an SPI Master or SPI Slave. Endpoints in SHDLC tests are identified as SUT and TT. There is no priority of traffic.

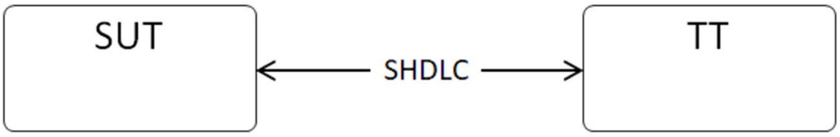

Figure A.4.1: Endpoints

Annex B (normative): Standard frames

B.1 Master frames

Master frames to be used by a **simulated** master:

Name	Content
MCT_MASTER_REQ_DEF	1D || 22 08 08 FF FF …FF || CRC
MCT_MASTER_REQ_64	1D || 22 08 0A FF FF …FF || CRC
MCT_MASTER_REQ_128	1D || 22 08 0C FF FF …FF || CRC
MCT_MASTER_REQ_256	1D || 22 08 0E FF FF …FF || CRC
MCT_MASTER_REQ_PSM_Y	1D || 22 08 08 75 30 FF…FF || CRC
MCT_MASTER_REQ_CONF	1D || 22 08 1E 47 10 FF … FF || CRC
MCT_MASTER_REQ_NC	1D || 20 00 00 00 00 FF … FF || CRC $_x$ (-1)

MCT_MASTER_REQ_DEF	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 32 bytes:	00
Flow control - SHDLC based:	0
T4 - No power save mode:	1111111111111111

MCT_MASTER_REQ_64	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 64 bytes:	01
Flow control - SHDLC based:	0
T4 - No power save mode:	1111111111111111

MCT_MASTER_REQ_128	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000

Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 128 bytes:	10
Flow control - SHDLC based:	0
T4 - No power save mode:	1111111111111111

MCT_MASTER_REQ_256	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU -256 bytes:	11
Flow control - SHDLC based:	0
T4 - No power save mode:	1111111111111111

MCT_MASTER_REQ_PSM_Y	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 32 bytes:	00
Flow control - SHDLC based:	0
T4 - 30000 ms:	0111010100110000

MCT_MASTER_REQ_CONF	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - test value: Full Power Mode 3	11
MTU - test value: 256 bytes:	11
Flow control - test value: SHDLC based	0
T4 - PSM test value: 10000 ms	0010011100010000

MCT_MASTER_REQ_NC	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000

MCT PAYLOAD	
Spec. Version:	
Major version:	00000
Minor version:	000
Capabilities:	
RFU:	000
Power mode - test value: Low Power Mode	00
MTU - test value: 32 bytes:	00
Flow control - test value: SHDLC based	0
T4 - PSM test value: 0 ms	0000000000000000

B.2 Slave frames

Slave frames to be used by a **simulated** slave:

Name	Content
MCT_READY_DEF	1D || 20 08 09 01 FF FF FF FF FF || CRC
MCT_READY_PSM	1D || 20 08 09 01 80 80 27 10 FF || CRC
MCT_READY_64	1D || 20 08 0B 01 FF FF FF FF FF || CRC
MCT_READY_128	1D || 20 08 0D 01 FF FF FF FF FF || CRC
MCT_READY_256	1D || 20 08 0F 01 FF FF FF FF FF || CRC
MCT_READY_CONF	1D || 20 08 <000x> <xyyz> 0A 64 64 27 10 0A || CRC
MCT_READY_NC	1D || 22 00 00 00 00 00 00 00 00 || CRC $_x$ (-1)

MCT_READY_DEF	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 32 bytes:	00
Flow control - SHDLC based:	1
SPI_CLK (MHz):	00000001
T1 - MAC ready time (µs):	11111111
T3 - Slave resume time from PSM (µs):	11111111
T4 - PSM (ms) - no power save mode:	1111111111111111
POT (ms):	11111111

MCT_READY_PSM	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 32 bytes:	00
Flow control - SHDLC based:	1

SPI_CLK (MHz):	00000001
T1 - MAC ready time (µs) - 128 µs:	10000000
T3 - Slave resume time from PSM (µs) - =T1:	10000000
T4 - PSM (ms) - 10000 ms:	0010011100010000
POT (ms):	11111111

MCT_READY_64	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 64 bytes:	01
Flow control - SHDLC based:	1
SPI_CLK (MHz):	00000001
T1 - MAC ready time (µs):	11111111
T3 - Slave resume time from PSM (µs):	11111111
T4 - PSM (ms) - no power save mode:	1111111111111111
POT (ms):	11111111

MCT_READY_128	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01
MTU - 128 bytes:	10
Flow control - SHDLC based:	1
SPI_CLK (MHz):	00000001
T1 - MAC ready time (µs):	11111111
T3 - Slave resume time from PSM (µs):	11111111
T4 - PSM (ms) - no power save mode:	1111111111111111
POT (ms):	11111111

MCT_READY_256	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Full Power Mode 1:	01

MTU -256 bytes:	11
Flow control - SHDLC based:	1
SPI_CLK (MHz):	00000001
T1 - MAC ready time (µs):	11111111
T3 - Slave resume time from PSM (µs):	11111111
T4 - PSM (ms) - no power save mode:	1111111111111111
POT (ms):	11111111

MCT_READY_DEF	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_READY:	00000
MCT PAYLOAD	
Spec. Version:	
Major version:	00001
Minor version:	000
Capabilities:	
RFU:	000
Power mode - <master value>:	xx
MTU - <master value>:	yy
Flow control - <master value>:	z
SPI_CLK; test value: 10 (MHz)	00001010
T1 - test value: 100 (µs)	01100100
T3 - test value: 100 (µs)	01100100
T4 - test value: 10000 (ms)	0010011100010000
POT test value: 10 (ms):	00001010

MCT_READY_NC	
MCT LLC CONTROL FIELD	
LLC CONTROL FIELD - MCT Type:	001
MCT TYPE - MCT_MASTER_REQ:	00010
MCT PAYLOAD	
Spec. Version:	
Major version:	00000
Minor version:	000
Capabilities:	
RFU:	000
Power mode - Low power mode:	00
MTU - 32 byte:	00
Flow control - no flow control:	0
SPI_CLK; test value: 0 (MHz)	00000000
T1 - test value: 0 (µs)	00000000
T3 - test value: 0 (µs)	00000000
T4 - test value: 0 (ms)	0000000000000000
POT test value: 0 (ms):	00000000

B.3 Standard data

Name	Content
DATA_1D	01 01 01… 01 (29 bytes)
DATA_01	01 (1 byte)
DATA_BIG	01 02 03 … 23 (35 bytes)

Annex C (normative): Definition of procedures used in test sequences

C.1 Definition of procedures used with a 5 signals SPI

C.1.1 MAC activation

In accordance to ETSI TS 103 713 [1], clause 7.2.3.4 the MAC activation procedure used to execute test cases on a 5 signal SPI in the context of the present document shall be the following:

1) The master shall set the SPI_MOSI to high impedance and the SPI_CLK at the low level. Master SPI_NSS output shall be set to high impedance.

2) The master shall drive the VDD power line ON.

C.1.2 MAC deactivation

In accordance to ETSI TS 103 713 [1], clause 7.2.3.5 the MAC deactivation procedure used to execute test cases on a 5 signal SPI in the context of the present document shall be the following:

1) The master shall set the SPI_MOSI to high impedance and the SPI_CLK at the low level.

2) Master SPI_NSS output shall be set to high impedance.

3) The master shall drive the VDD power line OFF.

C.1.3 Initiation of the data transfer from the master

In accordance to ETSI TS 103 713 [1], clause 7.2.3.1 and in the context of the present document the master of a 5 signal SPI shall run the following procedure to initiate a data transfer:

1) The Master asserts SPI_NSS.

2) The Master waits for min T1 seconds then goes to step 3. In use-cases when it is certain that the slave is not expected to initiate a MAC access request by SPI_INT assertion (e.g. at first access during SPI initialization) the master may skip this step.

3) The Master starts the bidirectional data transfer by toggling the SPI_CLK signal.

4) The Master de-asserts SPI_NSS after data transfer completion i.e. SPI_CLK stopped.

C.1.4 Initiation of the data transfer from the slave

In accordance to ETSI TS 103 713 [1], clause 7.2.3.2 and in the context of the present document the slave of a 5 signal SPI shall run the following procedure to initiate a data transfer:

1) If the slave determines that the SPI_NSS signal is de-asserted (i.e. at the high level) by reading SPI_NSS_SI, then the slave goes to step 2.

2) The slave asserts SPI_INT by generating an SPI_INT pulse with a minimum width of T2 seconds then goes to step 3.

3) Depending on its implementation, the slave configures its SPI module. The slave waits for data transfer from the master.

4) As a consequence of SPI_INT assertion by the slave at step 2., the master starts data transfer at a time greater than T1 following the leading edge of SPI_INT, by asserting SPI_NSS and then starts SPI_CLK. At data transfer completion, the master enters step 5.

5) After data transfer completion and SPI_CLK stop, the master de-asserts SPI_NSS.Master asserts SPI_NSS.

C.1.5 Simultaneous initiation of a data transfer from both master and slave

In accordance to ETSI TS 103 713 [1], clause 7.2.3.3 and in the context of the present document the simultaneous initiation of a data transfer on a 5 signal SPI shall run the following procedure:

1) Both endpoints request a data transfer and run simultaneously their respective procedures:

   - From the master perspective the resulting procedure is equivalent to the initiation from the master (clause C.1.3).

   - The slave makes a MAC access request by asserting SPI_INT according to the procedure described in clause C.1.4. and waits for the master to generate the access for data transfer.

2) The gap time T1 - T2 shall be long enough for the slave to prepare the SPI module for the data transfer.

C.2 Definition of procedures used with a 4 signals SPI

C.2.1 MAC activation

In accordance to ETSI TS 103 713 [1], clause 7.2.4.6 the MAC activation procedure used to execute test cases on a 4 signal SPI in the context of the present document shall be the following:

1) The master shall set the SPI_MOSI to high impedance and the SPI_CLK at the low level. The SS_MO shall be at the low level, then driving the SPI_NSS output to high impedance.

2) The master shall drive the VDD power line ON.

C.2.2 MAC deactivation

In accordance to ETSI TS 103 713 [1], clause 7.2.4.7 the MAC deactivation procedure used to execute test cases on a 4 signal SPI in the context of the present document shall be the following:

1) The master shall set the SPI_MOSI to high impedance and the SPI_CLK at the low level.

2) The SS_MO shall be at low level then driving the SPI_NSS output to high impedance.

3) The master shall drive the VDD power line OFF.

C.2.3 Initiation of the data transfer from the master

In accordance to ETSI TS 103 713 [1], clause 7.2.4.2 and in the context of the present document the master of a 4 signal SPI shall run the following procedure to initiate a data transfer:

1) The master checks the state of the SPI_NSS signal (by reading SS_MI) and if it is de-asserted (i.e. at the high state) the Master goes to the step 2 otherwise loops on the step 1.

2) The master asserts SS_MO (to drive SPI_NSS signal to the asserted state) then goes to the step 3.

3) The master waits for T1 seconds then goes to the step 4. If the master is certain that the slave will not initiate a MAC access request by asserting SPI_NSS then the master may skip this step.

4) The master starts the bidirectional data transfer by toggling the SPI_CLK signal.

5) After data transfer completion i.e. SPI_CLK stop, the master de-asserts SPI_NSS.

C.2.4 Initiation of the data transfer from the slave

In accordance to ETSI TS 103 713 [1] clause 7.2.4.3 and in the context of the present document the slave of a 4 signal SPI shall run the following procedure to initiate a data transfer:

1) If the SPI_NSS signal is found de-asserted i.e. at high level by reading SS_SI then the slave goes to step 2.

2) The slave disables its SPI module then goes to step 3.

3) The slave asserts SS_SO to drive SPI_NSS to the asserted state (i.e. low level) for at least T2 seconds then goes to step 4.

4) The slave enables its SPI module and waits for the master to initiate the data transfer.

5) The SS_MI signal generates an internal interrupt for the master, triggered on SPI_NSS falling edge. This interrupt initiates a data transfer procedure: the master asserts SS_MO after at least T1 following the SPI_NSS assertion by the slave for the MAC access request.

6) SPI_CLK starts after the SPI_NSS assertion by the master in the previous step, data transfer is performed.

7) After data transfer completion i.e. SPI_CLK stop, the master de-asserts SPI_NSS.

C.2.5 Simultaneous initiation of a data transfer from both master and slave

In accordance to ETSI TS 103 713 [1], clause 7.2.4.4 and in the context of the present document the simultaneous initiation of a data transfer on a 4 signal SPI shall run the following procedure:

1) Both endpoints initiate a MAC phase for data transfers and run simultaneously their respective procedures:

   - From the master perspective the resulting procedure is equivalent to the initiation from the master (clause C.2.3).

   - The slave initiates a MAC access request and waits for the access start from the master as per the timings defined. The time T1 - T2 shall be long enough for the slave to enable the SPI module.

C.2.6 Slave driven flow control

In accordance to ETSI TS 103 713 [1], clause 7.2.4.5 and in the context of the present document the slave driven flow control on a 4 signal SPI shall run the following procedure:

1) For flow control the slave asserts SPI_NSS (via its SS_SO) during a data transfer in progress (e.g. when the slave needs to delay a subsequent master access as the slave estimates it may not be ready to receive or transmit data):

   - It is assumed that the slave has already prepared data to be sent (if available) for the current data transfer before the data transfer started. Such an assertion of SS_SO while the master access is in progress is not a slave MAC access request and a slave shall not start to send a new frame in the middle of the current access, as a new frame shall be always aligned with the start of an SPI access.

   - As long as the SPI_NSS signal is asserted and even if the data transfer is completed, the master shall not run the MAC initiation by the master procedure as defined in clause C.2.1.

   - The slave may assert SS_SO for flow control at any time between the start of the data transfer and SPI_NSS de-assertion by master at end of the data transfer.

NOTE: Detection by the slave of a data transfer for initiating flow control could be performed either by sensing the assertion of the SPI_NSS by master or by detection of the first bytes transferred.

- The slave should not use that mechanism longer than 500 µs; after that duration, the master may use its own recovery mechanisms.

Annex D (informative): Change History

The table below indicates all changes that have been incorporated into the present document since it was published.

Change history								
Date	**Meeting**	**Plenary Doc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
17/12/2020	SCP#97	SCP(20)000172	-	-	-	Version 15.0.0 first publication	-	15.0.0

History

Document history		
V15.0.0	February 2021	Publication