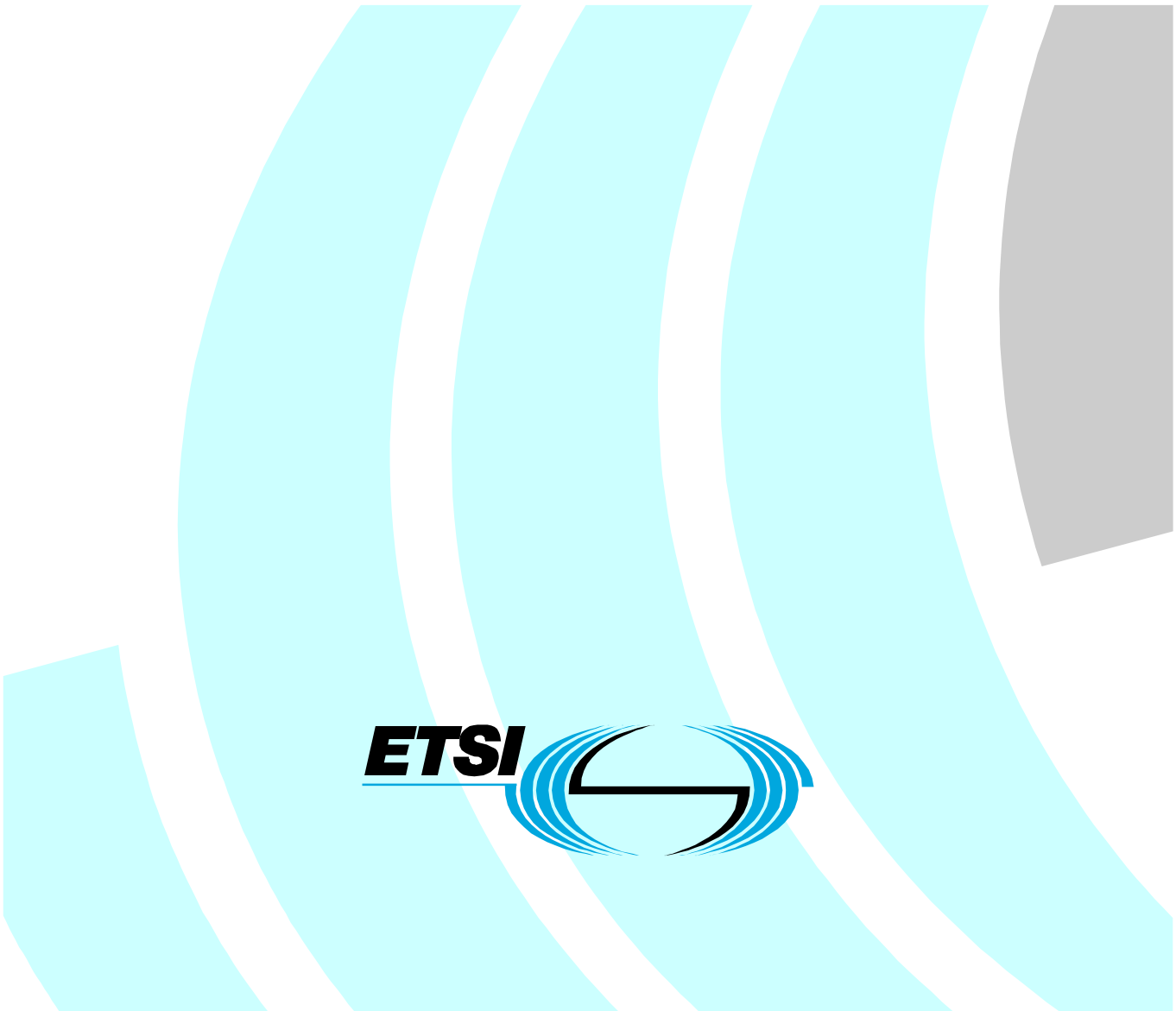
ETSI TS 102 362-3 V1.1.1 (2005-06)

Technical Specification

Electromagnetic compatibility and Radio spectrum Matters (ERM); Conformance testing for the Digital Mobile Radio (DMR); Part 3: Abstract Test Suite (ATS) specification



Reference
DTS/ERM-TGDMR-053-3

Keywords
ATS, digital, PMR, radio

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

IPRs essential or potentially essential to the present document 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 (http://webapp.etsi.org/IPR/home.asp).

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.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Electromagnetic compatibility and Radio spectrum Matters (ERM).

The present document is part 3 of a multi-part deliverable covering the Electromagnetic compatibility and Radio spectrum Matters (ERM); Conformance testing for the Digital Mobile Radio (DMR), as identified below:

Part 1: "Protocol Implementation Conformance Statement (PICS) proforma";

Part 2: "Test Suite Structure and Test Purposes (TSS&TP) specification";

Part 3: "Abstract Test Suite (ATS) specification".

1 Scope

The present document contains the Abstract Test Suite (ATS) to test the ERM DMR Call Control (CCL) and Data Link (DLL) layer.

The objective of the present document is to provide a basis for conformance tests for ERM DMR equipment giving a high probability of air interface inter-operability between different manufacturer's ERM DMR equipment.

The ISO standard for the methodology of conformance testing (ISO/IEC 9646-1 [3] and the ETSI rules for conformance testing (ETS 300 406 [6]) are used as a basis for the test methodology.

Clause 4 describes the Test Configuration used to test the DMR Call Control Layer (CCL) at the MS side and at the BS side.

Clause 5 describes the Test Configurations used to test the DMR Data Link Layer (DLL) at the MS side and at the BS side.

Clause 6 describes the ATS conventions, which are intended to give a better understanding of the ATS.

Annex A provides a guideline for Upper Tester implementation, Inhouse Testing and Send/Receive of DLL TDMA bursts.

Annex B provides the Tree and Tabular Combined Notation (TTCN-3) part of the ATS.

Annex C provides the Partial Protocol Implementation Extra Information for Testing (PIXIT) Proforma of DMR.

Annex D provides the Protocol Conformance Test Report (PCTR) Proforma of DMR.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

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

[1] ETSI TS 102 361-1: "Electromagnetic compatibility and Radio spectrum Matters (ERM); Technical Requirements for Digital Mobile Radio (DMR); Part 1: Air Interface (AI) protocol".

[2] ETSI TS 102 361-2: "Electromagnetic compatibility and Radio spectrum Matters (ERM); Technical Requirements for Digital Mobile Radio (DMR); Part 2: DMR voice and generic services and facilities".

[3] ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[4] ISO/IEC 9646-6: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 6: Protocol profile test specification".

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

[6] ETSI ETS 300 406: "Methods for testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[7] ETSI ES 201 873-1: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language".

[8] ETSI ES 201 873-2: "Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 2: TTCN-3 Tabular presentation Format (TFT)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions defined in ISO/IEC 9646-7 [5], TS 102 361-1 [1], TS 102 361-2 [2] and the following apply:

Lower DLL: all functions which are not part of upper DLL functions, like framing, interleaving and bit ordering

Upper DLL: DLL functions for DLL PDU management and DLL signalling

3.2 Abbreviations

For the purposes of the present document, the abbreviations defined in ISO/IEC 9646-1 [3], ISO/IEC 9646-7 [5], TS 102 361-1 [1], TS 102 361-2 [2] and the following apply:

AI	DMR Air Interface
ATS	Abstract Test Suite
CCL	Call Control Layer
DLL	Data Link Layer
IUT	Implementation Under Test
MTC	Main Test Component
PCTR	Protocol Conformance Test Report
PICS	Protocol Implementation Conformance Statement
PIXIT	Partial Protocol Implementation Extra Information for Testing
PTC	Parallel Test Component
SUT	System Under Test
TC	Test Case
TP	Test Purpose
TRI	TTCN-3 Runtime Interface
TSS	Test Suite Structure
TTCN-3	Testing and Test Control Notation edition 3
UT	Upper Tester

4 CCL Test Configuration

This clause describes the Test Configurations used to test the DMR Call Control Layer (CCL) and the DMR Data Link Layer (DLL) at the MS side and at the BS side. The Test Configurations are based on the Coordinated Test Method as describes in ISO/IEC 9646-1 [3].

Figure 1 shows the DMR protocol stack used to define the Test Configurations.

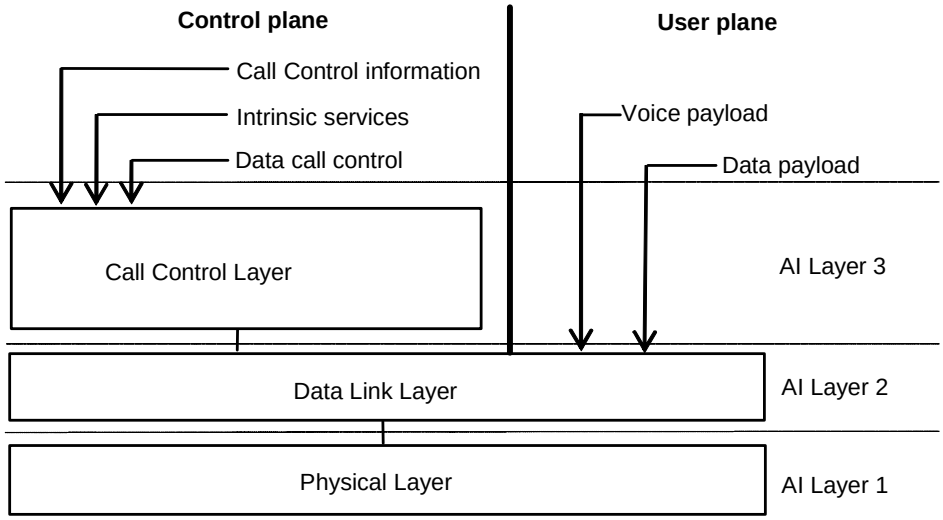


Figure 1: DMR protocol stack

4.1 CCL BS/MS Test Configuration

Figure 2 describes the CCL BS/MS Test Configuration for testing the CCL of a real product implementing the DMR base standard. More information for this architecture is provided below.

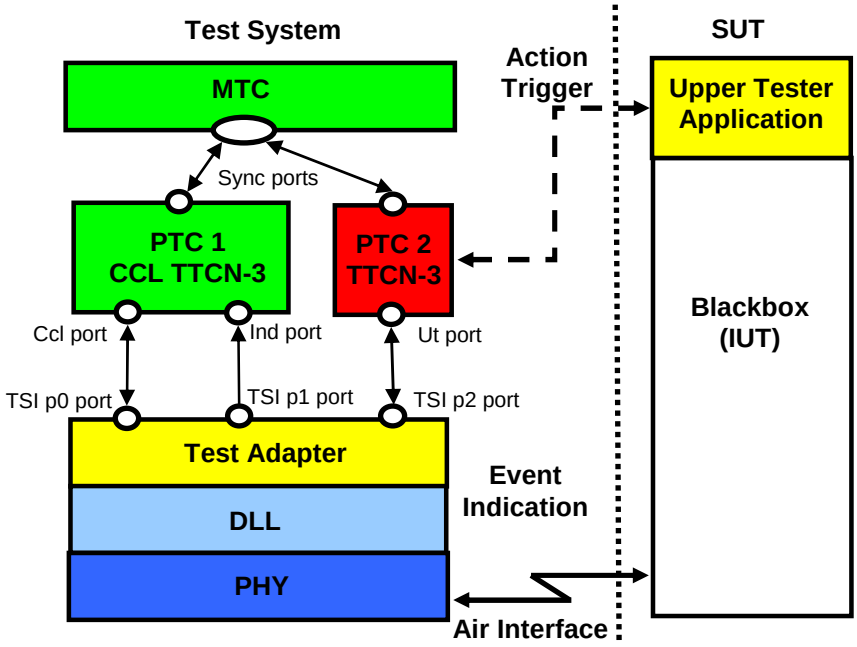


Figure 2: CCL MS/BS Test Configuration

The CCL MS/BS Test Configuration provides 3 test components:

- MTC:
  - Creating, synchronizing and terminating PTCs and setting the final test case verdict.
- PTC 1 - CclSimu:
  - CCL TTCN-3 uses Ccl port to send and receive CCL PDUs. Preliminary verdicts are set on the receive statements (MTC sets final verdict). The CCL PDUs that the Test Adapter shall support are listed in table 1.
  - CCL TTCN-3 uses Ind port to receive internal indications from DLL. Preliminary verdicts are set on the receive statements (MTC sets final verdict). The Indication message TaIndMsg that the Test Adapter shall support is listed in table 2.
  - PTC 1 controls via external functions the configuration of the Test System. Table 3 shows the list of Configuration Messages that the Test Adapter shall process.
  - For testing the BS (=IUT) by making two calls, another PTC of type CclSimu shall be added.
- PTC 2 - UpperTester:
  - TTCN-3 uses Ut port to control the Upper Tester Application.
  - The Upper Tester Application allows to observe IUT events. Preliminary verdicts are set on the receive statements of Indication Messages. The Indication message IutIndMsg that the Test Adapter shall support is listed in table 2.
  - The Upper Tester Application allows to configure the IUT. The Configuration messages that the Test Adapter shall support are listed in table 3.
  - The Upper Tester Application allows to trigger IUT actions such as initiating a PTT request. The IUT actions are observed on the Ccl port of PTC 1. The IUT Action messages that the Test Adapter shall support are listed in table 4.
- In the case where no Upper Tester is needed, the PTC becomes the MTC.
- MTC, PTC 1 and its Test Adapter with DLL and PHY form the Lower Tester.
- MTC, PTC 2 and its Test Adapter with Upper Tester Application form the Upper Tester.

4.2 CCL Test Adapter Requirements

- The Test Adapter implementation is outside the scope of the present document and is not part of the ATS development.
- Table 1 shows the CCL PDUs to be processed by the Test Adapter.

Table 1: CCL PDUs to be processed by the Test Adapter

CCL PDU	Port	Reference
BsDwnAct	Ccl port	clause 7.1 of TS 102 361-2 [2]
GrpVChUsr	Ccl port	clause 7.1 of TS 102 361-2 [2]
NackRsp	Ccl port	clause 7.1 of TS 102 361-2 [2]
UuAnsRsp	Ccl port	clause 7.1 of TS 102 361-2 [2]
UuVChUsr	Ccl port	clause 7.1 of TS 102 361-2 [2]
UuVReq	Ccl port	clause 7.1 of TS 102 361-2 [2]

- Table 2 shows the Indication Messages to be processed by the Test Adapter.
  - TA Indications refer to the slot on which the Ccl port is sending.

EXAMPLE 1: The TaIndMsg "eSlotIdle" refers to the slot on which Ccl port sent the preceding message.

- The Upper Tester Application reports the IUT events to the Test Adapter. Then the Test Adapter shall send the relevant IutIndMsg to PTC 2 where they are observed on the Ut port.

Table 2: Indication Messages to be processed by the Test Adapter

Indication mesage	Port	Reference
IutIndMsg	Ind port	DMR_Templates.ttcn
TaIndMsg	Ut port	DMR_Templates.ttcn

- Table 3 shows the Configuration Messages to be processed by the Test Adapter. The Configuration Messages describe the wanted configuration (for example parameters such as polite/impolite).

  - PTC 1 uses external functions (for example fx_taBsInit) to configure the Test System. The external functions are parameterized with Configuration Messages, and return FncRetCode.

  - PTC 2 sends Configuration Messages to the Test Adapter (and Upper Tester Application). (Upper Tester Application and) Test Adapter shall send FncRetCode to Ut port of PTC 2.

Table 3: Configuration Messages to be processed by the Test Adapter

Configuration message	Port	Reference
BsCfgParams	Ut port/ext fct	DMRTypes.asn
MsCfgParams	Ut port/ext fct	DMRTypes.asn
FncRetCode	Ut port/ext fct	DMRTypes.asn

- Table 4 shows the Action Messages to be processed by the Test Adapter.

  - PTC 1 uses external functions (for example fx_taMsAction) to trigger the Test System. The external functions are parameterized with Action Messages, and return FncRetCode.

  - PTC 2 sends an Action Message to the IUT. PTC 1 observes the IUT action.

Table 4: Action Messages to be processed by the Test Adapter

Action message	Port	Reference
BsActParams	Ut port	DMRTypes.asn
MsActParams	Ut port	DMRTypes.asn

- Table 5 shows the external functions to be processed by the Test Adapter.

EXAMPLE 2: The external function fx_taMsAction shall implement the sending of a voice burst with all related CCL PDUs.

Table 5: External functions to be processed by the Test Adapter

External function	Reference
Configuration functions	
fx_taBsInit	DMR_ExtFunctions.ttcn
fx_taMsInit	DMR_ExtFunctions.ttcn
Action functions	
fx_taMsAction	DMR_ExtFunctions.ttcn

5 DLL Test Configurations

The Testing Concept for DLL procedures is described in the text below.

TTCN-3 implements all Test Purposes defined in part 2 of this test specification. The Test Purposes cover DLL procedures like:

- CACH signalling;
- Channel Access Procedures;
- Channel Timing;
- DLL PDU management;
- Embedded Signalling;
- Voice signalling and voice transport.

The Test Adapter communicates with TTCN-3 and shall ensure sending and receiving of TDMA frames. Therefore the Test Adapter shall implement or provide access to DLL procedures like:

- Bit Ordering;
- Framing;
- Interleaving, De-Interleaving;
- Synchronization.

This concept of spliting the DLL procedures into a TTCN-3 part and a test adapter part is reflected in the naming of the test componentsand applies strictly only to the test system (and not to the IUT):

- TTCN-3 part is called "Upper DLL TTCN-3";
- Test Adaper part responsible for send/receive is called "Lower DLL".

5.1 Sending of voice bursts

5.2 DLL BS Test Configuration

Figure 3 describes the DLL BS Test Configuration for testing the DLL of a real product implementing the DMR base standard.

More information for this architecture is provided below.

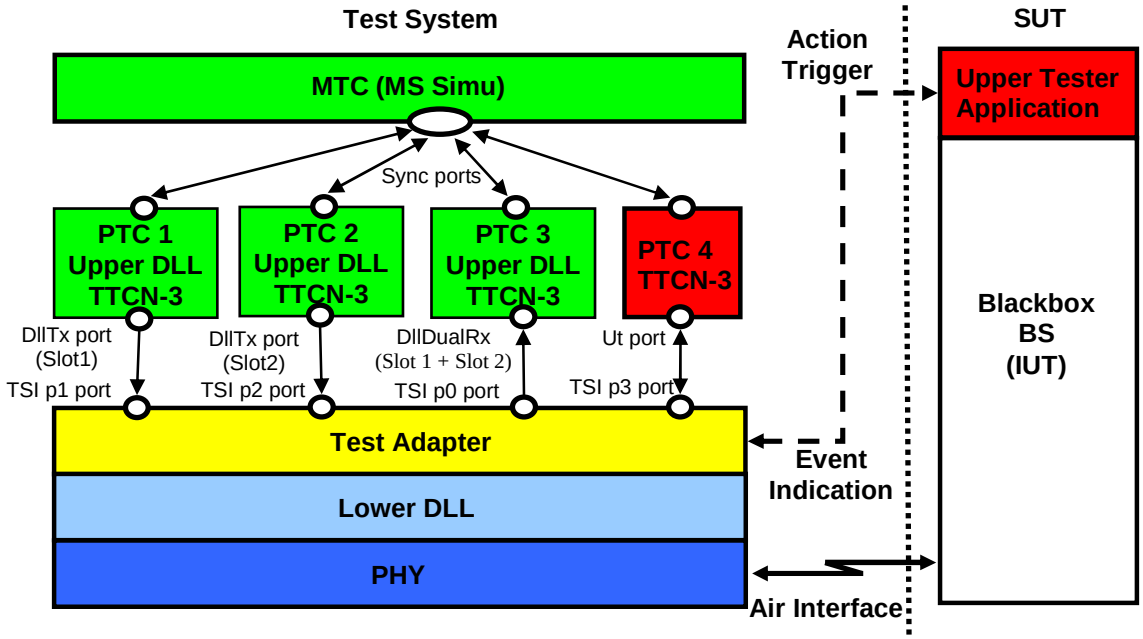


Figure 3: DLL BS Test Configuration

The DLL BS Test Configuration provides 4 test components:

- MTC:
  - Creating, synchronizing and terminating PTCs and setting the final test case verdict.
- PTC 1 - DllSlotTx:
  - Upper DLL TTCN-3 uses dllTx port to send a MsBurst. The MsBurst from PTC 1 shall be send in slot 1. The Test Adapter shall support the MsBursts, see table 6.
  - PTC 1 controls via external functions the configuration of the Test System. Table 8 shows the list of Configuration Messages that the Test Adapter shall process.
- PTC 2 - DllSlotTx:
  - The MsBursts from PTC 2 shall be send in slot 2. Otherwise same rules as for PTC 1 apply.
- PTC 3 - Dll2SlotRx:
  - Upper DLL TTCN-3 uses dllDualRx port to receive BsBursts. Preliminary verdicts are set on the receive statements (MTC sets final verdict). The Test Adapter shall support the BsBurst, see table 6.
- PTC 4 - UpperTester:
  - TTCN-3 uses Ut port to control the Upper Tester Application.
  - The Upper Tester Application allows to observe IUT events. Preliminary verdicts are set on the receive statements of Indication Messages. The Indication message IutIndMsg that the Test Adapter shall support is listed in table 7.
  - The Upper Tester Application allows to configure the IUT. The Configuration messages that the Test Adapter shall support are listed in table 8.
  - The Upper Tester Application allows to trigger IUT actions such as initiating a PTT request. The IUT actions are observed on the dllDualRx port of PTC 3. The IUT Action messages that the Test Adapter shall support are listed in table 9.
- MTC, PTC 1, PTC 2, PTC 3 and its Test Adapter with Lower DLL and PHY form the Lower Tester.

- MTC, PTC 4 and its Test Adapter with Upper Tester Application form the Upper Tester.

5.3 DLL MS Repeater Mode Test Configuration

Figure 4 describes the DLL MS Repeater Mode Test Configuration for testing the DLL of a real product implementing the DMR base standard.

More information for this architecture is provided below.

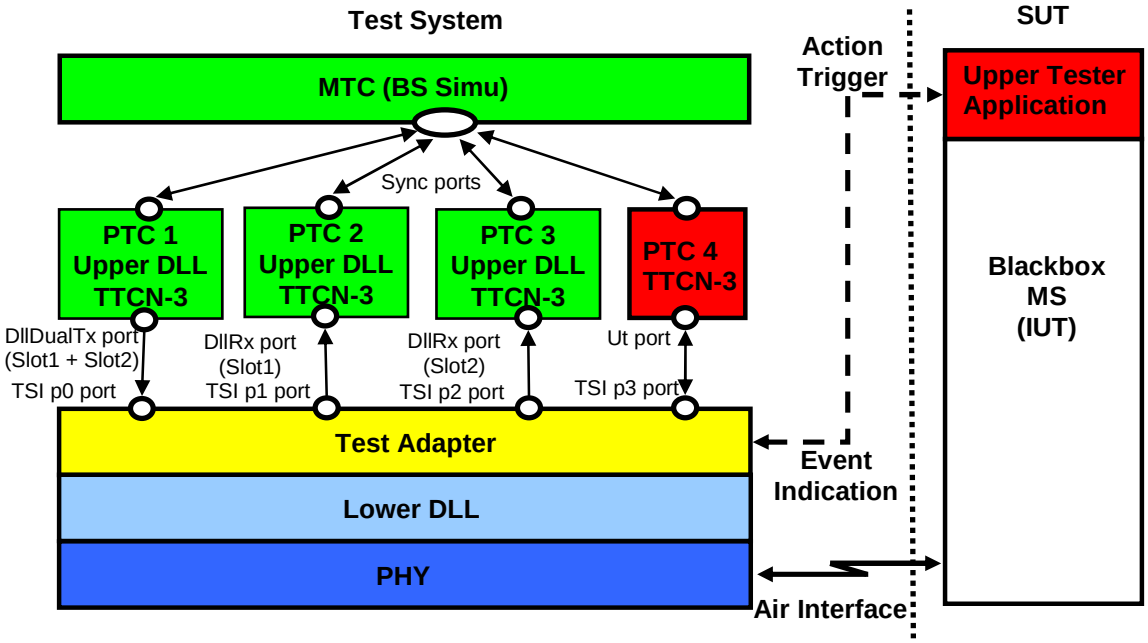


Figure 4: DLL MS Test Configuration

The DLL MS Repeater Mode Test Configuration provides 4 test components:

- MTC:
  - Creating, synchronizing and terminating PTCs and setting the final test case verdict.
- PTC 1 - Dll2SlotTx:
  - Upper DLL TTCN-3 uses dllDualTx port to send a BsBurst. The Test Adapter shall support the BsBursts, see table 6.
  - PTC 1 controls via external functions the configuration of the Test System. Table 8 shows the list of Configuration Messages that the Test Adapter shall process.
- PTC 2 - DllSlotRx:
  - Upper DLL TTCN-3 uses dllRx port to receive MsBurst. The MsBurst shall relate to slot 1. Preliminary verdicts are set on the receive statements (MTC sets final verdict). The Test Adapter shall support the MsBurst, see table 6.
- PTC 3 - DllSlotRx:
  - The MsBurst shall relate to slot 2. Otherwise same rules as PTC 2 apply.
- PTC 4 - UpperTester:
  - Same rules as in DLL BS Test Configuration apply.
- MTC, PTC 1, PTC 2, PTC 3 and its Test Adapter with Lower DLL and PHY form the Lower Tester.

- MTC, PTC 4 and its Test Adapter with Upper Tester Application form the Upper Tester.

5.4 DLL MS Direct Mode Test Configuration

Figure 5 describes the DLL MS Direct Mode Test Configuration for testing the DLL of a real product implementing the DMR base standard.

More information for this architecture is provided below.

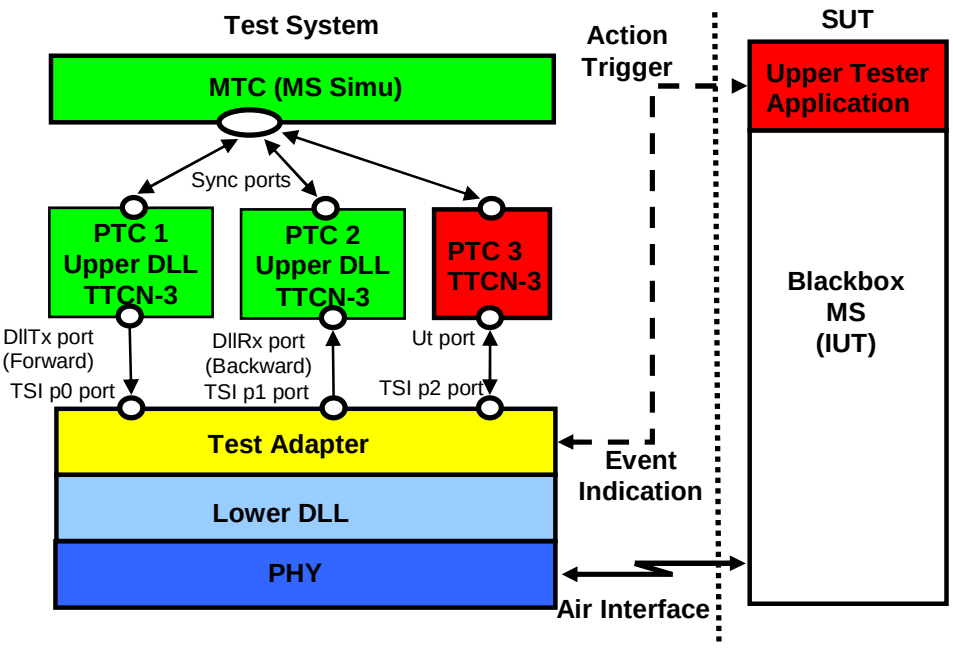


Figure 5: DLL MS Direct Mode Test Configuration

The DLL MS Direct Mode Test Configuration provides 3 test components:

- MTC:
  - Creating, synchronizing and terminating PTCs and setting the final test case verdict.
- PTC 1 - DllSlotTx:
  - Upper DLL TTCN-3 uses dllTx port to send a MsBurst. The MsBurst from PTC 1 shall be send in slot 1 or slot 2. The Test Adapter shall support the MsBursts, see table 6.
  - PTC 1 controls via external functions the configuration of the Test System. Table 8 shows the list of Configuration Messages that the Test Adapter shall process.
- PTC 2 - DllSlotRx:
  - Upper DLL TTCN-3 uses dllRx port to receive MsBurst. The MsBurst shall be received in slot 1 or slot 2. Preliminary verdicts are set on the receive statements (MTC sets final verdict). The Test Adapter shall support the MsBurst, see table 6.
- PTC 3 - UpperTester:
  - Same rules as in DLL BS Test Configuration apply.
- MTC, PTC 1, PTC 2 and its Test Adapter with Lower DLL and PHY form the Lower Tester.
- MTC, PTC 3 and its Test Adapter with Upper Tester Application form the Upper Tester.

5.5 DLL Test Adapter Requirements

- The Test Adapter implementation is outside the scope of the present document and is not part of the ATS development.
- Two TTCN-3 messages are defined:
  - The BsBurst contains all DLL PDUs to be sent/received in CACH 1 - Timeslot 1 and CACH 2 Timeslot 2.

EXAMPLE 1: {Cach, Sync, SlotType, Idle}+{Cach, Sync, SlotType, Idle} PDUs.

  - The MsBurst all DLL PDUs to be sent/received in either Timeslot 1 or Timeslot 2. Therefore in each case a mapping to a specific slot will be given.

EXAMPLE 2: {Sync, SlotType, Idle} PDUs.

- When receiving a TTCN-3 message from TRI, the Test Adapter shall de-assemble the TTCN-3 message into the DMR burst format and send it to the air interface.
- When receiving a DMR burst format from the air interface, the Test Adapter shall assemble the DMR burst format into a TTCN-3 message and send it to TRI.
- Table 6 shows the TTCN-3 messages to be processed by the Test Adapter. Further information can be found in clause A.3.

Table 6: TTCN-3 messages to be processed by the Test Adapter

TTCN-3 msg	Port	Reference
BsBurst	dllDualRx port dllDualTx port	DMRTypes.asn
MsBurst	dllRx port dllTx port	DMRTypes.asn

- Table 7 shows the Indication Messages to be processed by the Test Adapter.
  - The Upper Tester Application reports the IUT events to the Test Adapter. Then the Test Adapter shall send the relevant IutIndMsg to PTC 4 where they are observed on the Ut port.

Table 7: Indication Messages to be processed by the Test Adapter

Indication mesage	Port	Reference
IutIndMsg	Ind port	DMR_Templates.ttcn

- Table 8 shows the Configuration Messages to be processed by the Test Adapter. The Configuration Messages describe the wanted configuration (for example parameters such as polite/impolite):
  - non-UT test components use external functions (for example fx_taBsInit) to configure the Test System. The external functions are parameterized with Configuration Messages, and return FncRetCode.
  - UT test component sends Configuration Messages to the Test Adapter (and Upper Tester Application). (Upper Tester Application and) Test Adapter shall send FncRetCode to Ut port of PTC 4.

Table 8: Configuration Messages to be processed by the Test Adapter

Configuration message	Port	Reference
BsCfgParams	Ut port/ext fct	DMRTypes.asn
MsCfgParams	Ut port/ext fct	DMRTypes.asn
FncRetCode	Ut port/ext fct	DMRTypes.asn

- Table 9 shows the Action Messages to be processed by the Test Adapter:

  - non-UT test components use external functions (for example fx_taMsAction) to trigger the Test System. The external functions are parameterized with Action Messages, and return FncRetCode.

  - UT test component sends an Action Message to the IUT. PTC 3 observes the IUT action.

Table 9: Action Messages to be processed by the Test Adapter

Action message	Port	Reference
BsActParams	Ut port	DMRTypes.asn
MsActParams	Ut port	DMRTypes.asn

- Table 10 shows the external functions to be processed by the Test Adapter.

Table 10: External functions to be processed by the Test Adapter

External function	Reference
Configuration Functions	
fx_taBsInit	DMR_ExtFunctions.ttcn
fx_taMsInit	DMR_ExtFunctions.ttcn
Action Functions	
fx_taMsAction	DMR_ExtFunctions.ttcn
Calculation Functions	
fx_taCalTactParity	DMR_ExtFunctions.ttcn
fx_taCalSlotTypeParity	DMR_ExtFunctions.ttcn
fx_taCalEmbParity	DMR_ExtFunctions.ttcn
fx_taCalFlc24BitsCrc	DMR_ExtFunctions.ttcn
fx_taCalFlc5BitsCrc	DMR_ExtFunctions.ttcn
fx_taCalCsbkCrc	DMR_ExtFunctions.ttcn

6 ATS conventions

The ATS conventions are intended to give a better understanding of the ATS but they also describe the conventions made for the development of the ATS. These conventions shall be considered during any later maintenance or further development of the ATS.

The ATS conventions contain two clauses, the naming conventions and the implementation conventions. The naming conventions describe the structure of the naming of all ATS elements. The implementation conventions describe the functional structure of the ATS.

To define the ATS, the guidelines of the document ETS 300 406 [6] are considered.

6.1 Naming conventions

The naming convention is based on the following underlying principles:

- in most cases, identifiers should be prefixed with a short alphabetic string (specified in table 11) indicating the type of TTCN-3 element it represents;

- suffixes should not be used except in those specific cases identified in table 11;

- prefixes and suffixes should be separated from the body of the identifier with an underscore ("_"):

EXAMPLE 1: `c_sixteen, t_waitMax_g;`

- only module names, data type names and module parameters should begin with an upper-case letter. All other names (i.e. the part of the identifier following the prefix) should begin with a lower-case letter;

- the start of second and subsequent words in an identifier should be indicated by capitalizing the first character. Underscores should not be used for this purpose:

EXAMPLE 2: `f_authenticateUser();`

Table 11 specifies the naming guidelines for each element of the TTCN-3 language indicating the recommended prefix, suffixes (if any) and capitalization.

Table 11: TTCN-3 naming convention

Language element	Naming convention	Prefix	Example	Notes
Module	Use upper-case initial letter	none	DMR_TypesAndValues	
Item group within a module	Use lower-case initial letter	none	messageGroup	
Data type	Use upper-case initial letter	none	SetupContents	
Message template	Use lower-case initial letter	m_	m_setupInit m_setupBasic	Note 1
Message template with wildcard or matching expression	Use lower-case initial letters	mw_	mw_anyUserReply	Note 2
Port instance	Use lower-case initial letter	none	signallingPort	
Test component ref	Use lower-case initial letter	none	userTerminal	
Constant	Use lower-case initial letter	c_	c_maxRetransmission	
External constant	Use lower-case initial letter	cx_	cx_macId	
Function	Use lower-case initial letter	f_	f_authentication()	
External function	Use lower-case initial letter	fx_	fx_calculateLength()	
Altstep (incl. Default)	Use lower-case initial letter	a_	a_receiveSetup()	
Test case	Use all upper case letters	TC_	TC_BS_DLL_TACT_BV_001	
Variable (local)	Use lower-case initial letter	v_	v_macId	
Variable (defined within a component)	Use lower-case initial letters	vc_	vc_systemName	
Timer (local)	Use lower-case initial letter	t_	t_wait	
Timer (defined within a component)	Use lower-case initial letters	tc_	tc_authMin	
Module parameter	Use all upper case letters	none	PX_MAC_ID	
Parameterization	Use lower-case initial letter	p_	p_macId	
Enumerated Value	Use lower-case initial letter	e_	e_syncOk	

NOTE 1: This prefix must be used for all template definitions which do *not* assign or refer to templates with wildcards or matching expressions, e.g. templates specifying a constant value, parameterized templates without matching expressions, etc.

NOTE 2: This prefix must be used in identifiers for templates which either assign a wildcard or matching expression (e.g. ?, *, value list, ifpresent, pattern, etc) or reference another template which assigns a wildcard or matching expression.

6.2 Implementation conventions

6.2.1 Templates

- Templates should be identified with names rather than numbers.
- Templates should not modify other modified templates. Base templates which are modified must be identified in their naming.
- Templates should be specified separately for use in sending and receiving operations. The Prefixes as described above must be used in identifiers for templates which either assign a wildcard or matching expression (e.g. ?, *, value list, ifpresent, pattern, etc) or reference another template which assigns a wildcard or matching expression.
- Template definitions should avoid using matching attributes such as "*" or "?" for complete structured values, e.g. record or set of values.
- PIXIT parameter values should be passed as parameters into templates.

6.2.2 Functions

The DMR ATS differentiates between synchronization functions, verdict handling functions and other functions. Each type of function is implemented in a separate module, although there may be multiple modules for each function type. The following general rules apply:

- Functions should use the *runs on* statement wherever this is possible.
- Each function should provide a return value. It is recommended to use the return value enumeration defined in the DMRTypes.asn file.

EXAMPLE: DMRAts.FncRetCode.

- If a PIXIT parameter is used as condition of an *if* statement, then its body should contain only a function call.
- The *stop* statement should be used with care in functions (controlled test component shutdown should be always insured).

6.2.3 Synchronization functions

The following guidelines apply to functions handling the synchronization of multiple, parallel test components:

- Synchronization should be invoked by the MTC at least after the preamble and before the postamble. The MTC may also invoke synchronization at other appropriate times.
- A PTC should synchronize after setting a verdict. This is to ensure that the verdict is always set prior to a PTC shutdown.
- Synchronization should use "named" synchronization as implemented in CommonLib_SyncLib.ttcn:
  - Named synchronization uses a different synchronization message for each synchronization in order to avoid confusion where multiple synchronizations are required.
- Synchronization of test termination should use the stop message which is the character string "`STOP`".
- To terminate test execution a PTC should send the stop message to the MTC and wait for the corresponding STOP-notification from the MTC.
- If an MTC receives the stop message then it should send stop messages to all PTCs.
- To terminate test execution an MTC should send the stop message to all PTCs and wait for them to cease execution.
- If a PTC receives the stop message then it should execute the appropriate postamble. This could be implemented as default behaviour. As this notification may occur at any point of the PTC execution, the postamble should take its current state into account.

6.3 Test Case (TC) identifier

The identifier of the test case is built in the same way as for the test purpose described in part 2 of this TS, with the exception that "TP" is replaced by "TC".

6.3.1 CCL TP naming conventions

The identifier of the TP is built according to table 12.

Table 12: TC naming convention for CCL

Identifier	TC/<st>/<sl>/<sg>/<fm >/<x>-<nnn>		
	<st> = side type	BS	Base Station
		MS	Mobile Station
	<sl> = stack layer	CCL	Call Control Layer
		DLL	Data Link Layer
	<sg> = service group	BA	BS Downlink Activation
		VCR	Voice Call Repeating
		CHT	Voice Call Hangtime
		CR	CSBK Repeating
		BDA	BS Downlink Deactivation
		FNS	Feature Not Supported
		IC	Individual Call
		GC	Group Call
		UC	Unaddressed Voice Call
		AC	All Call Voice
		BC	Broadcast Call Voice
		OVCM	Open Voice Channel Mode
	<fm> = functional module	MS_INI	MS Initiating
		MS_TER	MS Terminating
	x = type of testing	BV	Valid Behaviour Tests
		TI	Timer and Constraints Tests
	<nnn> = sequential number	(000...)	

EXAMPLE: TC_BS_CCL_BA_MS_INI_BV_000 is the first test case for the valid behaviour testing of the MS_INItiated BS activation procedure of the Call Control layer at the BS side.

6.3.2 DLL TP naming conventions

The identifier of the TC is built according to table 13.

Table 13: TC naming convention for DLL

Identifier:	TC/<st>/<sl>/<sg>/<fm>/<x>-<nnn>		
	<st> = side type	BS	Base Station
		MS	Mobile Station
	<sl> = stack layer	CCL	Call Control Layer
		DLL	Data Link Layer
	<sg> = service group	CA	Channel Access
		SYNC	Synchronization
		ST	Slot Type
		EMB	Embedded Signalling
		TACT	TDMA Access Channel Type
		TT	Traffic Timing
	<fm> = functional module	DM	Direct Mode(Peer to Peer Mode)
		RM	Repeater Mode
	x = type of testing	BV	Valid Behaviour Tests
		TI	Timer Tests
	<nnn> = sequential number	(000...)	

EXAMPLE: TP_MS_DLL_CA_DM_BV_001 is the second test case for the valid behaviour testing of the channel accessing procedure in direct mode of the Data Link layer at the MS side.

Annex A (informative): Guideline on Upper Tester, Inhouse Testing and Send/Receive of DLL TDMA bursts

A.1 Specifying an Upper Tester

In order to completely automate conformance and interoperability testing, the upper interface of the IUT needs to be accessible to TTCN-3 test cases. The specification of this upper interface is not standardized by DMR and so there are no primitives defined for requesting the DMR stack to send a specific burst or to check if one has been received. Consequently, implementations of this interface are vendor specific and may even vary between different IUTs.

In conformance testing methodology the tight integration problem can be resolved by implementing an Upper Tester Application (UTA) in the SUT, i.e., outside of the test system. The purpose of the UTA is to play the role of a (dummy) DMR application which interacts with the DMR stack. It is, however, controlled by the test system with the Upper Tester Component via a message channel. Therefore, another task of the UT is to convert the messages sent by TTCN-3 into concrete DMR interface calls and vice versa. This allows a fairly generic design and encoding of a protocol between the UT and TTCN-3.

Table A.1 shows a test purpose which requires an Upper Tester.

Table A.1: Test Purpose which requires an Upper Tester

TP/MS/VT/BV-xxx	**Reference:** TS 102 361-2 [2], clauses 6.3.2.1 and 6.2.3.2.3 **Initial condition:** The IUT is in synchronization with the TS and the channel is idle. Check, that when the IUT initiates a PTT_Request and is granted the right to transmit, the IUT initially sends a Voice_LC_Header message.

A.1.1 The UT in the DMR test system

In the test system the UT is assigned in each test case an own UT port. During the execution of a test case commands are sent to the UTA in the SUT via the UT port. The commands:

- indicate the reception of an DMR burst:
- configure the SUT.

Further on the commands could:

- indicate the start and end of a test case:
- reset the UT in case of test case errors.

The UT commands that are used are listed in table A.2. The UT commands are non-standarized, but it could be considered to use AT commands instead.

Table A.2: UT commands

UT command	Port	Reference
BsActParams	Ut port	DMRTypes.asn
MsActParams	Ut port	DMRTypes.asn
BsCfgParams	Ut port	DMRTypes.asn
MsCfgParams	Ut port	DMRTypes.asn
IutIndMsg	Ut port	DMRTypes.asn
FncRetCode	Ut port	DMRTypes.asn

A.2 Using the ATS for Inhouse Testing

The delivered CCL and DLL test systems can be extended for Inhouse Testing. One example is the early prototype testing where:

- IUT is a software application (ETSI validates its TTCN-3 against Mirror TTCN-3 software application).
- The air interface is replaced with a TCP/IP interface.
- TTCN-3 is not changed, because it is independent from the Test Adapter.
- External functions and Upper Tester Application are modified to fit the new SUT.
- TCP/IP connection between Upper Tester and Upper Tester Application (a serial port interface could be used as well).

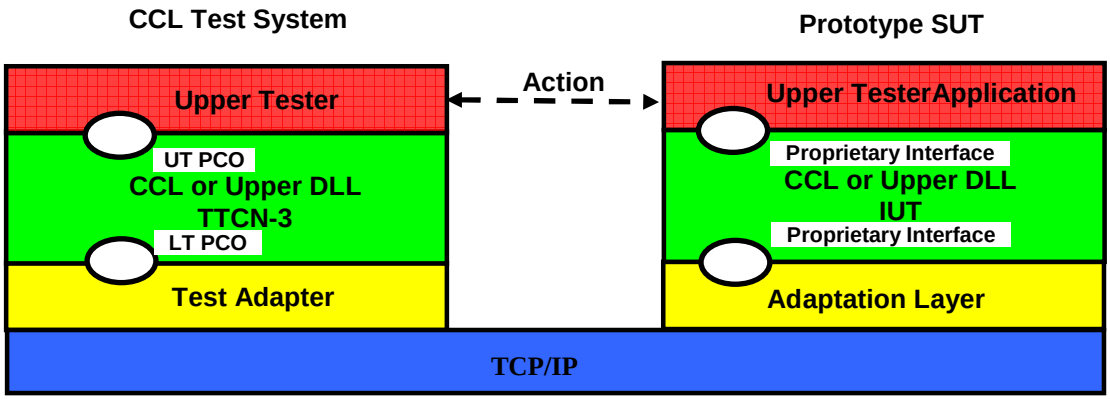


Figure A.1: Software Implementation Testing with TCP/IP

A.3 Sending and Receiving DLL TDMA bursts

Send and receive process is handeled separately on different test components. This split allows to run in parallel the:

- sending of TTCN-3 messages faster then the DMR system clock;
- receiving, enqueuing of TTCN-3 messages and discharging the queue uncoupled to the DMR system clock.

When sending e.g. Voice Superframes, a TTCN-3 message is sent for each burst of the superframe ("A" through "F"). TTCN-3 messages shall be sent fast enough so that TA can de-assemble the TTCN-3 message into the DMR burst format and send it to the air interface.

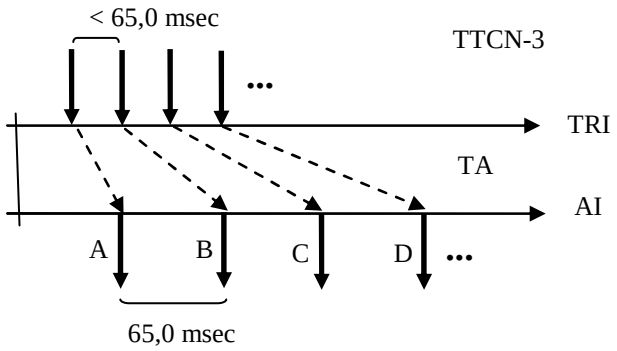


Figure A.2: Sending DLL TDMA bursts

Figure A.3 shows the process of receiving DLL TDMA bursts.

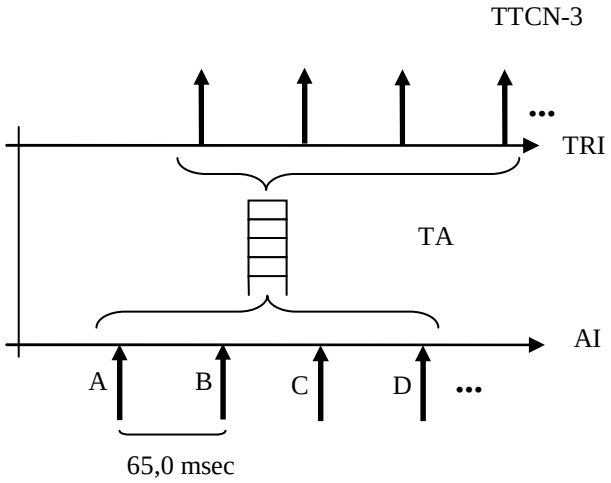


Figure A.3: Receiving DLL TDMA bursts

Annex B (normative): Abstract Test Suite (ATS)

B.1 The ATS in TTCN-3 core (text) format

This ATS has been produced using the Testing and Test Control Notation (TTCN) according to ES 201 873-1 [7].

The TTCN-3 core (text) representation corresponding to this ATS is contained in an ASCII file(s) (DMR_TTCN3_v001r1.zip contained in archive ts_10236203v010101p0.zip) which accompanies the present document.

NOTE: Where an ETSI Abstract Test Suite (in TTCN-3) is published in both core and tabular format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

B.2 The ATS in TTCN-3 tabular format

This ATS has been produced using the Testing and Test Control Notation (TTCN) according to ES 201 873-2 [8].

The TTCN-3 Tabular representation of this ATS is contained in an Adobe Portable Document Format™ file (DMR_T3DOC_v001r1.zip contained in archive ts_10236203v010101p0.zip) which accompanies the present document.

NOTE: Where an ETSI Abstract Test Suite (in TTCN-3) is published in both core and tabular format these two forms shall be considered equivalent. In the event that there appears to be syntactical or semantic differences between the two then the problem shall be resolved and the erroneous format (whichever it is) shall be corrected.

Annex C (normative): Partial PIXIT proforma for DMR

The PIXIT Proforma is based on ISO/IEC 9646-6. Any needed additional information can be found in the present document.

C.1 Identification summary

Table C.1

PIXIT Number:	
Test Laboratory Name:	
Date of Issue:	
Issued to:	

C.2 ATS summary

Table C.2

Protocol Specification:	TS 102 361-1 and TS 102 361-2
Protocol to be tested:	
ATS Specification:	TS 102 362-3
Test Configuration:	TS 102 362-3 clauses 4 and 5

C.3 Test laboratory

Table C.3

Test Laboratory Identification:	
Test Laboratory Manager:	
Means of Testing:	
SAP Address:	

C.4 Client identification

Table C.4

Client Identification:	
Client Test manager:	
Test Facilities required:	

C.5 SUT

Table C.5

Name:	
Version:	
SCS Number:	
Machine configuration:	
Operating System Identification:	
IUT Identification:	
PICS Reference for IUT:	
Limitations of the SUT:	
Environmental Conditions:	

C.6 Protocol layer information

C.6.1 Protocol identification

Table C.6

Name:	TS 102 361-1 and TS 102 361-2
Version:	
PICS References:	

C.6.2 IUT information

C.6.2.1 Timers

Table C.7 : Timers

Name	Type	Comment
PXT_MAX_CASE_EXEC_PERIOD	float	Time of Max Case Execution
PXT_MAX_BS_REPEATING_DELAY	float	Time of Max Bs Repeating Delay
PXT_MAX_TIME_RECV_NEXT_FRM	float	Timer for receving next TDMA frame (it should be greater than 60E-3 sec)
PXT_MAX_TIME_CFG_ACT_RLY	float	Max Time of IUT sending back response of configuration/action's func/msg
PXT_GUARD_TIME	float	General Guard Timer
PXT_MS_HOLD_TRANSMISSION_TIME	float	Timer for MS holding transmisson
PXT_GUARD_TIME_CALL_HT	float	Guard Time when testing Call HT
PXT_GUARD_TIME_MS_INACTIV	float	Guard Time of Ms inactive
PXT_GUARD_TIME_TRANSMITTING	float	Guard Time of Ms transmitting

C.6.2.2 Common Configuration

Table C.8 : Common Configuration

Name	Type	Comment
PXT_VALIDATION_MODE	boolean	Debug flag
PXT_BS_ADDR	BsAddr	Value of BS Address
PXT_MS_SIMU_SRC_ADDR	SrcAddr	Value of MS Simu Source Address
PXT_GRP_ADDR	GrpAddr	Value of Group Address
PXT_WRONG_TARGET_ADDR	TargetAddr	Wrong Value of Target Address
PXT_TARGET_ADDR	TargetAddr	Value of Target Address
PXT_UNADDR_V_CALL_ADDR	TargetAddr	Value of Unaddress Idn Address from FFFFE0 to FFFFEF
PXT_ALL_UNIT_V_CALL_ADDR	TargetAddr	Value of All Unit Idn Address from FFFFF0 to FFFFFF
PXT_ADDI_INFO	AdditionalInfo	Value of Additonal Information used in NackRsp PDU
PXT_MY_SYSTEM_CC	Cc	Value of My System Color Code
PXT_OTHER_SYSTEM_CC	Cc	Value of Other System Color Code

Annex D (normative): PCTR proforma for DMR

The PCTR proforma is based on ISO/IEC 9646-6. Any needed additional information can be found in the present document.

D.1 Identification summary

D.1.1 Protocol conformance test report

Table D.1

PCTR Number:	
PCTR Date:	
Corresponding SCTR Number:	
Corresponding SCTR Date:	
Test Laboratory Identification:	
Test Laboratory Manager:	
Signature:	

D.1.2 IUT identification

Table D.2

Name:	
Version:	
Protocol specification:	
PICS:	
Previous PCTR if any:	

D.1.3 Testing environment

Table D.3

PIXIT Number:	
ATS Specification:	
Test Configuration:	
Means of Testing identification:	
Date of testing:	
Conformance Log reference(s):	
Retention Date for Log reference(s):	

D.1.4 Limits and reservation

Additional information relevant to the technical contents or further use of the test report, or the rights and obligations of the test laboratory and the client, may be given here. Such information may include restriction on the publication of the report.

.....

.....

.....

.....

.....

D.1.5 Comments

Additional comments may be given by either the client or the test laboratory on any of the contents of the PCTR, for example, to note disagreement between the two parties.

.....

.....

.....

.....

.....

D.2 IUT Conformance status

This IUT has or has not been shown by conformance assessment to be non-conforming to the specified protocol specification.

Strike the appropriate words in this sentence. If the PICS for this IUT is consistent with the static conformance requirements (as specified in clause D.3 in the present document) and there are no "FAIL" verdicts to be recorded (in clause D.6 in the present document) strike the words "has or", otherwise strike the words "or has not".

D.3 Static conformance summary

The PICS for this IUT is or is not consistent with the static conformance requirements in the specified protocol.

Strike the appropriate words in this sentence.

D.4 Dynamic conformance summary

The test campaign did or did not reveal errors in the IUT.

Strike the appropriate words in this sentence. If there are no "FAIL" verdicts to be recorded (in clause D.6 of the present document) strike the words "did or" otherwise strike the words "or did not".

Summary of the results of groups of test:

.....

.....

.....

.....

.....

.....

.....

.....

.....

D.5 Static conformance review report

If clause D.3 indicates non-conformance, this subclause itemizes the mismatches between the PICS and the static conformance requirements of the specified protocol specification.

.....

.....

.....

.....

.....

.....

.....

.....

.....

D.6 Test campaign report

Table D.4

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause D.7)
MS DLL				
TC_MS_DLL_SYNC_BV_000	yes/no	yes/no		
TC_MS_DLL_SYNC_BV_001	yes/no	yes/no		
TC_MS_DLL_SYNC_BV_002	yes/no	yes/no		
TC_MS_DLL_ST_BV_000	yes/no	yes/no		
TC_MS_DLL_ST_BV_001	yes/no	yes/no		
TC_MS_DLL_ST_BV_002	yes/no	yes/no		
TC_MS_DLL_EMB_DM_BV_000	yes/no	yes/no		
TC_MS_DLL_EMB_RM_BV_000	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_000	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_001	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_002	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_003	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_004	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_005	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_006	yes/no	yes/no		
TC_MS_DLL_CA_DM_BV_007	yes/no	yes/no		
TC_MS_DLL_CA_DM_TI_000	yes/no	yes/no		
TC_MS_DLL_CA_DM_TI_001	yes/no	yes/no		
TC_MS_DLL_CA_DM_TI_002	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_000	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_001	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_002	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_003	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_004	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_005	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_006	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_007	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_008	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_009	yes/no	yes/no		
TC_MS_DLL_CA_RM_BV_010	yes/no	yes/no		
TC_MS_DLL_CA_RM_TI_000	yes/no	yes/no		
TC_MS_DLL_CA_RM_TI_001	yes/no	yes/no		
TC_MS_DLL_CA_RM_TI_002	yes/no	yes/no		
TC_MS_DLL_CA_CRC_BV_000	yes/no	yes/no		
TC_MS_DLL_CA_CRC_BV_001	yes/no	yes/no		
TC_MS_DLL_CA_CRC_BV_002	yes/no	yes/no		
TC_MS_DLL_CA_CRC_BV_003	yes/no	yes/no		
TC_MS_DLL_CA_CRC_BV_004	yes/no	yes/no		
BS DLL				
TC_BS_DLL_TACT_BV_000	yes/no	yes/no		
TC_BS_DLL_TACT_BV_001	yes/no	yes/no		
TC_BS_DLL_TACT_BV_002	yes/no	yes/no		
TC_BS_DLL_TACT_BV_003	yes/no	yes/no		
TC_BS_DLL_SYNC_BV_000	yes/no	yes/no		
TC_BS_DLL_SYNC_BV_001	yes/no	yes/no		
TC_BS_DLL_ST_BV_000	yes/no	yes/no		
TC_BS_DLL_TT_BV_000	yes/no	yes/no		
TC_BS_DLL_TT_BV_001	yes/no	yes/no		
TC_BS_DLL_CRC_BV_000	yes/no	yes/no		

ATS Reference	Selected?	Run?	Verdict	Observations (Reference to any observations made in clause D.7)
MS CCL				
TC_MS_CCL_BA_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_BA_MS_INI_TI_000	yes/no	yes/no		
TC_MS_CCL_BA_MS_INI_TI_001	yes/no	yes/no		
TC_MS_CCL_BA_MS_INI_TI_002	yes/no	yes/no		
TC_MS_CCL_FNS_MS_TER_BV_000	yes/no	yes/no		
TC_MS_CCL_GC_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_GC_MS_INI_BV_001	yes/no	yes/no		
TC_MS_CCL_GC_MS_INI_BV_002	yes/no	yes/no		
TC_MS_CCL_GC_MS_TER_BV_000	yes/no	yes/no		
TC_MS_CCL_GC_MS_TER_BV_001	yes/no	yes/no		
TC_MS_CCL_GC_MS_TER_BV_002	yes/no	yes/no		
TC_MS_CCL_GC_MS_TER_BV_003	yes/no	yes/no		
TC_MS_CCL_GC_MS_TER_BV_004	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_001	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_002	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_003	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_004	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_BV_005	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_TI_000	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_TI_001	yes/no	yes/no		
TC_MS_CCL_IC_MS_INI_TI_002	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_000	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_001	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_002	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_003	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_004	yes/no	yes/no		
TC_MS_CCL_IC_MS_TER_BV_005	yes/no	yes/no		
TC_MS_CCL_UC_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_AC_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_BC_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_OVCM_MS_INI_BV_000	yes/no	yes/no		
TC_MS_CCL_OVCM_MS_INI_BV_001	yes/no	yes/no		
TC_MS_CCL_TI_MS_INI_BV_000	yes/no	yes/no		
BS CCL				
TC_BS_CCL_BA_MS_INI_BV_000	yes/no	yes/no		
TC_BS_CCL_VCR_MS_INI_BV_000	yes/no	yes/no		
TC_BS_CCL_VCR_MS_INI_BV_001	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_000	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_001	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_002	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_003	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_004	yes/no	yes/no		
TC_BS_CCL_CHT_MS_INI_TI_005	yes/no	yes/no		
TC_BS_CCL_CR_MS_INI_BV_000	yes/no	yes/no		
TC_BS_CCL_CR_MS_INI_BV_001	yes/no	yes/no		
TC_BS_CCL_CR_MS_INI_BV_002	yes/no	yes/no		
TC_BS_CCL_BDA_MS_INI_TI_000	yes/no	yes/no		
TC_BS_CCL_BDA_MS_INI_TI_001	yes/no	yes/no		
TC_BS_CCL_AC_MS_INI_BV_000	yes/no	yes/no		
TC_BS_CCL_BC_MS_INI_BV_000	yes/no	yes/no		

D.7 Observations

Additional information relevant to the technical content of the PCTR is given here.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

History

Document history		
V1.1.1	June 2005	Publication