MACHINE READABLE TRAVEL DOCUMENTS

TECHNICAL REPORT

RF PROTOCOL AND APPLICATION TEST STANDARD FOR E-PASSPORT - PART 4

E-PASSPORT READER TESTS FOR AIR INTERFACE, INITIALISATION, ANTICOLLISION AND TRANSPORT PROTOCOL

Version: **1.01**
Date – Feb 20, 2007

Published by authority of the Secretary General

INTERNATIONAL CIVIL AVIATION ORGANIZATION

File	: RF protocol Part 4 V1.01.doc
Author	: ISO/JTC1/SC17/WG3/TF4 for ICAO-NTWG

Release Control

Release	Date	Description
0.1	2006-02-01	First draft based on the German WG3 TF4 contribution “e-Passport Conformity Testing” version 0.5.2 presented at TF4 meeting in Ottawa Jan 30- Feb 01, 2006
0.2	2006-05-19	Updated version based on “e-Passport Conformity Testing” PCD, layer 1-4, version 0.6, and new ICAO TR layout. Changes in chapters 3.3.1 and 3.3.2 concerning reference SCIC.
0.3	2006-09-27	Updated version based on disposition of comments from July 14th, 2006 (resolutions of Graz meeting).
0.9	2006-11-23	Updated version based on disposition of comments from October 30th, 2006 (resolutions of Bled meeting)
1.0	2006-12-12	Final version including editorial comments from the WG8 editor meeting in Graz.
1.01	2007-02-20	Annex A and corresponding chapter 2.1 removed. Editorial error in table 4 removed.

Release Note:

Release 2007-02-20 is the official version of the technical report by SC17 WG3 TF4 endorsed by ICAO-NTWG at the Portugal meeting. It replaces version V1.0.

Table of contents

1 Introduction

1.1 Scope and purpose

An essential element of the new ICAO compliant e-Passport is the addition of a Secure Contactless Integrated Circuit (SCIC) that securely holds biometric data of the passport bearer within the ICAO defined Logical Data Structure (LDS).

Successful integration of the SCIC into the e-Passport depends upon an active international cooperation between many companies and organizations.

The e-Passport and the e-Passport reader (proximity coupling device or PCD) have been specified and designed to operate correctly across a wide variety of infrastructures worldwide. The risk profile for the e-Passport and the reader indicates a high impact if that design includes a widespread error or fault. Therefore, it is essential that all companies and organizations involved make all reasonable efforts to minimize the probability that this error or fault remains undetected before that design is approved and e-Passports and e-Passport readers are issued.

This document defines a test plan for the contactless part of the PCD. These tests are divided into tests of the electrical parameters, according to ISO/IEC14443-2 and tests of the initialisation & anticollision and the frame protocol according to ISO/IEC14443-3 and -4.

In order for the PCD to operate correctly, many functional layers of technology must work together. The purpose of this document is to define in depth the tests to be performed to minimize the probability that an error or fault remain undetected before the design is approved.

1.2 Assumptions

1.2.1 Maintenance of this TR regarding ISO standards

This technical report is based on the currently available versions of ISO standards and amendments as they are referenced in chapter 1.6. Based on the further development of these referenced ISO standards and their amendments, this report will be revised by SC17/WG3/TF4.

1.3 Terminology

The key words "MUST", "SHALL", "REQUIRED", "SHOULD", "RECOMMENDED", and "MAY" in this document are to be interpreted as described in [R3].

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1.4 Glossary

ATTRIB Activation command used with modulation type B.

TR0 TR0 defines the guard time between the end of a PCD transmission and the start of the SCIC subcarrier generation.

TR1 TR1 defined the synchronization time between the start of the SCIC subcarrier generation and the start of the SCIC subcarrier modulation.

TR2 TR2 defines the synchronization time between the start of the SCIC's EOF and the start of the PCD's next SOF.

Sample A sample is one piece of the total number of e-Passports required and presented for testing according to this specification.

DUT A device under test is a sample that has been placed in the test apparatus.

Room temperature Room temperature (RT) is defined as any convenient temperature within the range of 23 °C ± 3 °C (73 °F ± 5 °F).

Modulation index The modulation index m is calculated as follows:

$$m = \frac{\text{maxlevel} - \text{minlevel}}{\text{maxlevel} + \text{minlevel}}$$

1.5 Abbreviations

Abbreviation	
AC	Anti-collision command
APDU	Application protocol data unit
ATQA	Answer to REQA
ATQB	Answer to REQB
ATS	Answer to select
CID	Card identifier
DUT	Device under test
EGT	Extra guard time
EOF	End of frame
etu	Elementary time unit
fc	Carrier frequency (13.56 MHz)

Abbreviation	
FDT	Frame delay time
fs	Sub-carrier frequency (847.5 kHz)
LMA	Load modulation amplitude
m	Modulation index
PCD	Proximity coupling device
PICC	Proximity integrated circuit card
PPS	Protocol and parameter selection
RATS	Request for ATS
REQA	Request A command
REQB	Request B command
RT	Room temperature
SCIC	Secure contactless integrated circuit
SOF	Start of frame
TM-PDU	Test management protocol data unit
t_r, t_f	Rise time, fall time

1.6 Reference documentation

The following documentation serves as a reference for this technical report:

[R1] *Technical Report: Development of a Logical Data Structure – LDS for optional capacity expansion technologies, version 1.7*
[R2] *Technical Report: PKI for Machine Readable Travel Documents offering ICC Read-Only access, version 1.1*
[R3] *RFC 2119, S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997*
[R4] *CD ISO/IEC 14443-1:2005, Proximity Cards: Physical Characteristics*
[R5] *ISO/IEC 14443-2:2001, Proximity Cards: Radio Frequency Power and Signal Interface*
[R6] *ISO/IEC 14443-3:2001, Proximity Cards: Initialization and Anticollision*
[R7] *ISO/IEC 14443-4:2001, Proximity Cards: Transmission protocol*
[R8] *ISO/IEC 14443-2:2001/AM1:2005, Proximity Cards: Radio Frequency Power and Signal Interface (Amendment 2: Bit Rates of fc/64, fc/32 and fc/16)*
[R9] *ISO/IEC 14443-3:2001/AM1:2005, Proximity Cards: Initialization and Anticollision (Amendment 1: Bit Rates of fc/64, fc/32 and fc/16)*
[R10] *ISO/IEC 10373-6:2001, Test Methods for Proximity Cards*
[R11] *ISO/IEC 10373-6:2001/AM1:2006, Test Methods for Proximity Cards (Amendment 1: Protocol Test Methods for Proximity Cards)*
[R12] *ISO/IEC 10373-6:2001/AM2:2006, Test Methods for Proximity Cards (Amendment 2: Improved RF Test Methods)*
[R13] *ISO/IEC 10373-6:2001/AM3:2006, Test Methods for Proximity Cards (Amendment 32: Protocol test methods for proximity coupling devices)*
[R14] *ISO/IEC 10373-6:2001/AM4:2006, Test Methods for Proximity Cards (Amendment 4: Additional Test Methods for PCD RF Interface and PICC Alternating Field Exposure)*
[R15] *ISO/IEC 10373-6:2001/AM5:2006, Test Methods for Proximity Cards (Amendment 5: Bit Rates of fc/64, fc/32 and fc/16)*
[R16] *ICAO Doc 9303 Part 1 Volume 2, 6th edition, 2005.*
[R17] *Defect Report and Technical Corrigendum 1 for - International Standard ISO/IEC 14443-3:2001/AM1: Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: initialization and anticollision – Amendment 1: Bit rates for fc/64, fc/32 and fc/16*

2 General test requirements

Following sub-clauses specify the different test setups, the nominal values used for the tests and a recommendation of the report. Tests for bit rates of fc/128 and fc/32 are mandatory and SHALL be applied. Other bit rates indicated in the PPS/ATTRIB command SHALL also be tested.

This test specification refers to the ISO/IEC standard 10373-6 and its corresponding amendments.

The PCD SHOULD support asymmetric communication speeds from/to the e-Passport if offered by the e-Passport to minimize transaction time.

The tolerance for the resonance frequency of reference e-Passport is ± 2 %.

All given temperature values MAY have a tolerance value of ± 0.5 °C.

All other value MAY have a tolerance value as specified in the base standards.

2.1 Test environment

2.1.1 Test setup

The PCD assembly (test apparatus) that is defined in [R10] is the basis for the physical and electrical tests. This test apparatus is used to calibrate the Reference e-Passports that are defined in [R10] (chapter 6.3).

In addition to [R10], the samples SHALL provide the features as described in chapter 2.3. The manufacturer provides a description how to switch the sample into the test mode and how to operate the sample for the test cases described in this document.

2.2 Implementation conformance statement

In order to set up the tests properly, an applicant SHALL provide the information specified in Table 1 below.

Table 1: Test precondition table "Information on the product"

Information for test setup	Applicant declaration
Reader class	
Bit rates supported as claimed by the PPS/ATTRIB • 106 kbit/sec • 212 kbit/sec • 424 kbit/sec • 848 kbit/sec	
Access control supported • Plaintext • Basic Access Control • Extended Access Control	
Authentication supported • Passive Authentication • Active Authentication	
Operating temperature range	

2.3 PCD test features

The test apparatus SHALL be capable of sending contiguous activation commands. For type A, these commands are REQA, AC, SELECT, RATS, and PPS. For type B, these are REQB, the optional Slot-MARKER and ATTRIB. If there is no response from the e-Passport, the communication type SHOULD be changed and the activation procedure SHALL start from beginning. If this is not possible, the dedicated commands SHALL be available to expose all commands to the upper tester. Therefore, it is possible to start each command from host side if necessary.

Additionally the ISO/IEC 144443-4 command set SHALL be available to exchange data. It SHALL also be possible to receive chained data, e.g. BlockExchange.

Errors SHALL be handled in the reader and not in the upper tester or host. If possible, the final operating system SHALL be tested.

For synchronization purposes the PCD MAY provide a test pin output. The test apparatus MAY also be synchronized by probing the backscattered signal using an ISO pickup coil. This pickup coil MUST NOT influence the field significantly.

The applicant MAY provide the reader test interface specified in Annex C.

2.4 Nominal values

Unless otherwise specified, the following environment parameters and nominal values SHALL be used:

Table 2: Environment parameters

Parameter	Value	Applies to
Environment temperature	23 °C ± 3 °C (73 °F ± 5 °F)	Type A and B
Relative humidity	25 % to 75 %	Type A and B

Tests have to be done as the same temperature range as the e-Passport tests (-10 °C … 50 °C). The customer is free to specify a limited range (for example for indoor systems) in the implementation conformance statement.

2.5 Definition of measurement points

All layer 2 tests SHALL be performed over a certain set of points within the defined volume.

Volume definition:

Annex A specifies volume dimensions, so called “reader design types”. If due to the construction and/or normal use of the reader other dimension sizes are RECOMMENDED by the manufacturer of the reader, the test institute SHALL check if these dimension sizes are appropriate and define the dimensions of the volume accordingly.

Volume location:

The PCD manufacturer SHALL define the position of the volume in the technical documentation of the PCD. The volume MUST be located with one surface exactly on the surface of the reader device.

Alternatively, the volume MAY be located within the reader. In this case, the volume size definition MUST be adopted accordingly.

Readers MUST be tested inside of their housing, exactly as they are used in border control applications.

Applying the reader design type concept, it is required to consider mechanical and optical constraints specific to a reader. The test MAY be adapted to match these constraints. The report SHALL state the specific operating conditions during a particular test.

Measurement points:

Annex A specifies measurement points.

Height Z = 0 mm: The measuring antenna SHALL be placed exactly at the bottom of the volume (at the surface of the scanner plate, if appropriate).

Height Z = x mm: The top surface of the test antenna SHALL be located in a distance of x mm of the bottom of the volume (in a distance of x mm from the surface of the scanner plate, if appropriate).

2.6 Definition of the reference e-Passport for load modulation reception test

The reference e-Passport introduced for the load modulation reception test is based on ISO 10373-6 Annex E [R10] with improved functionality.

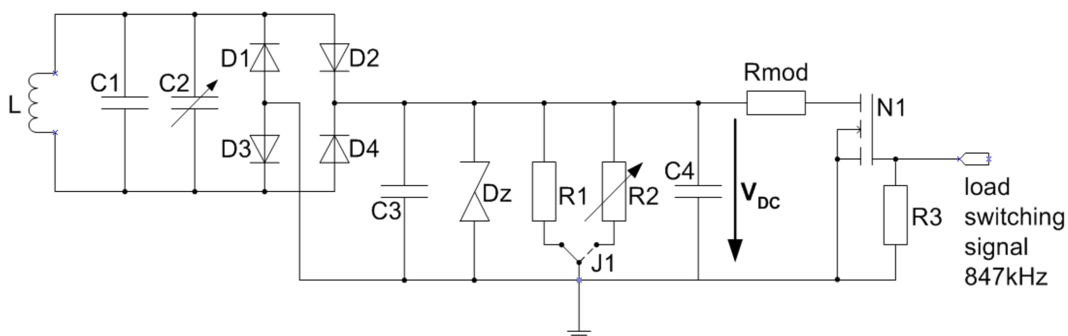

Figure 1: Reference e-Passport

The signal at pin 'load switching signal 847 kHz' shall have an amplitude value between 0 and 4 V, for stable switching of the recommended transistor. In order to reduce common-mode current, a balun MAY be used between the signal generator and the reference e-Passport.

Table 3: Reference e-Passport

Component	Value
L	See [R10], chapter 6.3.5
R1	4.3 kΩ
R2	Adjustable, see table below
R3	5 kΩ
R_{mod}	Adjustable, see table below
C1	10pF
C2	5 – 30 pF
C3	100 pF
C4	470 pF
D1, D2, D3, D4	Recommended: BAR43
Dz	Recommended: BZX84-C14 / T1, 300 mW, SMD 15 V
N1	N-MOS FET, 10 pF max. Output capacitance to ground. Recommended: BSS83

For X=30 and H = 2 A/m the load modulation amplitude ($X/H^{1.2}$) is 13.1 mV.

Adjustment of the R2 and R_{mod} values must be done in the Test PCD assembly [R12].

Adjustment of the Reference e-Passport load modulation amplitude level for V_{DC} = 6 V at field strength values from 2 A/m up to 6 A/m result in the following nominal modulation resistor values.

Table 4: Values of R_{mod} and R2

Load Modulation Amplitude	Freq	H	R2	R_{mod}
13.1 mV	15MHz	2 A/m	705 Ω	1.118 kΩ
13.1 mV	15MHz	2.5 A/m	525 Ω	1.047 kΩ
13.1 mV	15MHz	3 A/m	437 Ω	0.966 kΩ
13.1 mV	15MHz	3.5 A/m	370 Ω	0.898 kΩ
13.1 mV	15MHz	4 A/m	320 Ω	0.845 kΩ
13.1 mV	15MHz	4.5 A/m	285 Ω	0.800 kΩ
13.1 mV	15MHz	5 A/m	255 Ω	0.765 kΩ
13.1 mV	15MHz	5.5 A/m	230 Ω	0.745 kΩ
13.1 mV	15MHz	6 A/m	210 Ω	0.732 kΩ
13.1 mV	18MHz	2 A/m	5.1 kΩ	4.670 kΩ
13.1 mV	18MHz	2.5 A/m	703 Ω	1.376 kΩ
13.1 mV	18MHz	3 A/m	510 Ω	1.045 kΩ
13.1 mV	18MHz	3.5 A/m	380 Ω	0.941 kΩ
13.1 mV	18MHz	4 A/m	320 Ω	0.880 kΩ
13.1 mV	18MHz	4.5 A/m	285 Ω	0.825 kΩ
13.1 mV	18MHz	5 A/m	255 Ω	0.785 kΩ
13.1 mV	18MHz	5.5 A/m	230 Ω	0.760 kΩ
13.1 mV	18MHz	6 A/m	210 Ω	0.740 kΩ

Notes:

- These resistor values are nominal values and SHALL be used as a guideline when adjusting R2 and R_{mod}.
- V_{DC} SHALL be measured using reference e-Passport when no 847 KHz switching signal is applied to the probe.

2.7 Report

The test report SHALL include the number of passed tests versus the total number of tests. A description of each test, the information if the test was pass or fail, the number of different samples and the date of the tests SHOULD be included.

3 Layer 2 tests

3.1 Operating field strength test (Types A and B)

Test description: The purpose of this test is to check if the PCD meets the energy performance requirements according to [R5] (chapter 6.2) and [R10] (chapter 8.1). To include a margin of 0.5 A/m to the ISO limits, the field strength under loaded conditions SHALL be between 2.0 A/m (1.5 + 0.5) and 7.0 A/m (7.5 - 0.5) at all measurement positions defined in chapter 2.5. As a measurement device, the Reference e-Passport for field and power measurements defined in [R10], Annex D SHALL be used.

Additionally, the value of V_{DC} SHOULD be recorded for all positions. These values are required for the load amplitude modulation test in chapter 3.2.

For H_{min} perform the following steps:

1. Adjust the resonance frequency of the Reference e-Passport to **15 MHz** as described in [R12] (8.1.2).
2. Put the Reference e-Passport into the ISO test PCD assembly.
3. Adjust the resistor R2 to get a V_{DC} of 6 V at field strength of 2 A/m.
4. At any measurement position defined in chapter 2.5 the V_{DC} SHALL be greater or equal to 6 V.

For H_{max} perform the following steps:

1. Adjust the resonance frequency of the Reference e-Passport to **18 MHz** as described in [R12] (8.1.2).
2. Put the Reference e-Passport into the ISO test PCD assembly.
3. Adjust the resistor R2 to get a V_{DC} of 6 V at field strength of 7 A/m.
4. At any measurement position defined in chapter 2.5 the V_{DC} SHALL be less or equal to 6 V.

Modifications are done according:

- Reference: ISO/IEC 14443-2 [R5] (chapter 6.2), Operating field strength H_{min} - H_{max}
- Reference: ISO/IEC 10373-6 [R10] (chapter 8.1), PCD field strength H_{min} - H_{max}

Conditions: Minimum number of samples: 1

Parameters:

Measurement position:	As defined in chapter 2.5
Bit rate:	fc/128
Temperature:	-10 °C, RT, 50 °C (see restriction in chapter 2.4 “Nominal Values”)
Reference e-Passport resonance frequency:	15 MHz, 18 MHz

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

Notes:

- This test includes field strength and power measurements, see [R10] (chapter 8.1 and 8.2).
- H_{min} and H_{max} values are defined to keep a margin to the field strength range for the e-Passport test.
- In addition, the tests recommended by ISO 10373-6 MAY be performed using 13.56 MHz and 19 MHz.

3.2 Load modulation reception test (Type A and B)

An ISO Reference e-Passport for load modulation tests does not give the possibility to test the load modulation amplitude with each PCD. In order to adapt this test, the load-switching signal SHALL be a response to a request command and SHALL be synchronous to a request command as defined in [R12].

Test description: The purpose of this test is to determine if the PCD is able to receive and demodulate signals with minimum load modulation amplitude. The PCD SHOULD provide a trigger signal (e.g. pulse at beginning or end of PCD command) to the load switching signal source (e.g. an arbitrary waveform generator) to send the response with required timings. The Reference e-Passport defined in chapter 2.6 SHALL be used.

For each measurement position and resonance frequency perform the following steps:

1. Adjust the resonance frequency of the Reference e-Passport to 15MHz as described in [R12], chapter 8.1.2.
2. Put the Reference e-Passport into the Test PCD assembly [R12].
3. Adjust the resistor R2 to get a V_{DC} of 6V at H = 2 A/m field strength according to AM4 [R14].
4. Adjust the resistor R_{mod} to get the required load modulation amplitude of 13.1 mV (for nominal resistor values see Table in chapter 2.6)
5. Put the Reference e-Passport to the measurement position defined in chapter 2.5.
6. Adjust the distance along z axis between the Reference e-Passport and PCD until V_{DC} reaches 6V.
7. Check if the PCD is able to receive and demodulate a valid response with required bit rate (see table below).
8. Repeat steps 2 to 7 where the field strength should be increased in 0.5 A/m steps up to H_{max}
9. Repeat steps 1 to 8 at Reference e-Passport resonance frequency of 18MHz.

Table 5: Communication sequences

Bit rate	PCD command	e-Passport response
fs/128	REQ	ATQ coding & framing according to higher bit rate spec.
fs/64	REQ	ATQ coding & framing according to higher bit rate spec.
fs/32	REQ	ATQ coding & framing according to higher bit rate spec.
fs/16	REQ	ATQ coding & framing according to higher bit rate spec.

Conditions:	Minimum number of samples:	1
Parameters:	Measurement position:	As defined in chapter 2.5
	Bit rate:	fc/128, fc/64, fc/32, fc/16
	Temperature:	-10 °C, RT, 50 °C (see restriction in chapter 2.4 “Nominal Values”)
	Reference e-Passport resonance frequency:	15 MHz, 18 MHz
	Load modulation amplitude:	$LMA_{min} = 30/H^{1.2}$ mVpeak
Report:	The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.	

Note: For bit rates higher than fc/128, the ‘Transmit Pattern and Receive 14443’ command as specified in Annex C.1.3.2 MAY be executed by the PCD.

3.3 Modulation index and waveform test

Digital amplitude demodulation SHALL be used for calculating the envelope of the modulated carrier amplitude, e.g. Hilbert transformation. For an example program of the Hilbert transformation, see Annex D.

3.3.1 Type A

This test shall verify if the modulated field of the PCD is complying with the described waveform explained in [R5] (chapter 8.1) and [R8] (8.1). The parameters are rise and fall times, modulation index, and overshoots

Modification done according:

- Reference: ISO/IEC 10373-6/Amd4 [R14](Annex I), Reference PICC for modulation index and waveform test
- Reference: ISO/IEC 14443-2 [R5] (chapter 8.1, 9.1)
- Reference ISO/IEC 14443-2/Amd 1 [R8] (8.1, 9.1)

Test conditions for fc/128 are shown in [R5] (chapter 8.1.2), the test conditions for fc/64, fc/32, fc/16 in [R8] (8.1.2).

Test description: The purpose of this test is to determine the compliance of the PCD regarding waveform shapes. The test SHALL show if the shapes of the modulated field is within the defined limits. The test SHALL be performed according to [R10] (chapter 8.3). The Reference e-Passport for modulation index and waveform test [R14] (Annex I) SHALL be used in addition to the calibration coil. A command with the required bit rate shall be sent by the PCD.

Adjust the resonance frequency of the Reference e-Passport to 16.5 MHz as described in [R12] (8.1.2).

For each measurement position perform the following steps:

1. Put the Reference e-Passport into position defined in chapter 2.1.1.
2. Adjust the resistor R2 to get a V_{DC} of 6 V at current position
3. Check if the waveform shapes are within the specified limits for all bit rates at current position.

Conditions:	Minimum number of samples:	1
Parameters:	Measurement position:	As defined in chapter 2.5
	Bit rate:	fc/128, fc/64, fc/32, fc/16
	Temperature:	-10 °C, RT, 50 °C (see restriction in chapter 2.4 “Nominal Values”)
	Reference e-Passport resonance frequency:	16.5 MHz
Report:	The test report shall include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.	

Notes:

- For better interoperability, a Reference e-Passport resonance frequency of 15 MHz SHOULD be used.
- For each bit rate, the corresponding table SHOULD be taken in account.
- For bit rates higher than fc/128, the ISO 14443-2 test command as specified in Annex C.1.3.1 MAY be executed with the PCD.

3.3.2 Type B

This test SHALL check if PCD meets the requirements concerning waveform shapes, i.e. rise and fall times, modulation index, and overshoots.

Test description:

The purpose of this test is to determine the compliance of the PCD regarding waveform shapes (see table 36 below). The test SHALL show if the shapes of the modulated field is within the defined limits. The test SHALL be performed according to [R10] (chapter 8.3). The Reference e-Passport for modulation index and waveform test [R14] (Annex I) SHALL be used in addition to the calibration coil. A command with the required bit rate shall be sent by the PCD.

1. Adjust the resonance frequency of the Reference PICC to 16.5 MHz as described in [R12] (8.1.2).

For each measurement position perform the following steps:

2. Put the reference e-Passport into position defined in chapter 2.1.1.
3. Adjust the resistor R2 to get a V_{DC} of 6 V at current position
4. Check at that position if the waveform shapes are within the specified limits for all bit rates, as defined in table below.

The ISO 14443-2 test command as specified in Annex C.1.3.1 MAY be executed with the PCD.

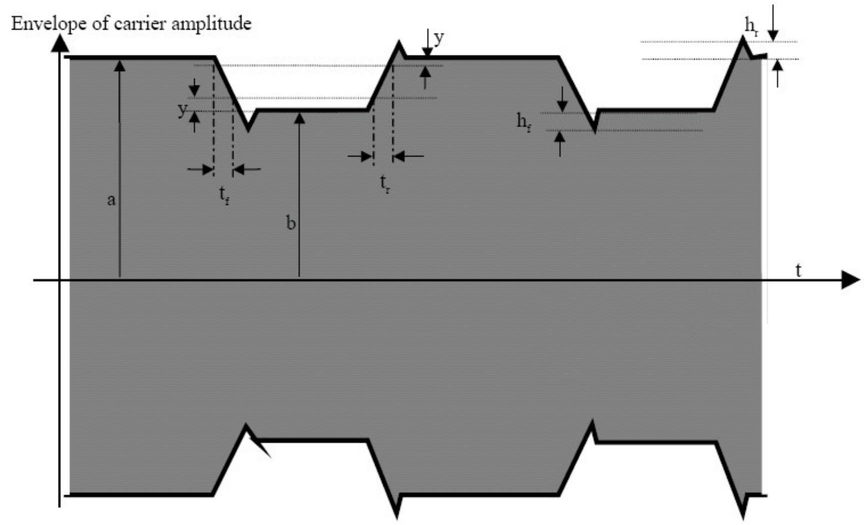

Figure 2: Envelope of type B carrier amplitude

Table 6: Waveform shape requirements

Parameter	min	max
m = (a-b)/(a+b)	10 %	14 %
t_r, t_f	0 µs	0.8 µs
h_r, h_f	0	0.1(a-b)

Conditions: Minimum number of samples: 1

Parameters:

Measurement position: As defined in chapter 2.5

Bit rate: fc/128, fc/64, fc/32, fc/16

Reference e-Passport resonance frequency: 16.5 MHz

Temperature: -10 °C, RT, 50 °C (see restriction in chapter 2.4 “Nominal Values”)

Report: The test report shall include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

Notes:

- For better interoperability a Reference e-Passport resonance frequency of 15 MHz SHOULD be used.
- For bit rates higher than fc/128, the ISO 14443-2 test command as specified in Annex C.1.3.1 MAY be executed with the PCD.
- The higher bit rate requirements of 0.8 µs are sufficient for all bit rates defined here because the higher bit rate of 424 kbps is mandatory for the e-Passport.

4 Layer 3 timing and framing tests

All tests SHALL be performed with one specific field strength between 2 A/m and 7 A/m within the operating volume of the PCD if not further specified.

All tests SHALL be performed at RT if not further specified.

4.1 Frame delay time (Type A only)

4.1.1 Frame delay time SCIC to PCD

This test SHALL check if the PCD can handle a FDT according to [R6]. For this test, the same setup SHALL be used as for the load modulation reception test.

Modification done according:

- Reference: ISO/IEC 14443-3 [R6] (chapter 6.1.3), Frame delay time SCIC to PCD

Test description:	This test SHALL check if a PCD command after a SCIC response is not sent before a minimum frame delay time of 1172/fc after the SCIC has sent ATQ. After ATQ the PCD SHALL send an AC frame.	
Conditions:	Minimum number of samples:	1
Parameters:	Bit rate:	fc/128
Report:	The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.	

Note: If possible, this test SHOULD be done for all commands, even during the protocol test.

4.1.2 Frame delay time PCD to SCIC

This test SHALL check if the PCD can handle a FDT according to [R6]. For this test, the same setup SHALL be used as for the load modulation reception test.

Modification done according:

- Reference: ISO/IEC 14443-3 [R6] (chapter 6.1.2), Frame delay time PCD to SCIC

Test description: This test SHALL check if the PCD is able to receive a SCIC response within the FDT limits.

Table 7: FDT limits

Last Bit	Min FDT	Max FDT
0	1172/fc	1172/fc + 0.4 µs
1	1236/fc	1236/fc + 0.4 µs

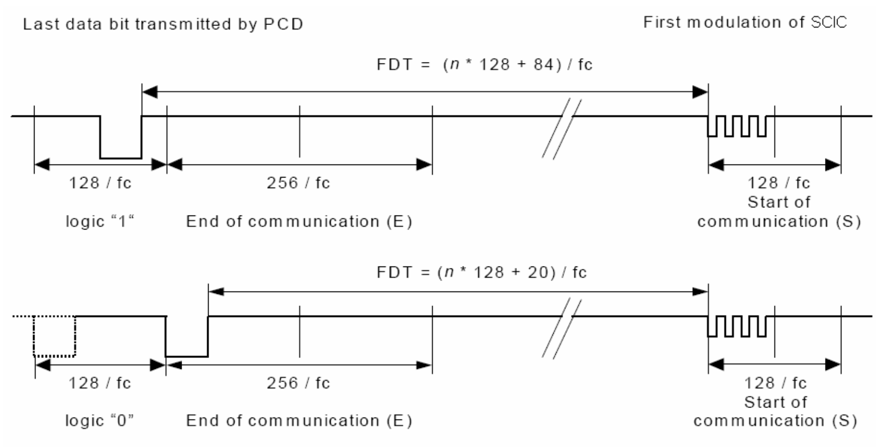

Figure 3: Frame Delay Time

Conditions: Minimum number of samples: 1

Parameters: Bit rate: fc/128

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

Note:

- The test SHALL be done for a REQA/WUPA command and SHOULD be carried out with other commands, too (see [R9]).
- In order to improve interoperability, the following values SHOULD be used (ISO limits ± 1 carrier period):

Table 8: Parameters for improved interoperability

Last Bit	Min FDT	Max FDT
0	1172/fc – **1/fc**	1172/fc + 0.4 µs + **1/fc**
1	1236/fc – **1/fc**	1236/fc + 0.4 µs + **1/fc**

4.2 Request guard time (Type A only)

This test SHALL check if the PCD can handle multiple REQA commands according to [R6]. For this test, the same setup SHALL be used as for the load modulation amplitude test.

Modification done according:

- Reference: ISO/IEC 14443-3 [R6] (chapter 6.1.4), Request Guard Time

Test description:	The purpose of this test is to determine the Request Guard Time of two consecutive REQA/WUPA commands.	
Conditions:	Minimum number of samples:	1
Parameters:	Bit rate:	fc/128
Report:	The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.	

Note: This test is only relevant for PCD's, which send consecutive REQA/WUPA.

4.3 Bit boundaries (Type B only)

Test description: The purpose of this test is to check whether the PCD meets the bit boundary requirements according to [R9] (7.1.1).

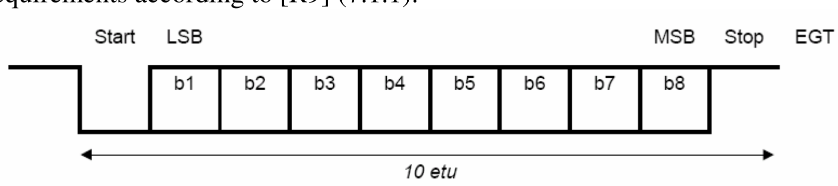

Figure 4: Bit boundary

Conditions:	Minimum number of samples:	1
Parameters:	Bit rate:	fc/128, fc/64, fc/32, fc/16
Report:	The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.	

4.4 Start-of-Frame & End-of-Frame-Timing (SOF & EOF) (Type B only)

Test description: The purpose of this test is to check whether the PCD meets SOF & EOF requirements according to [R6] (chapter 7.1.4 and 7.1.5).

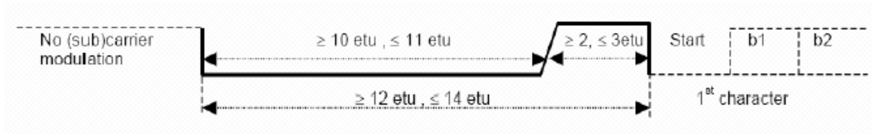

Figure 5: SOF

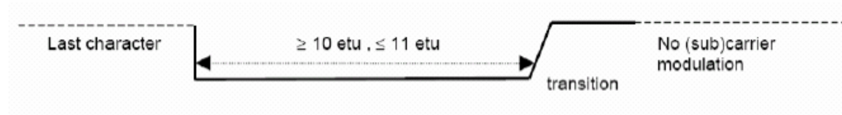

Figure 6: EOF

Conditions: Minimum number of samples: 1

Parameters: Bit rate: fc/128, fc/64, fc/32, fc/16

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

4.5 Extra guard time (EGT) (Type B only)

Test description: The purpose of this test is to check whether the PCD meets the EGT requirements according to [R9] (7.1.2).

Table 9: EGT limits

	min	**max**
EGT	1 etu	6 etu

Conditions: Minimum number of samples: 1

Parameters: Bit rate: fc/128, fc/64, fc/32, fc/16

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

4.6 Timing before SCIC Start-of-Frame (TR0 & TR1) (Type B only)

Test description: The purpose of this test is to check whether the PCD meets the TR0 and TR1 requirements according to [R6] (chapter 7.1.6).

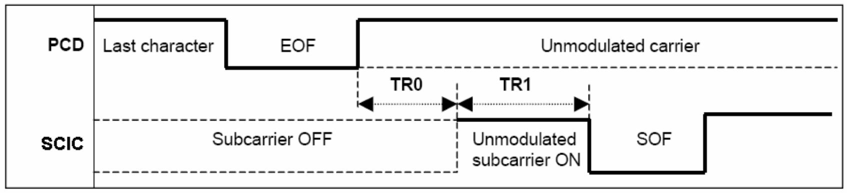

Figure 7: TR0 & TR1

Table 10: TR0 & TR1 limits

	min	max
TR0	64/fs	256/fs
TR1	80/fs	200/fs

Conditions: Minimum number of samples: 1

Parameters: Bit rate: fc/128, fc/64, fc/32, fc/16

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

Note: In order to improve interoperability, the following values SHOULD be used:

Table 11: TR0 & TR1 limits for improved interoperability

	min	max
TR0	60/fs	260/fs
TR1	76/fs	204/fs

4.7 Timing before PCD Start-of-Frame (TR2) (Type B only)

Test description: The purpose of this test is to check whether the PCD meets the minimum TR2 requirements. TR2 limits are tested according to the defect report [R17].

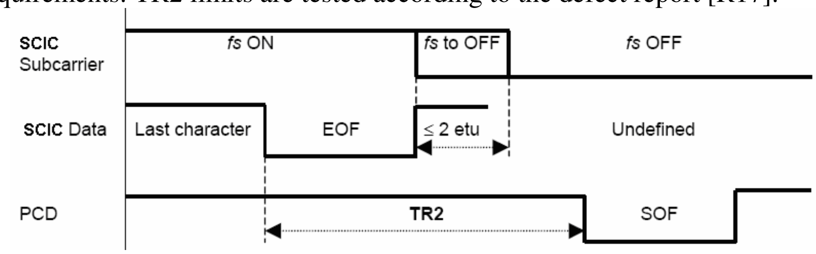

Figure 8: TR2

Table 12: TR2 limits

b3	b2	minimum TR2	maximum TR2
0	0	10etu +32/fs	n/a
0	1	10etu+128/fs	n/a
1	0	10etu+256/fs	n/a
1	1	10etu+512/fs	n/a

The bits b2 and b3 are negotiated in the SCIC ATQB 'Protocol Type' half byte.

Conditions: Minimum number of samples: 1

Parameters: Bit rate: fc/128, fc/64, fc/32, fc/16

Report: The test report SHALL include the number of passed tests versus the total number of tests, a test description, and the number of different samples and the date of the tests.

5 Layer 3 and layer 4 protocol tests

These tests provide a basic set of tests to be performed to check compliance to ISO/IEC 14443 protocol layers. All tests are based on and SHALL be evaluated according to the current ISO 10373-6 standards.

For all test cases, make sure that the physical and electrical tests as mentioned in the chapters above have passed.

Test setup:
Setup as defined for the electrical tests SHALL be used and is afterwards called “test apparatus”. All tests SHALL be performed with one specific field strength between 2 A/m and 7 A/m within the operating volume of the PCD if not further specified.

All tests SHALL be performed at RT if not further specified.

The test apparatus SHALL be able to emulate the protocol, to measure, monitor the timing of the logical Input/Receive line relative to the CLK frequency, and be able to analyze the I/O-bit stream in accordance with the protocol.

The tests are based on the ISO/IEC10373-6/Amd 3 [R13] specification. For the test commands defined in [R13], typical commands SHOULD be used. This could be for example for TEST_COMMAND1 the READ BINARY command. Other commands specified dependent on their expected behaviour might also be used. The command used MAY differ between different products and SHALL be documented in the report.

The functionality as described in clause 2.1.1 “Test setup” SHALL be used either with the final operating system (preferred way) or with dedicated test commands.

5.1 Type A activation

These tests SHALL ensure the start-up and the activation is according to [R6] (chapter 6). These tests are split up to collision handling, the handling of RATS and PPS, and the handling of CID during activation.

5.1.1 Handling of collisions

Test description:	The purpose of this test is to check the correct behaviour on collisions as defined in [R6]. The tests specified in the sub-clause “Handling of bit collision during ATQA” (chapter H.2.3) and “Handling of anticollision loop” (chapter H.2.4) of [R13] SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R6] respectively to [R13] or not. Additionally possible proprietary paths of the “Select sequence flow chart” specified in [R6] (chapter 6.4.1) SHALL not negatively affect the report. The report SHALL include the number of samples tested and the date.

5.1.2 Handling of RATS (including frame size selection)

Test description:	The purpose of this test is to check the correct behaviour of RATS and the handling of ATS as defined in [R7] (chapter 5.6.1.1). The tests specified in the sub-clause “Handling of RATS and ATS” (chapter H.2.5) and “Frame size selection mechanism” (chapter H.2.7) of [R13] SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R7] respectively to [R13] or not. The report SHALL include the number of samples tested and the date.

5.1.3 Handling of PPS

Test description:	The purpose of this test is to check the correct behaviour on handling a PPS response as defined in [R7] (chapter 5.6.2.1). The tests specified in the sub-clause “Handling of PPS response” (chapter H.2.6) of [R13] SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R7] respectively to [R13] or not. The report SHALL include the number of samples tested and the date.

5.1.4 Handling of CID during activation

Test description:	The purpose of this test is to check the correct behaviour on handling CID during activation as defined in [R7] (chapter 5.6.3). The tests specified in the sub-clause “Handling of the CID during activation by PCD” [R13] (chapter H.2.9) of SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R7] respectively to [R13] or not. The report SHALL include the number of samples tested and the date.

5.2 Type B activation

5.2.1 Frame size selection

Test description:	The purpose of this test is to check the correct behaviour of the frame size selection mechanism as defined in [R6] (chapter 7.9). The tests specified in the sub-clause “Frame Size Selection Mechanism” (chapter H.3.2) of [R13] SHOULD be used.
Report:	The test report SHALL state whether the response was according to the scenario defined in [R13] (chapter H.3.2). The report SHALL include the number of samples tested and the date.

5.2.2 Bit rate selection

Test description:	The purpose of this test is to check the correct behaviour of the bit rate selection mechanism as defined in [R9]. The tests specified in [R15] (Annex K) SHOULD be used.
Report:	The test report SHALL state whether the behaviour was according to [R9] respectively [R15]. The report SHALL include the number of samples tested and the date.

5.2.3 Handling of CID during activation

Test description:	The purpose of this test is to check the correct behaviour on handling CID during activation as defined in [R7]. The tests specified in the sub-clause "Handling of the CID during activation by the PCD" (chapter H.3.3) of [R13] SHOULD be used.
Report:	The test report SHALL state whether the response was according to [R7] respectively to [R13] or not. The report SHALL include the number of samples tested and the date.

5.3 Handling of the polling loop (Type A and B)

Test description:	The purpose of this test is to check the correct behaviour during polling for Type A and Type B SCIC's as defined in [R6] (chapter 5). The test specified in the sub-clause "Handling of the polling loop" of [R13] (chapter H.4.1) SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R6] respectively to [R13] or not. The report SHALL include the command set used for testing.

5.4 Data exchange protocol tests (Type A and B)

Data exchange protocol tests SHALL ensure the logical operation of the PCD is according to [R7]. These tests are valid for both, Type A, Type B whereas the activation before running these tests is different, and listed below. All tests are based on the currently available standards.

The activation for Type A SHALL be:

1. Configuration to emulate Type A protocol

2. Activation according to [R6] (Request, Anticollision loop, Select)

3. Activation according to [R7] (RATS, optional PPS)

The activation for Type B SHALL be:

1. Configuration to emulate Type B protocol

2. Activation according to [R6] (Request, Attrib, optional Slot-marker)

5.4.1 Error detection and recovery

The purpose of this test is to check the behaviour of PCD when transmission errors occur according to [R7] (chapter 7.5). These tests specified in [R13] cover standard communication blocks, blocks where the PCD uses chaining and blocks where the SCIC uses chaining.

The SCIC chaining tests could be performed without knowing dedicated command behaviour on the device under test. Any command could be divided into two parts e.g. the response to a READ BINARY could be sent in two chained packets.

The PCD chaining is harder to achieve. If the higher layer functionality is not known in detail or a chaining command is not used in the application these tests could not be performed. Therefore, it is optional.

Test description:	The purpose of this test is to check the behaviour of the PCD when transmission errors occur as defined in [R7] (chapter 7.5). The test specified in the sub-clause “Error detection and recovery” of [R13] (chapter H.4.3) SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R7] respectively to [R13]. The report SHALL include the command set used for testing.

5.4.2 Request for waiting time extension

Test description:	The purpose of this test is to check the behaviour of the PCD when the SCIC use a request for waiting time extension as defined in [R7] (chapter 7.3). The test specified in the sub-clause “Reaction of the PCD to request for waiting time extension” of [R13] (chapter H.4.2) SHOULD be used. The detailed test procedure is not specified herein.
Report:	The test report SHALL state whether the response was according to [R7] and [R13]. The report SHALL include the command set used for testing.

Annex A Measurement points

Table 13: Application specific measurement points

Application	Reader design type		Volume definition			Measurement points	
			X dimension	Y dimension	Z dimension	X-Y-plane	Height
ePassport	1	Single step readers[1]	Twice ID3 + 20 % (of ID3) for the size of an open passport booklet (ID3) enlarged by 10 %	Size of a passport booklet (ID3) enlarged by 10 %	7.5 mm	All four corners of both connected ID 3 + 10 % fields, additionally in the centre of both fields	Z_0 = 0 mm; Z_1 = 7.5 mm; Z_{max} as specified by manufacturer (if Z_{max} > 7.5 mm)
ePassport	1a	Similar to reader design type 1, but the two parts of the volume are arranged angularly, not in-line					
ePassport	2	Full page readers[2]	ID3 + 10 % for the size of a passport booklet (ID3) enlarged by 10 %	Size of a passport booklet (ID3) enlarged by 10 %	7.5 mm	All four corners of the ID 3 + 10 % field, additionally in the centre of the field	Z_0 = 0 mm; Z_1 = 7.5 mm; Z_{max} as specified by manufacturer (if Z_{max} > 7.5 mm)

[1] Document readers that are able to read the entire data page of an opened passport and that are able to read the data from the SCIC without any replacement of the passport on the reader, independently from the location of the chip inside the passport document (i.e., front cover, back cover, data page, middle page)

[2] Document readers that are able to read the entire data page of an opened passport as well as the data contained in the SCIC. A replacement of the passport may be required.

Application	Reader design type		Volume definition			Measurement points	
			X dimension	Y dimension	Z dimension	X-Y-plane	Height
ePassport	99	Other readers	ID3 +10 % for the size of a passport booklet (ID3) enlarged by 10 %	Size of a passport booklet (ID3) enlarged by 10 %	20 mm	All four corners of the ID 3 + 10 % field, additionally in the centre of the field	Z_0 = 0 mm; Z_1 = 7.5 mm; Z_{max} = 20 mm or as specified by manufacturer (if $Z_{max} > 20$ mm)

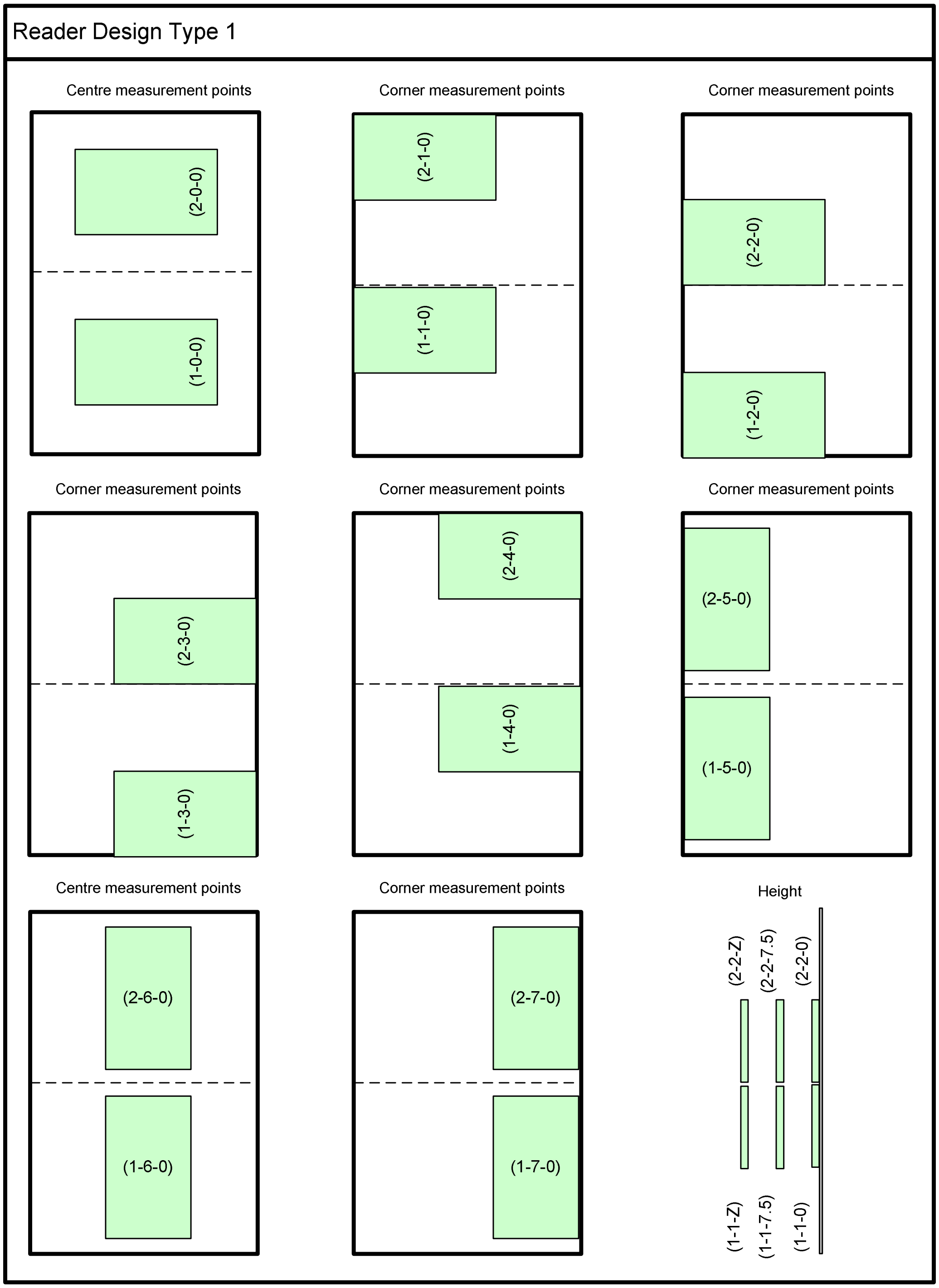

Figure 9: Measurement points e-Passport reader design type 1

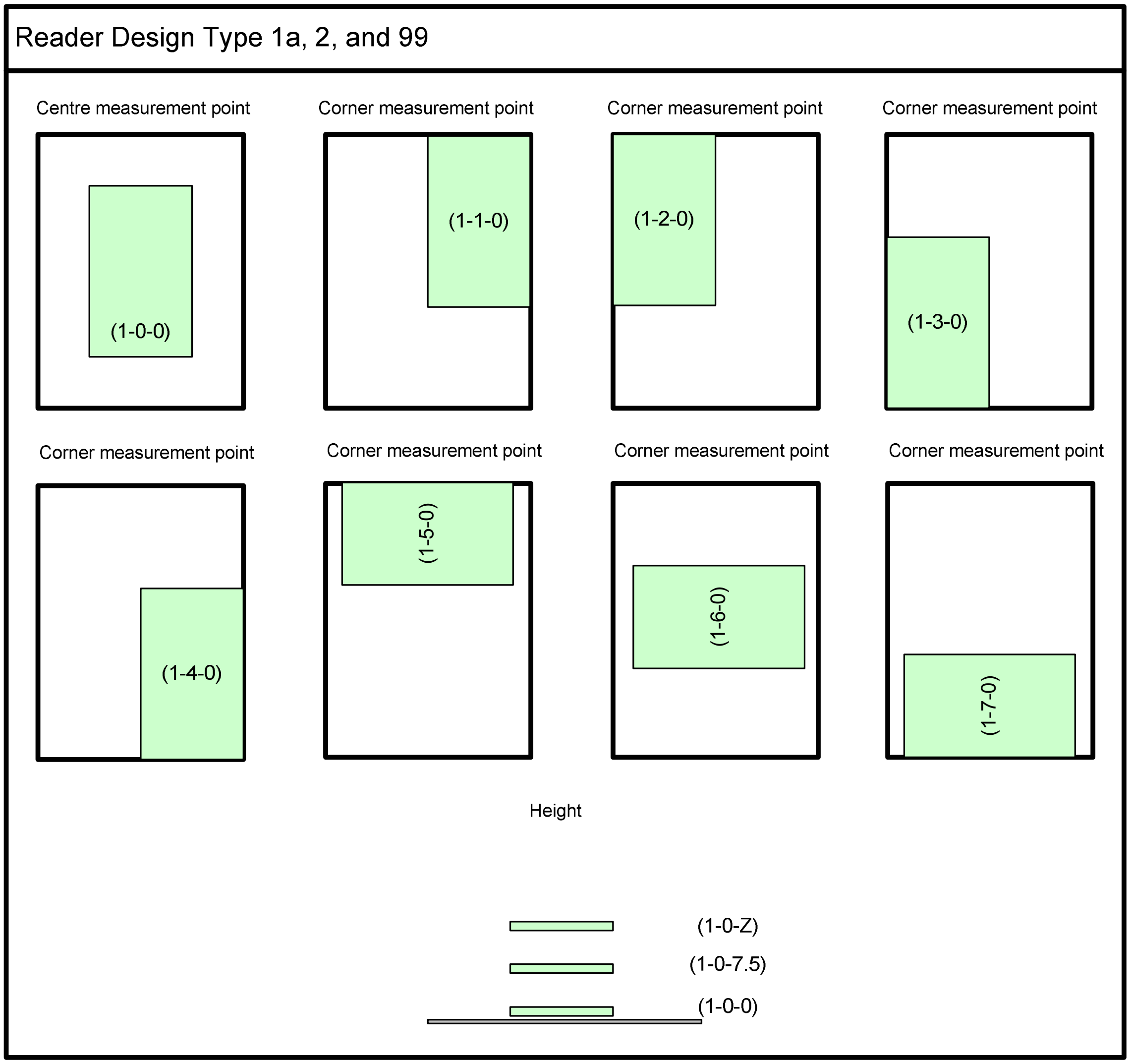

Figure 10: Measurement points e-Passport reader design type 1a, 2, and 99

Annex B Comparison PCD – e-Passport test

Table 14: Comparison PCD – e-Passport Test

ISO Layer	Test description	e-Passport	PCD
1	Coil Dimension Check	Conditional	No
	Static Electricity (ESD) Tests	Yes	No
	X-Ray Tests	No	No
	UV Tests	No	No
	H_{max} (12A/m) Test	Yes	No
2	Load modulation amplitude	Check if the load modulation amplitude is higher than the limit	Check if the PCD is able to handle the whole load modulation amplitude range
	Operating field strength	Check the whole range from 1.5 – 7.5 A/m	Check if the available power of the PCD is in-between 1.5 and 7.5 A/m
	Communication stability	Check the whole range of valid signal shapes, (modulation index, rising- and falling edges, overshoot and timings)	Check if the shape of the modulated field is in-between the valid limits (modulation index, rising- and falling edges, overshoot and timings)
	Threshold resonance frequency	Optional	No
3	Frame Delay Time (Type A)	Check if the response starts after the right time.	Check the whole range, ±200 ns
	Bit Boundaries (Type B)	No	Check if the bit boundaries are within the valid limits.
	Start-of-Frame & End-of-Frame Timings (SOF & EOF) (Type B)	Check if the SOF and EOF timings are within the valid limits.	Check if the SOF and EOF timings are within the valid limits.
	Extra Guard Time (EGT) (Type B)	Check if the EGT timing is within the valid limits.	Check if the EGT timing is within the valid limits.
	Timing Before SCIC Start of Frame (TR0 & TR1) (Type B)	Check if the TR0 & TR1 timings are within the valid limits.	Check if the TR0 & TR1 timings are within the valid limits.
	Timing Before PCD Start of Frame (TR2) (Type B)	Check if the TR2 timing is within the valid limits.	Check if the TR2 timing is within the valid limits.

Annex C Interface definition of TM-PDUs (informative)

C.1.1 Scope

This annex specifies a test management protocol according to ISO10373-6:AM3 [R13] to be applied to an e-Passport reader during conformity evaluations. The test commands, called test management protocol data units (TM-DPU), follow the architecture described in the PC/SC Part 3: Requirements for PC-Connected Interface Devices, Revision 2.01.05.

These test commands are persistent in each certified reader to perform tests in the field as well.

This PC/SC implementation is optional. Using the PC/SC framework, test houses can minimize their efforts to establish the evaluation environment. Reader providers can use universal test applications if there are any.

If a supplier provides a PC/SC implementation, it has to be compliant to the latest released version on the consortium website (http://www.pcscworkgroup.com/specifications/overview.php).

C.1.2 Command syntax and transportation

The structure of the TM-PDUs follows the byte sequential command structure for smart cards, so called APDUs (Application Protocol Data Unit) according to ISO-7618-4. The APDUs are distinguished by their direction, to or from the PCD:

- Command-APDU (C-APDU)
- Response-APDU (R-APDU)

For each C-APDU sent to the PCD by the test application, an R-APDU will be returned as a confirmation. R-APDUs will not be confirmed by the PCD.

C.1.2.1 Command APDU

C-APDUs are byte sequences consisting of two parts: Header and Body.

Table 15: Structure of a C-APDU

<table>
<tr><th colspan="7"><C-APDU></th></tr>
<tr><td colspan="7">Length: 4 ... ((0 … 3) + Lc + (0 … 3))</td></tr>
<tr><td colspan="4">Header
mandatory</td><td colspan="3">Body
[optional]</td></tr>
<tr><td colspan="4">Length: 4</td><td colspan="3">Length: 0 ... ((0 ... 3) + Lc + (0 ... 3))</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5 ... 7</td><td>((5 ... 7) + 1) ...
(Lc + ((5 ... 7) + 1))</td><td>(Lc + ((5 ... 7) + 2)) ...
(Lc + ((5 ... 7) + 2) +
(1 ... 3))</td></tr>
<tr><td rowspan="2">CLA</td><td rowspan="2">INS</td><td rowspan="2">P1</td><td rowspan="2">P2</td><td>[Lc]</td><td rowspan="2">[Data Field
<Lc Bytes of Data>]</td><td>[Le]</td></tr>
<tr><td>Max. Value indicated
by
Lc < = 65535</td><td>Max. Value indicated
by
Le < = 65536</td></tr>
</table>

The header only (first four bytes) is mandatory. The body is optional and MAY contain data with prior length indicator and/or length indicator for expected R-APDU. The presence depends on the command and application case or context.

C.1.2.2 Response APDU

The response to a C-APDU will be returned in general as R-APDU. Optionally the R-APDU MAY contain data. The two status bytes SW1SW2 are mandatory.

Table 16: Structure of the R-APDU

<R-APDU>		
Body [optional]	**Trailer mandatory**	
Requested Information	Status Word	
Length: 0 ... [<Le> of C-APDU]	Length: 2 Byte	
Position: 1 ... (1 + [<Le> of C-APDU])	Pos.: 1 ... (1 + [<Le> of C-APDU])	Pos.: 2 ... (2 + [<Le> of C-APDU])
[Information Field]	SW1	SW2

The trailer of the R-APDU transports the result of an operation. It has to be interpreted byte wise. SW1 classifies the result in general and SW2 gives an exact value for the indicated error class. The following classes are defined:

Table 17: Return Code Classes

Class (SW1)	Description
‘90’	Normal Processing
‘62’, ‘63’, ‘6C’	Warning
‘64’, ‘65’	Execution Error
‘67’ to ‘6F’	Checking Error

The value of SW2 is class specific.

C.1.2.3 Common return codes

Table 18: Common return codes

	SW1	SW2	Description
Success	‘90’	‘00’	Command successful
Warning	‘6C’	‘XX’	Le and available data are not same; ‘XX’ is the number of available data. If Le= 0x00, all available data is returned.
Error	‘64’	‘00’	Timeout, expecting response from card but no response within the time
	‘64’	‘01’	Internal error

	SW1	SW2	Description
	'67'	'00'	Wrong length
	'68'	'00'	Class byte is not correct
	'6A'	'81'	INS not supported
	'6A'	'82'	Function is not supported
	'6B'	'00'	Wrong parameter P1-P2

C.1.2.4 Command transportation

The commands can be transported via any interface. The applicant SHALL enable the test house to issue the specified test commands.

The applicant MAY provide the test houses with PC/SC drivers. In this case, all commands will be transported using the SCARD_Transmit function.

A reader (or its related driver) can distinct between e-Passport commands and PCD control commands via the class byte. The value 0xFF is reserved for other purposes and can never be used by an ISO compliant e-Passport. Therefore, the test commands can be sent on the same 'channel' as the e-Passport commands. In addition, there is no need for a special function call or address to indicate control commands.

Following this way, existing test equipment can be used to apply reader test commands and real e-Passport commands in parallel for testing.

C.1.3 Commands for testing

The following commands MAY be used to test the PCD.

Note: All RFU bits and bytes described here must be set to 0.

C.1.3.1 ISO 14443-2 test command

This command MAY be used to test the RF interface, the modulation index, framing and coding of the data, transmitted by the PCD. Testing with this command does not mandate the presence of an e-Passport but it MAY be inserted to the field in order to check the mutual induction to the magnetic field.

Table 19: C-APDU for ISO14443-2 Test Command

Command	CLA	INS	P1	P2	LC	Data in	Le
ISO14443-2 Test	0xFF	0x92	xx	RFU	xx	xx	-

Table 20: R-APDU for ISO14443-2 Test Command

Data out	SW1SW2
-	xxxx

Table 21: P1-Subcarrier and Data Coding Parameter

<table>
<tr><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td colspan="6">------</td><td>No carrier, RF is turned off</td></tr>
<tr><td>0</td><td>1</td><td colspan="6">------</td><td>No sub-carrier, just carrier, RF is turned on</td></tr>
<tr><td>1</td><td>0</td><td colspan="6">------</td><td>Carrier modulated with sub-carrier, if there are some bytes to transmit, Lc = n, n≠ 0</td></tr>
<tr><td>1</td><td>1</td><td colspan="6">------</td><td>RFU</td></tr>
<tr><td colspan="2">-----</td><td rowspan="2">RFU</td><td>0</td><td colspan="4">-------</td><td>ISO14443A type transmission *</td></tr>
<tr><td colspan="2">-----</td><td>1</td><td colspan="4">-------</td><td>ISO14443B type transmission *</td></tr>
<tr><td colspan="4">-----</td><td colspan="2" rowspan="4">RFU</td><td>0</td><td>0</td><td>Transmission at 106 kbps</td></tr>
<tr><td colspan="4">-----</td><td>0</td><td>1</td><td>Transmission at 212 kbps</td></tr>
<tr><td colspan="4">-----</td><td>1</td><td>0</td><td>Transmission at 424 kbps</td></tr>
<tr><td colspan="4">-----</td><td>1</td><td>1</td><td>Transmission at 848 kbps</td></tr>
</table>

Note: * The transmission according to the normal frame, which includes all framing e.g. start bit, stop bit, parity bit, SOF, EOF, CRC etc.

The following table lists the return codes in addition to the common return codes:

Table 22: Return Codes of ISO14443-2 Test Command

SW1SW2	Meaning
‘6A83’	Transmission type not supported
‘6A84’	Transmission speed is not supported

C.1.3.2 Transmit pattern and receive 14443 command

This command transmits a bit pattern independent from any SCIC command structure and coding. The pattern can be used to measure modulation index, rise and fall times, overshoots etc. This command can be used to test by using a reference e-Passport to receive any pattern by the SCIC.

Table 23: C-APDU for Transmit Pattern and Receive 14443

Command	CLA	INS	P1	P2	LC	Data in	Le
Tx Pattern	0xFF	0x94	xx	xx	xx	Pattern	xx

Table 24: R-APDU for Transmit Pattern and Receive 14443

Data out	SW1SW2
-	xxxx

Table 25: P1 of Transmit Pattern and receive 14443

b7	b6	b5	b4	b3	b2	b1	b0	Description
0	0	0	0	-	-	-	-	RFU
-	-	-	-	-	-	-	0	ISO 14443A type Transceive data coding
-	-	-	-	-	-	-	1	ISO 14443B type Transceive data coding
				xxx			-	No of bits of last byte will be transmitted, 0 means all bits will be transmitted

Note: The data in the 'data in' field is not interrupted; the complete data is sent to the air. No framing, e.g. start bit, stop bit, CRC, SOF, EOF is not added.

Table 26: P2 of Transmit Pattern and Receive (bit rate) 14443

b7	B6	b5	b4	b3	b2	b1	b0	Description
RFU		-	-	-	RFU	0	0	Transmit @106 kbps
		-	-	-		0	1	Transmit @212 kbps
		-	-	-		1	0	Transmit @424 kbps
		-	-	-		1	1	Transmit @848 kbps
		RFU	0	0	-	-	-	Receive @106 kbps
			0	1				Receive @212 kbps
			1	0				Receive @424 kbps
			1	1				Receive @848 kbps

Table 27: Return codes of Transmit Pattern and Receive 14443

SW1SW2	Meaning
'6A83'	Transmission type not supported
'6A84'	Transmission speed is not supported
'6A87'	Different bit rate is not supported
'6A8A'	Modulation index is not supported

C.1.3.3 ISO 14443-3 test command

This command transmits a 14443-3 command and returns the data received from the SCIC.

Table 28: C-APDU for ISO14443-3 Test Command

Command	CLA	INS	P1	P2	LC	Data in	Le
ISO14443-3 Test	0xFF	0x96	xx	xx	xx	xx	xx

Table 29: R-APDU for ISO14443-3 Test Command

Data out	SW1SW2
Response of the card	xxxx

Table 30: P1: Command byte

b7	b6	b5	b4	b3	b2	b1	b0	Description
--			0	ISO 14443A type command				
				0	0	0	1	REQA
				0	0	1	0	WUPA
				0	0	1	1	HLTA
				0	1	0	0	PCD does complete part 3 A type, returns UID+SAK
				1	Anti-collision is handled by user			
					0	0	1	ANTICOLLISION Sel_level 1
					0	1	0	ANTICOLLISION Sel_level 2
					0	1	1	ANTICOLLISION Sel_level 3
					1	0	0	SELECT (Data in field must be 70+ last 4-byte UID + BCC)
				Other values are RFU				
			1	ISO 14443B type command				
				0	0	0	1	REQB (P2 sets the number of slot)
				0	0	1	0	WUPB (P2 sets the number of slot)
				0	0	1	1	HLTB
				0	1	0	0	Slot-MARKER (slot number in P2)
				0	1	0	1	ATTRIB (P2 sets the communication speed)
	xx		----					No of repetitions of the command
x	---		----					0 means set all other parameters to the default value which is not stated in P2. 1 means send all data given in 'Data in' field with the command, P2 has no significance.

Lc: No of bytes has to be sent to the card within this command except the command itself.

Data in: The data byte has to be sent within the command.

Table 31: P2: Coding of Bit rate for ATTRIB command

b7	b6	b5	b4	b3	b2	b1	b0	Description
					RFU	0	0	PCD to SCIC 106 Kbps
						0	1	PCD to SCIC 212 Kbps
						1	0	PCD to SCIC 424 Kbps
						1	1	PCD to SCIC 848 Kbps
		RFU	0	0		--		SCIC to PCD 106 Kbps
			0	1		--		SCIC to PCD 212 Kbps
			1	0		--		SCIC to PCD 424 Kbps
			1	1		--		SCIC to PCD 848 Kbps
CID								Logical card identifier *

Only four cards are supported, for more cards, select option b7 of P1 =1.

Table 32: P2: Coding of Slot number of REQB/WUPB command

b7	b6	b5	b4	b3	b2	b1	b0	Description
RFU					xxx			Number of slot. $N = 2^{(b2b1b0)}$; (b2b1b0 = 0 means $N = 2^0 = 1$)

Table 33: P2: Coding of slot number of Slot-MARKER command

b7	b6	b5	b4	b3	b2	b1	b0	Description
RFU				0	0	0	1	Slot number = 2
				0	0	1	0	Slot number = 3
				..	..	..	..	………………..
				1	1	1	1	Slot number = 16

The return codes in addition to the common return codes are listed in the following table:

Table 34: Return codes of ISO14443-3 Test Command

SW1SW2	Meaning
'6A85'	The command is not supported
'6A86'	The repetition is not allowed
'6A87'	Different bit rate is not supported
'6A88'	Requested buffer size is bigger than the PCD buffer size

C.1.3.4 ISO 14443-4 test command

This command transmits a 14443-4 command and returns the data received from the SCIC.

Table 35: C-APDU for ISO14443-4 Test Command

Command	CLA	INS	P1	P2	LC	Data in	Le
ISO14443-4 Test	0xFF	0x98	xx	xx	xx	xx	xx

Table 36: R-APDU for ISO14443-4 Test Command

Data out	SW1SW2
Response of the card**	xxxx

** If there is a card response SW1SW2, this is considered as the ‘Response of the card’ data, e.g. if a ‘select file’ command returns ‘6A82’, here the complete R-APDU will be ‘6A829000’, similarly if the card returns ‘9000’, the complete R-APDU will be ‘90009000’.

Table 37: P1 is structured as follows

<table>
<tr><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th><th>Description</th></tr>
<tr><td colspan="6" rowspan="4">RFU</td><td>0</td><td>0</td><td>RATS (Type A only), P2 codes the parameter (FSDI and CID) according to ISO 14443-4</td></tr>
<tr><td>0</td><td>1</td><td>PPS (Type A only), P2 codes the communication speed</td></tr>
<tr><td>1</td><td>0</td><td>The complete data of ‘Data in’ field is transmitted to the card, user must add PCB and CID according to T=CL protocol.</td></tr>
<tr><td>1</td><td>1</td><td>The complete data of ‘Data in’ field is transmitted to the card, PCD must take care of the PCB and CID, only I block is allowed in this mode.</td></tr>
</table>

P2: Is coded according to P1 or ‘00’.

Note: CRC is calculated by the PCD and appended to the end of the frame.

Table 38: P2 coded the communication speed for PPS command

b7	b6	b5	b4	b3	b2	b1	b0	Description
					RFU	0	0	PCD to SCIC 106 Kbps
						0	1	PCD to SCIC 212 Kbps
						1	0	PCD to SCIC 424 Kbps
						1	1	PCD to SCIC 848 Kbps
		RFU	0	0	--			SCIC to PCD 106 Kbps
			0	1	--			SCIC to PCD 212 Kbps
			1	0	--			SCIC to PCD 424 Kbps
			1	1	--			SCIC to PCD 848 Kbps
RFU								---

Table 39: P2 coded the Parameter byte of RATS command

b7	b6	b5	b4	b3	b2	b1	b0	Description
FSDI				CID				FSDI is according to the following table

Table 40: Coding of FSDI

FSDI	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’-‘F’
FSD	16	24	32	40	48	64	96	128	256	RFU

Table 41: Return codes of ISO14443-4 Test Command

SW1SW2	Meaning
‘6A87’	Different bit rate is not supported
‘6A88’	Requested buffer size is bigger than the PCD buffer size

C.1.3.5 Miscellaneous command

This command can be used for proprietary purposes by the PCD vendor. A separated description has to be added by the vendor.

Table 42: Miscellaneous Command

Command	CLA	INS	P1	P2	LC	Data in	Le
Miscellaneous	0xFF	0x9A	xx	xx	xx	xx	xx

As example to retrieve the PCD information, the command MAY be interpreted as follows:

Table 43: P1 structure of the Miscellaneous Command

b7	b6	b5	b4	b3	b2	b1	b0	Description
0	0	0	0	0	0	0	1	Returns PCD information as requested through P2
0	0	0	0	0	0	1	0	Generate trigger signal for ISO 14443A*
0	0	0	0	0	0	1	1	Generate trigger signal for ISO 14443B*
RFU								Other values are RFU

*Reader manufacturer will provide a description about the trigger signal and the port/pin to acquire it.
P2 = ‘00’.

Table 44: P2 structure of the Miscellaneous Command for P1 = ‘01’

b7	b6	b5	b4	b3	b2	b1	b0	Description
0	0	0	0	0	0	0	1	Returns vendor name
0	0	0	0	0	0	1	0	Returns vendor ID
0	0	0	0	0	0	1	1	Returns product name
0	0	0	0	0	1	0	0	Returns product ID
0	0	0	0	0	1	0	1	Returns product serial number
0	0	0	0	0	1	1	0	Returns product firmware version
0	0	0	0	0	1	1	1	Returns driver version
0	0	0	0	1	0	0	0	Returns PCD buffer size
0	0	0	0	1	0	0	1	Returns maximum bit rate supported by PCD
RFU								Other values are RFU

Note: All return values are in ASCII string format.

Table 45: Return codes of the Miscellaneous Command

SW1SW2	Meaning
‘6A89’	Information not available
‘6A90’	Trigger signal is not supported

Annex D Hilbert transformation (informative)

Example program for the evaluation of the waveform and modulation index:

The following programs written in C language extract the envelope of the modulated carrier using Hilbert transformation.

```
/ * * *                                                                         * * * /
/ * * * This program extract the envelope of modulated carrier                  * * * /
/ * * * Input:                                                                  * * * /
/ * * * File in text  format containing a table of two columns                  * * * /
/ * * * (time and test PCD output voltage vd)                                   * * * /
/ * * *                                                                         * * * /
/ * * * Data format of input-file:                                              * * * /
/ * * *                                                                         * * * /
/ * * * One data-point per line,                                                * * * /
/ * * *                                                                         * * * /
/ * * * {time[seconds], sense-coil-voltage[volts])                              * * * /
/ * * *                                                                         * * * /
/ * * *  Data-points shall be equidistant time                                  * * * /
/ * * * Minimum sampling rate: 100 MSamples/second                              * * * /
/ * * * example for spreadsheet file (start in next line):                      * * * /
/ * * * (time)          , (voltage )                                            * * * /
/ * * * 3.00000e-06,1.00                                                        * * * /
/ * * * 3.00200e-06,1.01                                                        * * * /
/ * * *                                                                         * * * /
/ * * * Run:                                                                    * * * /
/ * * * hilberttransformation Filename.txt                                      * * * /
/ * * * or                                                                      * * * /
/ * * * hilberttransformation  (default file name input.txt)                    * * * /
/ * * *                                                                         * * * /
/ * * *                                                                         * * * /
/ * * * Description:                                                            * * * /
/ * * *        HilbertTransformation.c--- Main program for extracting envelope   * * * /
/ * * *        fftrm.c --- Code to perform fourier and inverse fourier           * * * /
/ * * *        transformation                                                   * * * /
/ * * *        fftrm.h --- Header file for fftrm.c                              * * * /
/ * * *                                                                         * * * /


/*************************************************************************************/
/*HilberTrnsformation.c                                                             */
/*Main program                                                                      */
/*************************************************************************************/


# include <stdio.h>
# include <math.h>
# include <malloc.h>
#include <ctype.h>
#include <string.h>
# include "fftrm.h"

#define MAX_POINT 5000
#define M_PI 3.14159265358979323846264338327950

int debug=0;
int fftdebug=0;

double *Gvalue;
double *Gtime;
double *Gr;
double *Gi;
double **G; /*Phase Changed*/
double *Gc;
doublecomplex *Gt_ifft;


/*File containing the input data*/

char *InputFileName     ="input.txt"  ;

/*This function reads the sampled data recorded in the file*/
int ReadData(void);
/*This function function performs the fourier transform*/
void Fft(void);
/*This function performs the necessary phase shift*/
```

```
void PhaseShifting(void);
/*This function performs the inverse fourier transfor*/
void Ifft(void);
/*Envelope reconstruction is done by this function*/
int EnvelopeReconstruction(void);

/*For fourier and inverse fourier transformation these two functions are used */
/*These functions are defined in fftrm.c                                      */
int zffts ( int debug,doublecomplex *X,int M );  /*Defined in fftrm.c*/
int ziffts( int debug,doublecomplex *X,int M );  /*Defined in fftrm.c*/


int SampledPoints=0;
int N;
int row;
const int col=2;

int ReadData(void)
{
        float a,b;
        int i=0,num1;
        FILE *fp1;
        i=0;

        if ((fp1 = fopen(InputFileName,"r")) == NULL)
        {
                printf("Cannot open input file.\n");
                return 1;
        }


        printf("\nReading data from file ... ... ...%s\n",fp1);

        while(!feof(fp1))
        {

                 fscanf(fp1,"%e,%e\n", &a, &b);

                 Gtime[SampledPoints]  = a;
                 Gvalue[SampledPoints] = b;
                 SampledPoints++;
                 if (SampledPoints>= MAX_POINT) break;

        }

        fclose(fp1);

        fp1=fopen("inputfile.txt","w");
    if (!fp1)
        {
         fprintf(stdout,"Cann't write the sampled data in inputfile.txt. \n");

                return 1;

        }
    for(i=0; i<SampledPoints; i++)
         fprintf(fp1,"%e\n",Gvalue[i]); /*Gtime[i] has been omitted*/
    fclose(fp1);

        if(debug)
        {
        fp1=fopen("inputtime.txt","w");
    if (!fp1)
        {
         fprintf(stdout,"Cann't write the sampled data in inputtime.txt. \n");

                return 1;

        }
    for(i=0; i<SampledPoints; i++)
         fprintf(fp1,"%e\n",Gtime[i]); /*Gtime[i] has been omitted*/
    fclose(fp1);
        }

        if(debug)
        {
        if((fp1=fopen("inputfile.bin","wb"))!=NULL)  {
                num1=fwrite(Gvalue,sizeof(double),SampledPoints,fp1);
        fclose(fp1);
        }
        }

        if(SampledPoints<N)
        {
```

```
                for(i=SampledPoints;i<=N;i++)
                {
                        Gvalue[i] = 0;
                }
        }

        fprintf(stdout,"\nInput file name = %s\n",InputFileName);
       fprintf(stdout,"Number of sampled data = %d\n",SampledPoints);
        return 0;

}/*End Of Function ReadData;*/


void Fft(void)
{
        doublecomplex *Gt_freq;

        FILE *fp1,*fp2,*fp3;
        int k,num1,num2,num3,z1;

        Gt_freq = (doublecomplex *)calloc(sizeof(doublecomplex),row);

        printf("\nPerforming FFT ... ... ...\n");

        /* FFT Procedure Starts for Sampled Data*/
        for(k=0;k<=N;k++){
                RE(Gt_freq[k])=Gvalue[k];
                IM(Gt_freq[k])=0.0;
        }

        if(debug){
                if((fp3=fopen("f.bin","wb"))!=NULL) {
                num3=fwrite(Gvalue,sizeof(double),row,fp3);
                fclose(fp3);
                }
        }

        z1=zffts(fftdebug,Gt_freq,row);/*FFT is done in spatial coordicate*/

        for (k=0;k<=N;k++)  {
                Gr[k]=RE(Gt_freq[k]);
                Gi[k]=IM(Gt_freq[k]);
        }
  /* FFT Procedure Ends for Sampled Data*/

  /* Writing The Real And Imaginary Part Of Reflected Part for Debuging*/
  /* Writing the real part of sampled data*/

        if(debug) {
                if((fp1=fopen("Gr.bin","wb"))!=NULL){
                        num1=fwrite(Gr,sizeof(double),row,fp1);
                        fclose(fp1);
                }
                else
                        fprintf(stdout,"Cann't Open Gr.bin");

                // Writing the img part of sampled data
                if((fp2=fopen("Gi.bin","wb"))!=NULL)  {
                        num2=fwrite(Gi,sizeof(double),row,fp2);
                        fclose(fp2);
                }
                else
                        fprintf(stdout,"Cann't Open Gi.bin");
                fprintf(stdout,"Num of Real Part Data after FFT  = %d\n",num1);
                        fprintf(stdout,"Num of Img  Part Data after FFT  = %d\n",num2);
        }

  free(Gt_freq);

}/* End Of The Function Fft */


void PhaseShifting(void)
{
        double *tempr, *tempi;
        int k,num1;
        FILE *fp1;

        printf("\nPerforming phase shift  ... ... ...\n");

        tempr = (double *)calloc(sizeof(double),row);
        tempi = (double *)calloc(sizeof(double),row);

        for ( k=0; k<=N; k++ )
```

```
        {
                tempr[k]=Gr[k];
                tempi[k]=Gi[k];
        }

        for ( k=0; k<=ceil(N/2); k++ )
        {
                Gr[k] =  tempi[k];
                Gi[k] = -tempr[k];
        }

        for ( k=(int)ceil(N/2)+1; k<=N; k++ )
        {
                Gr[k] = -tempi[k];
                Gi[k] = tempr[k];
        }

        if(debug){
                if((fp1=fopen("ffrpt.bin","wb"))!=NULL)      {
                        num1=fwrite(Gr,sizeof(double),row,fp1);
                        fclose(fp1);
                }
                if((fp1=fopen("ffipt.bin","wb"))!=NULL)      {
                        num1=fwrite(Gi,sizeof(double),row,fp1);
                        fclose(fp1);
                }
        }

        free (tempr);
        free (tempi);
}/*End of PhaseShift() function*/


void Ifft(void)
{
        double *Gt_tmp; /* It takes the real part of R_ifft*/
        double *Gt_tmpi;
        FILE *fp1;
        int k,i,z1,num1;

        Gt_tmp   = (double *)calloc(sizeof(double),row);
        Gt_tmpi   = (double *)calloc(sizeof(double),row);

        printf("\nPerforming IFFT ... ... ...\n");

        for (k=0;k<=N;k++){
                Gt_ifft[k].r=Gr[k];
                Gt_ifft[k].i=Gi[k];
        }

        z1=ziffts(fftdebug,Gt_ifft,row);/*IFFT of the signal in spatial coordinate*/

        printf("\nEnd of IFFT ... ...\n");

        for (k=0;k<=N;k++)  {
                Gt_tmp[k]=Gt_ifft[k].r;
        }

        if(debug){
        fp1=fopen("ifft.txt","w");
    if (!fp1)
         fprintf(stdout,"Cann't write in %s\n",fp1);
    for(i=0; i<=N; i++)
         fprintf(fp1,"%.4e\n",(Gt_ifft[i].r));
    fclose(fp1);
    }

        printf("\nPerforming IFFT writing... ... ...\n");

        if(debug){
                if((fp1=fopen("iffrpt.bin","wb"))!=NULL)     {
                        num1=fwrite(Gt_tmp,sizeof(double),row,fp1);
                        fclose(fp1);
                }
                if((fp1=fopen("iffipt.bin","wb"))!=NULL)     {
                        num1=fwrite(Gt_tmpi,sizeof(double),row,fp1);
                        fclose(fp1);
                }
        }

  free(Gt_tmp );
  free(Gt_tmpi );
}/* End Of Function Ifft*/
```

```
int EnvelopeReconstruction(void)
{
	FILE *fp1;
	int k;

	doublecomplex *G;		/*Input signal readed from input file in complex form*/
	doublecomplex *Ganalytical;/*Analytical function of our input signal*/

	double *test;
	double *sqrtr;
	double *sqrti;

	G = (doublecomplex *)calloc(sizeof(doublecomplex),row);
	Ganalytical = (doublecomplex *)calloc(sizeof(doublecomplex),row);

	test = (double *)calloc(sizeof(double),row);
	sqrtr=(double *)calloc(sizeof(double),row);
	sqrti=(double *)calloc(sizeof(double),row);

	printf("\nPerforming envelope extraction ... ... ...\n");

	for (k=0;k<=N;k++){
		RE(G[k]) = Gvalue[k];
		IM(G[k]) = 0.0;
	}

	for (k=0;k<=N;k++){
		RE(Ganalytical[k])=G[k].r;
		IM(Ganalytical[k])=Gt_ifft[k].r;
	}

	for (k=0;k<=N;k++){

	sqrtr[k]=sqrt(Ganalytical[k].r*Ganalytical[k].r+Ganalytical[k].i*Ganalytical[k].i);

	}
		fp1=fopen("output.txt","w");
		if (!fp1)
		{
			fprintf(stdout,"Cann't write extracted envelope in output.txt.\n");
			free(G);
			free(Ganalytical);
			free(test);
			free(sqrtr);
			free(sqrti);
			return 1;
		}
		for(k=0; k<SampledPoints; k++)
			fprintf(fp1,"%e,%e\n",Gtime[k],sqrtr[k]);

		printf("\nExtracted envelope is written in %s\n","output.txt");
		fclose(fp1);

	free(G);
	free(Ganalytical);
	free(test);
	free(sqrtr);
	free(sqrti);
	return 0;
}

/*Main Function*/

int main(int argc, char *argv[])
{
	int status=0,i=1;
	char fname[256],c;

	if(argc==2)
	{
		printf("\nInput File Name: ");
		//scanf("%s,InputFileName");
		strcpy(fname, argv[1]);
		InputFileName= fname;
		printf("%s\n",InputFileName);
	}
	else
	{
	printf("\nUse default file : %s\n", InputFileName);

	}

	//Reading the sampled data
```

```
        do
        {
                N=(int)pow(2,i)-1;
                i++;

        }while (MAX_POINT > N);

        if (debug)
        {
        printf("N= %d\n",N);
        }

        row=N+1;

        Gvalue = (double *)calloc(sizeof(double),row);
        Gtime  = (double *)calloc(sizeof(double),row);
        Gr     = (double *)calloc(sizeof(double),row);
        Gi     = (double *)calloc(sizeof(double),row);
        Gt_ifft= (doublecomplex *)calloc(sizeof(doublecomplex),row);
        Gc     = (double *)calloc(sizeof(double),row);

        status = ReadData();
        if (status== 1) goto MainExit;
        /*Does FFT*/
    Fft();
        /*Appropriate Phahe has been Shifted*/
    PhaseShifting();
        /*Does IFFT*/
     Ifft();
        /*Envelope Reconstruction */
    status = EnvelopeReconstruction();
        if (status== 1) goto MainExit;
        printf("\n\n==================================\n\n");
        printf("Input file name : %s \n",InputFileName);
        //printf("Gvalue[] has the input value\n");
        //printf("sqrtr[] has the output value\n");
        printf("Output file name output.txt\n");
        printf("\n==================================\n\n");

        MainExit:
        free(Gvalue);
        free(Gtime);
        free(Gr);
        free(Gi);
        free(Gt_ifft);
        free(Gc);
        printf("\n\nPress any key to exit.\n");

        scanf("%c",&c);

        return (0);
}/*End Of Main*/



/*********************************************************************************************/
/*fftrm.h                                                                                    */
/*This is the header file for fftrm.c                                                        */
/*********************************************************************************************/

#ifndef FFTRM_H
#define FFTRM_H


#define RE(z) ((z).r)
#define IM(z) ((z).i)

typedef float real;
typedef double doublereal;
typedef struct { real r, i; } complex;
typedef struct { doublereal r, i; } doublecomplex;

int zffts  (int debug, doublecomplex *X, int M);
int ziffts (int debug, doublecomplex *X, int M);
void  zfftrmc(doublecomplex *X, int M, int P, float D);
void rmpo  (int *rv, int *rvp );

#endif


/*********************************************************************************************/
/*fftrm.c                                                                                    */
/*This code contains the necessary function for fourier and inverse fourier transformation*/
/*********************************************************************************************/
```

```
#include <stdio.h>
#include <math.h>
#include <malloc.h>
#include "fftrm.h"
//#ifndef M_PI
#define M_PI 3.14159265358979323846264338327950
//#endif

float  *WR;
float  *WI;

doublereal  *DWR;
doublereal  *DWI;

void rmpo( int *rv, int *rvp )

{
  int   value_h;
  int   n;

  n       =  1;
  *rvp    = -1;
  value_h =  1;

  while ( value_h > 0 )  {

    value_h = *rv - n;

    (*rvp)++;

    n += n;
  }
}


void  zfftrmc( doublecomplex *X, int M, int P, float D )
{
  int    MV2,MM1,J,I,K,L,LE,LE1,IP,IQ,IND,IND1,R;
  int    I1,J1,IPOTR;

  float  A,B;
  float  WCOS,WSIN;
  float  VR,VI;
  float  ARG;

  static int   IPOTC;
  static float DALT;

  IPOTR = 0;

  DWR = (doublereal *)calloc(M,sizeof(doublereal));
  DWI = (doublereal *)calloc(M,sizeof(doublereal));

/*  if (IPOTC == P & D == DALT) goto warmstart;  */

  IPOTC = P;
  DALT  = (float)D;
  LE    = 1;
  IND   = 0;

  for (L=1;L<=P;L++)  {
    LE1     = LE;
    LE      = LE*2;
    DWR[IND] = 1.0;
    DWI[IND] = 0.0;
    ARG     = (float)M_PI/(float)LE1;
    WCOS    = (float)cos(ARG);
    WSIN    = (float)(D*sin(ARG));

    for (R=1;R<=LE1;R++)  {
      IND1     = IND+1;
      A        = (float)DWR[IND];
      B        = (float)DWI[IND];
      DWR[IND1] = A*WCOS - B*WSIN;
      DWI[IND1] = B*WCOS + A*WSIN;
      ++IND;
    }
  }

  /* warmstart: */

  MV2=M/2;
```

```
  MM1=M-1;
  J=1;

  for (I=1; I<=MM1; I++) {
    if (I >= J)
      goto P1;

    J1     = J-1;
    I1     = I-1;

    VR     = (float)RE(X[J1]);
    VI     = (float)IM(X[J1]);

    RE(X[J1]) = RE(X[I1]);
    IM(X[J1]) = IM(X[I1]);

    RE(X[I1]) = VR;
    IM(X[I1]) = VI;

P1: K      = MV2;
P2: if (K >= J) goto P3;
    J = J-K;
    K = K/2;
    goto P2;
P3: J = J+K;

  }

  IND = 0;
  LE  = 1;

  for (L=1; L<=P; L++)  {
    LE1 = LE;
    LE  = LE*2;

    for (R=0; R<LE1; R++)  {
      WCOS = (float)DWR[IND];
      WSIN = (float)DWI[IND];
      IND  = IND+1;

      for (IQ=R; IQ<M; IQ+=LE)  {

        IP     = IQ+LE1;

        A      = (float)RE(X[IP]);
        B      = (float)IM(X[IP]);

        VR     = A*WCOS - B*WSIN;
        VI     = B*WCOS + A*WSIN;

        RE(X[IP]) = RE(X[IQ]) - VR;
        IM(X[IP]) = IM(X[IQ]) - VI;

        RE(X[IQ]) = RE(X[IQ]) + VR;
        IM(X[IQ]) = IM(X[IQ]) + VI;
      }
    }
  }

  free(DWR);
  free(DWI);
}

/*============================================================================*/
/*___1-D FFT with respect to a spatial coordinate________________________________*/
/*============================================================================*/
int zffts( int debug, doublecomplex *X, int M )
{
  int   P;
  float D;

  D = -1.0;

  rmpo( &M, &P);

  if ( debug ) {
    printf("P = %d\n",P);
    printf("FFT ...\n");
  }

  zfftrmc( X, M, P, D);                                        /* fftrm.c */

  return 0;
}
```

```
/*============================================================================*/
/*___1-D Inverse FFT with respect to a spatial coordinate______________________*/
/*============================================================================*/
int ziffts( int debug, doublecomplex *X, int M )
{
  int   i;
  int   P;
  float D;

  D = 1.0;

  rmpo( &M, &P);

  if ( debug ) {
    printf("P = %d\n",P);
    printf("IFFT ...\n");
  }

  zfftrmc( X, M, P, D);                                      /* fftrm.c */


  /*___Multiply with 1/M____*/

  for (i=0; i<M; i++) {
    RE(X[i]) /= (doublereal)M;
    IM(X[i]) /= (doublereal)M;
  }

  return 0;
}/*End of fftrm.c*/
```