

INTERNATIONAL
STANDARD

ISO/IEC
14443-4

Fourth edition
2018-07

Cards and security devices for personal identification — Contactless proximity objects —

Part 4: Transmission protocol

*Cartes et dispositifs de sécurité pour l'identification personnelle —
Objets sans contact de proximité —*

Partie 4: Protocole de transmission

ISO/IEC 14443-4:2018(E)



Reference number
ISO/IEC 14443-4:2018(E)

© ISO/IEC 2018

Contents

	Page
Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols, abbreviated terms and notation	2
4.1 Symbols and abbreviated terms	2
4.2 Notations	4
5 Protocol activation of PICC Type A	5
5.1 Activation sequences	5
5.2 Request for answer to select	6
5.3 Answer to select	7
5.3.1 Structure of the bytes	8
5.3.2 Length byte	8
5.3.3 Format byte	8
5.3.4 Interface byte TA(1)	9
5.3.5 Interface byte TB(1)	9
5.3.6 Interface byte TC(1)	10
5.3.7 Historical bytes	11
5.4 Protocol and parameter selection request	11
5.4.1 Start byte	11
5.4.2 Parameter 0	11
5.4.3 Parameter 1	12
5.5 Protocol and parameter selection response	12
5.6 Activation frame waiting time	13
5.7 Error detection and recovery	13
5.7.1 Handling of RATS and ATS	13
5.7.2 Handling of PPS request and PPS response	13
5.7.3 Handling of the CID during activation	14
6 Protocol activation of PICC Type B	14
7 Half-duplex block transmission protocol	15
7.1 Elements and mechanisms	15
7.2 Block format	15
7.2.1 Length field	16
7.2.2 Prologue field	16
7.2.3 Information field	19
7.2.4 Epilogue field	19
7.3 Frame waiting time	19
7.4 Frame waiting time extension	20
7.5 Power level indication	21
7.6 Protocol operation	21
7.6.1 S(PARAMETERS) blocks	21
7.6.2 Multi-Activation	23
7.6.3 Chaining	23
7.6.4 Block numbering rules	24
7.6.5 Block handling rules	25
7.6.6 PICC presence check	26
7.6.7 Error detection and recovery	26
8 Protocol deactivation of PICC Type A and Type B	27
8.1 Deactivation frame waiting time	27
8.2 Error detection and recovery	27

9	Activation of bit rates and framing options in PROTOCOL state	27
10	Frame with error correction	30
10.1	General	30
10.2	Type A PCD frame format for bit rates up to $fc/16$ and higher than $fc/2$ and Type A PICC frame format for all bit rates	30
10.3	Type A PCD frame format for bit rates of $fc/8$, $fc/4$ and $fc/2$ and Type B PCD and PICC frame format for all bit rates	31
10.4	Enhanced block with error correction	31
10.4.1	General	31
10.4.2	Modified Hamming sub-block format	31
10.4.3	Hamming control byte	31
10.4.4	Hamming control generation matrix A	32
10.4.5	Hamming control bits calculation	32
10.4.6	Hamming control check matrix H	32
10.4.7	Error correction	33
10.5	Activation of frame with error correction in PROTOCOL state	33
	Annex A (informative) Multi-Accivation example	37
	Annex B (informative) Protocol scenarios	38
	Annex C (informative) Block and frame coding overview	47
	Annex D (deliberately left blank)	49
	Annex E (informative) CRC_32 encoding	50
	Annex F (informative) Frame with error correction	52
	Annex G (informative) Framing options	54
	Bibliography	55

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

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

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

This document was prepared by ISO/IEC JTC 1, *Information technology, SC 17, Cards and security devices for personal identification*.

This fourth edition cancels and replaces the third edition (ISO/IEC 14443-4:2016), which has been technically revised.

A list of all the parts in the ISO/IEC 14443 series can be found on the ISO website.

Introduction

The ISO/IEC 14443 series of standards describes the parameters for identification cards or objects for international interchange.

The protocol, as defined in this document, is capable of transferring the application protocol data units as defined in ISO/IEC 7816-4. Thus, application protocol data units and application selection may be used as defined in ISO/IEC 7816-4.

The ISO/IEC 14443 series of standards is intended to allow operation of proximity cards in the presence of other contactless cards or objects conforming to the ISO/IEC 10536 series of standards and the ISO/IEC 15693 series of standards and near field communication (NFC) devices conforming to ISO/IEC 18092 and ISO/IEC 21481.

Cards and security devices for personal identification — Contactless proximity objects —

Part 4: Transmission protocol

1 Scope

This document specifies a half-duplex block transmission protocol featuring the special needs of a contactless environment and defines the activation and deactivation sequence of the protocol.

This document is intended to be used in conjunction with other parts of ISO/IEC 14443 and is applicable to proximity cards or objects of Type A and Type B.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 7816-3, *Identification cards — Integrated circuit cards — Part 3: Cards with contacts — Electrical interface and transmission protocols*

ISO/IEC 7816-4:2013, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange*

ISO/IEC 14443-2¹⁾, *Cards and security devices for personal identification — Contactless proximity objects — Part 2: Radio frequency power and signal interface*

ISO/IEC 14443-3, *Cards and security devices for personal identification — Contactless proximity objects — Part 3: Initialization and anticollision*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

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

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

3.1

bit duration

one elementary time unit (etu), calculated by the following formula:

$$1 \text{ etu} = 128/(D \times fc)$$

Note 1 to entry: The initial value of the divisor D is 1, giving the initial etu as follows:

$$1 \text{ etu} = 128/fc$$

1) Fourth edition to be published. Current stage: 40.60.

where f_c is the carrier frequency as defined in ISO/IEC 14443-2.

3.2

block

special type of frame, which contains a valid protocol data format

Note 1 to entry: A valid protocol data format includes I-blocks, R-blocks or S-blocks.

3.3

invalid block

type of frame, which contains an invalid protocol format

Note 1 to entry: A time-out, when no frame has been received, is not interpreted as an invalid block.

3.4

frame

sequence of bits as defined in ISO/IEC 14443-3

Note 1 to entry: The PICC independent from its type may use the frame with error correction defined in [Clause 10](#). Alternatively, the PICC Type A can use one of the standard frames defined for Type A and the PICC Type B can use the frame defined for Type B. This Type B frame is called standard frame, too, within this document.

4 Symbols, abbreviated terms and notation

4.1 Symbols and abbreviated terms

A	Hamming control bits generation matrix (6 rows, 56 columns)
ACK	positive ACKnowledgement
ATS	Answer To Select
ATQA	Answer To reQuest, Type A
ATQB	Answer To reQuest, Type B
CID	Card IDentifier
CRC	Cyclic Redundancy Check, as defined for each PICC Type in ISO/IEC 14443-3
CRC1	most significant byte of CRC (b16 to b9)
CRC2	least significant byte of CRC (b8 to b1)
CRC_32	Cyclic Redundancy Check error detection code used within enhanced block
c _n	Hamming control bit n
<u>d</u>	vector containing 56 data bits
d _n	data bit n
D	Divisor
DR	Divisor Receive (PCD to PICC)
DRI	Divisor Receive Integer (PCD to PICC)
DS	Divisor Send (PICC to PCD)

DSI	Divisor Send Integer (PICC to PCD)
EDC	Error Detection Code
etu	elementary time unit
<i>fc</i>	carrier frequency
FSC	Frame Size for proximity Card
FSCI	Frame Size for proximity Card Integer
FSD	Frame Size for proximity coupling Device
FSDI	Frame Size for proximity coupling Device Integer
FWI	Frame Waiting time Integer
FWT	Frame Waiting Time
FWT _{TEMP}	temporary Frame Waiting Time
H	matrix needed to calculate Hamming syndrome \underline{s} (6 rows, 62 columns)
$h'_{m,n}$	element in row m and column n of matrix H'
H'	matrix needed to get matrix A (6 rows, 62 columns)
\underline{h}'_n	column vector of matrix H'
HLTA	HALT command, Type A
I_{6 × 6}	6 by 6 Identity matrix
I-block	Information block
INF	INFormation field
LEN	two bytes LENgth field used within enhanced block
m	row index
MAX	index to define a MAXimum value
MIN	index to define a MINimum value
n	column index
NAD	Node ADdress
NAK	Negative AcKnowledgement
OSI	Open Systems Interconnection
PCB	Protocol Control Byte
PCD	Proximity Coupling Device
PICC	Proximity card or object
PPS	Protocol and Parameter Selection

PPSS	Protocol and Parameter Selection Start
PPS0	Protocol and Parameter Selection parameter 0
PPS1	Protocol and Parameter Selection parameter 1
R-block	Receive ready block
R(ACK)	R-block containing a positive acknowledgement
R(NAK)	R-block containing a negative acknowledgement
RATS	Request for Answer To Select
REQA	REQuest command, Type A
RFU	Reserved for Future Use
<u>s</u>	6-bit vector containing Hamming syndrome
s'	error position code
s	error position
S-block	Supervisory block
SAK	Select AcKnowledge
SFGI	Start-up Frame Guard time Integer
SFGT	Start-up Frame Guard Time
SYNC	SYNChronization sequence
WUPA	Wake-UP command, Type A
WTX	Waiting Time eXtension
WTXM	Waiting Time eXtension Multiplier
<u>y</u>	64-bit vector (y' with no padding bits)
<u>y'</u>	64-bit vector containing received modified Hamming sub-block
y'_n	received bit n in each modified Hamming sub-block

4.2 Notations

For the purposes of this document, the following notations apply:

- (xxxxx)b data bit representation;
- 'XY' hexadecimal notation, equal to XY to the base 16.

5 Protocol activation of PICC Type A

5.1 Activation sequences

The following activation sequence shall be applied.

- PICC activation sequence as defined in ISO/IEC 14443-3 (request, anticollision loop and select).
- The SAK byte shall be checked to get information if the PICC is compliant with ISO/IEC 14443-4. The SAK byte is defined in ISO/IEC 14443-3.
- The PICC may be set to HALT state, using the HLTA command as defined in ISO/IEC 14443-3, if e.g. no ISO/IEC 14443-4 protocol is used at the PCD (the PCD cannot continue the activation sequence in that case).
- If the PICC is compliant to ISO/IEC 14443-4, the RATS may be sent by the PCD as next command after receiving the SAK.
- The PICC shall send its ATS as answer to the RATS. The PICC shall only answer to the RATS if the RATS is received directly after the selection.
- If the PICC supports any changeable parameters in the ATS, a PPS request may be used by the PCD as the next command after receiving the ATS to change parameters.
- The PICC shall send a PPS Response as answer to the PPS request.

The PICC does not need to implement the PPS, if it does not support any changeable parameters in the ATS.

The PCD activation sequence for a PICC Type A is shown in [Figure 1](#).

The RFU handling specified in ISO/IEC 14443-3:2018, 5.3 applies for [Clause 5](#).

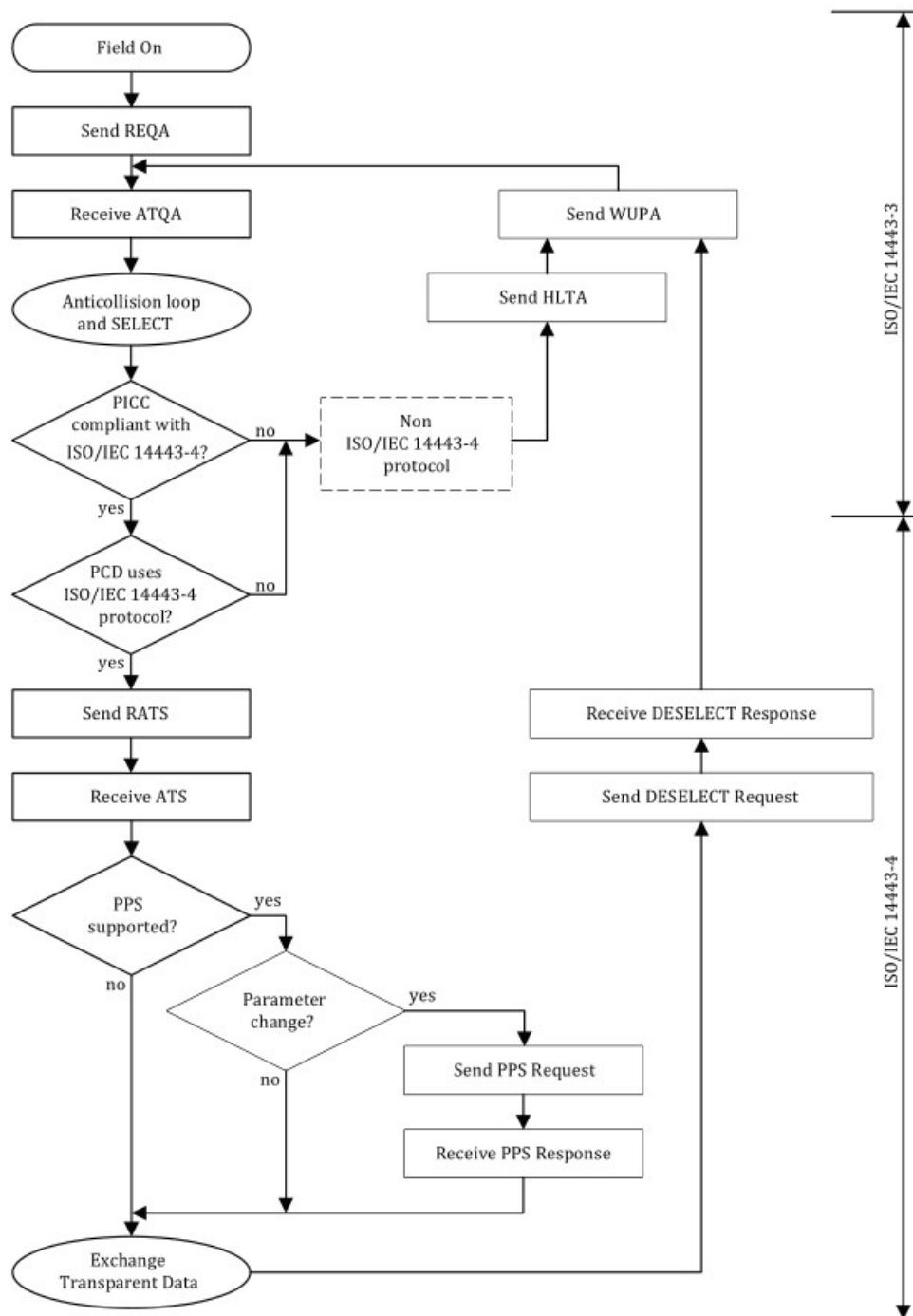
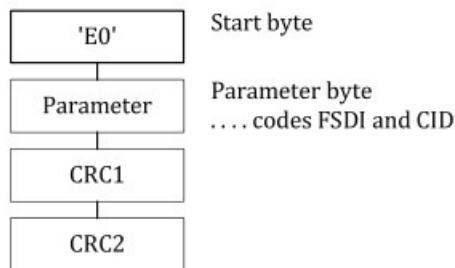


Figure 1 — Activation of a PICC Type A by a PCD

5.2 Request for answer to select

This clause defines the RATS with all its fields (see [Figure 2](#)).

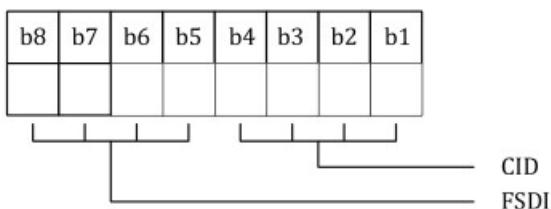
**Figure 2 — Request for answer to select**

The parameter byte consists of two parts (see [Figure 3](#)).

- The most significant half-byte b8 to b5 is called FSDI and codes FSD. The FSD defines the maximum size of a frame the PCD is able to receive. The coding of FSD is given in [Table 1](#).
- Until the RFU values 'D'-'F' are assigned, a PICC receiving an FSDI with a value = 'D'-'F' shall interpret it as FSDI = 'C' (FSD = 4 096 bytes).

NOTE This PCD requirement is added for PCD's compatibility with future PICCs when a revision to this document further defines the behaviour for the RFU values of 'D'-'F'.

- The least significant half byte b4 to b1 is named CID and it defines the logical number of the addressed PICC in the range from 0 to 14. The value 15 is RFU. The CID is specified by the PCD and shall be unique for all PICCs, which are in ACTIVE state at the same time. The CID is fixed for the time the PICC is active and the PICC shall use the CID as its logical identifier, which is contained in the first error-free RATS received.

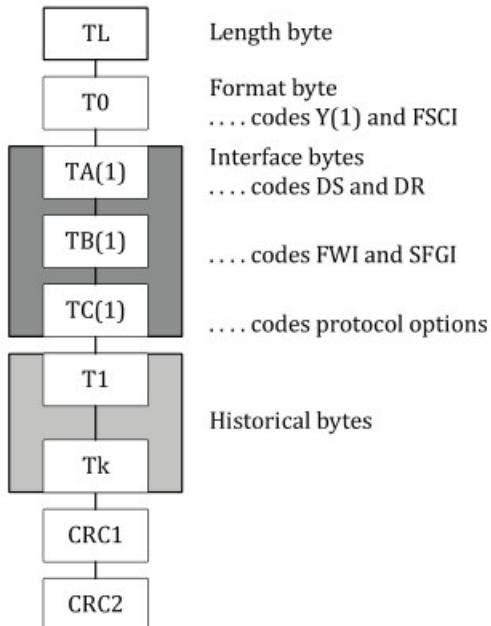
**Figure 3 — Coding of RATS parameter byte****Table 1 — FSDI to FSD conversion**

FSDI	'0'	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'	'9'	'A'	'B'	'C'	'D' - 'F'
FSD (bytes)	16	24	32	40	48	64	96	128	256	512	1 024	2 048	4 096	RFU

5.3 Answer to select

This clause defines the ATS with all its available fields (see [Figure 4](#)).

In the case that one of the defined fields is not present in an ATS sent by the PICC, the default values for that field shall apply.

**Figure 4 — Structure of the ATS**

5.3.1 Structure of the bytes

The length byte TL is followed by a variable number of optional subsequent bytes in the following order:

- format byte T0;
- interface bytes TA(1), TB(1), TC(1);
- historical bytes T1 to Tk.

5.3.2 Length byte

The length byte TL is mandatory and specifies the length of the transmitted ATS including itself. The two CRC bytes are not included in TL. The maximum size of the ATS shall not exceed the indicated FSD. Therefore, the maximum value of TL shall not exceed FSD-2.

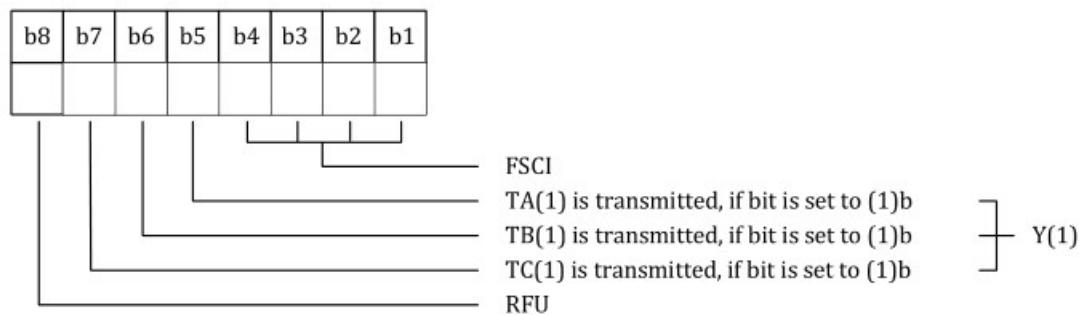
5.3.3 Format byte

The format byte T0 is optional and is present as soon as the length is greater than 1. The ATS can only contain the following optional bytes when this format byte is present.

T0 consists of three parts (see [Figure 5](#)).

- b8 is RFU.
- b7 to b5 contain Y(1) indicating the presence of subsequent interface bytes TC(1), TB(1) and TA(1).
- The least significant half byte b4 to b1 is called FSCI and codes FSC. The FSC defines the maximum size of a frame accepted by the PICC. The default value of FSCI is 2 and leads to a FSC of 32 bytes. The coding of FSC is equal to the coding of FSD (see [Table 1](#)).
- Until the RFU values 'D'-'F' are assigned, a PCD receiving an FSCI with a value = 'D'-'F' shall interpret it as FSCI = 'C' (FSC = 4 096 bytes).

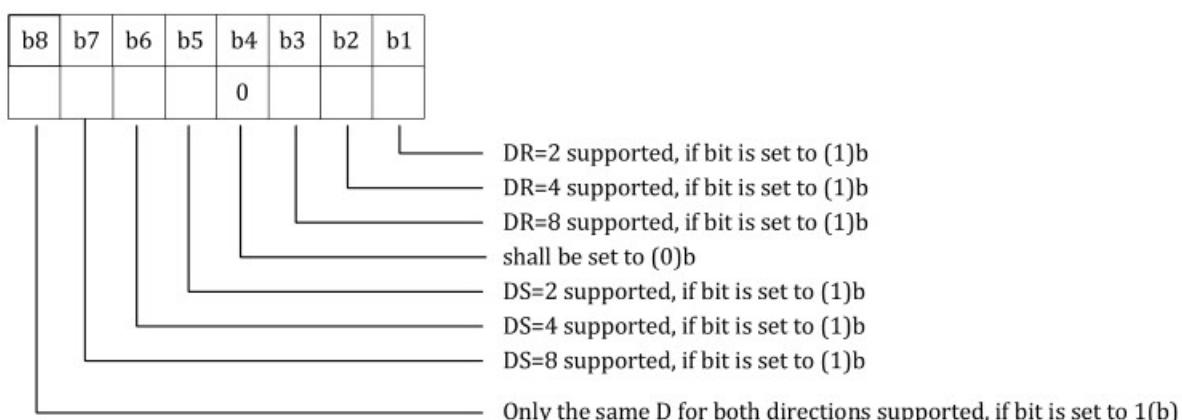
NOTE This PICC requirement is added for PICC's compatibility with future PCDs when a revision to this document further defines the behaviour for the RFU values 'D' – 'F'.

**Figure 5 — Coding of format byte**

5.3.4 Interface byte TA(1)

The interface byte TA(1) consists of four parts (see [Figure 6](#)).

- b8 codes the possibility to handle different divisors for each direction. When this bit is set to 1 the PICC is unable to handle different divisors for each direction.
- b7 to b5 code the bit rate capability of the PICC for the direction from PICC to PCD, called DS. The default value shall be (000)b.
- b4 shall be set to (0)b.
- b3 to b1 code the bit rate capability of the PICC for the direction from PCD to PICC, called DR. The default value shall be (000)b.

**Figure 6 — Coding of interface byte TA(1)**

The selection of a specific divisor D for each direction may be done by the PCD using PPS.

A PCD receiving TA(1) with b4 = (1)b shall interpret it as (b8 to b1) = (00000000)b, implying only ~106 kbit/s supported in both directions. The definition of TA(1) with b4 = (1)b is otherwise undefined.

5.3.5 Interface byte TB(1)

The interface byte TB(1) conveys information to define the frame waiting time and the start-up frame guard time.

The interface byte TB(1) consists of two parts (see [Figure 7](#)).

- The most significant half-byte b8 to b5 is called FWI and codes FWT (see [7.3](#)).

- The least significant half byte b4 to b1 is called SFGI and codes a multiplier value used to define the SFGT. The SFGT defines a specific guard time needed by the PICC before it is ready to receive the next frame after it has sent the ATS. SFGI is coded in the range from 0 to 14. The value of 15 is RFU. The value of 0 indicates no SFGT needed and the values in the range from 1 to 14 are used to calculate the SFGT with the formula given below. The default value of SFGI is 0.

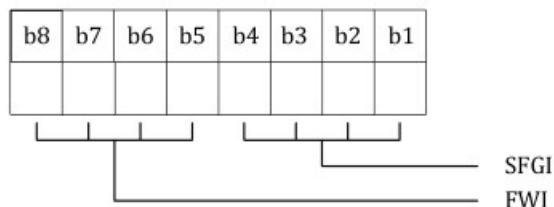


Figure 7 — Coding of interface byte TB(1)

SFGT is calculated by the following formulae:

$$\text{SFGT} = (256 \times 16/fc) \times 2\text{SFGI}$$

SFGT_{MIN} = minimum value of the frame delay time as defined in ISO/IEC 14443-3

$\text{SFGT}_{\text{DEFAULT}}$ = minimum value of the frame delay time as defined in ISO/IEC 14443-3

$$\text{SFGT}_{\text{MAX}} = (256 \times 16/fc) \times 2^{14} (\sim 4\ 949\ \text{ms})$$

Until the RFU value 15 is assigned, a PCD receiving $\text{SFGI} = 15$ shall interpret it as $\text{SFGI} = 0$.

Until the RFU value 15 is assigned, a PCD receiving $\text{FWI} = 15$ shall interpret it as $\text{FWI} = 4$.

5.3.6 Interface byte TC(1)

The interface byte TC(1) specifies a parameter of the protocol.

The specific interface byte TC(1) consists of two parts (see [Figure 8](#)).

- b8 to b3 are each RFU.
- b2 and b1 define which optional fields in the prologue field a PICC does support. The PCD is allowed to skip fields, which are supported by the PICC, but a field not supported by the PICC shall never be transmitted by the PCD. The default value shall be (10)b indicating CID supported and NAD not supported.

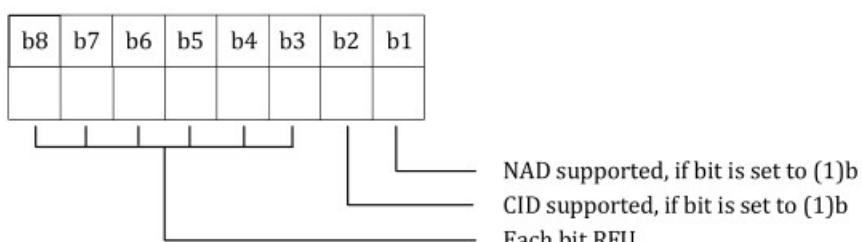


Figure 8 — Coding of interface byte TC(1)

5.3.7 Historical bytes

The historical bytes T1 to Tk are optional and designate general information. The maximum length of the ATS gives the maximum possible number of historical bytes. ISO/IEC 7816-4 specifies the content of the historical bytes.

5.4 Protocol and parameter selection request

PPS request contains the start byte that is followed by two parameter bytes (see [Figure 9](#)).

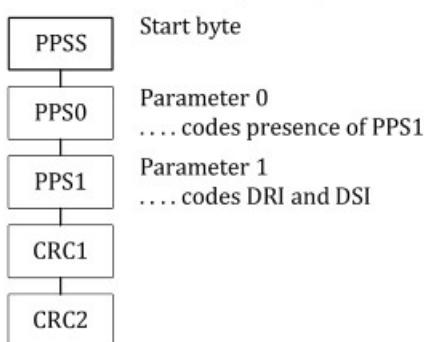


Figure 9 — Protocol and parameter selection request

5.4.1 Start byte

PPSS consists of two parts (see [Figure 10](#)).

- The most significant half byte b8 to b5 shall be set to (1101)_b and identifies the PPS.
- The least significant half byte b4 to b1 is named CID and it defines the logical number of the addressed PICC.

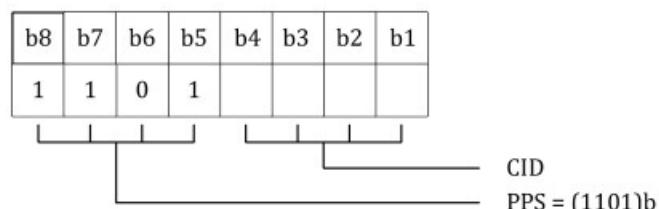
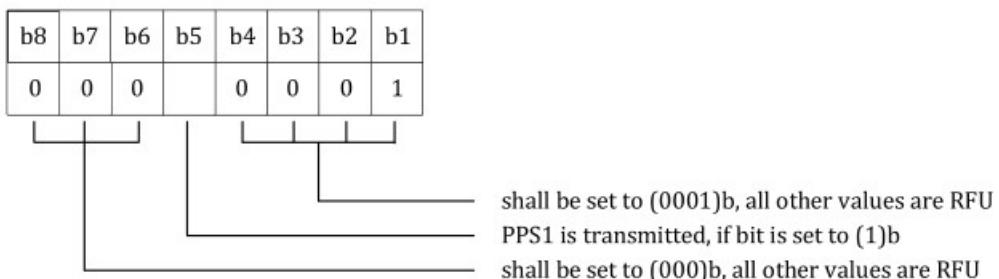


Figure 10 — Coding of PPSS

5.4.2 Parameter 0

PPS0 indicates the presence of the optional byte PPS1 (see [Figure 11](#)).

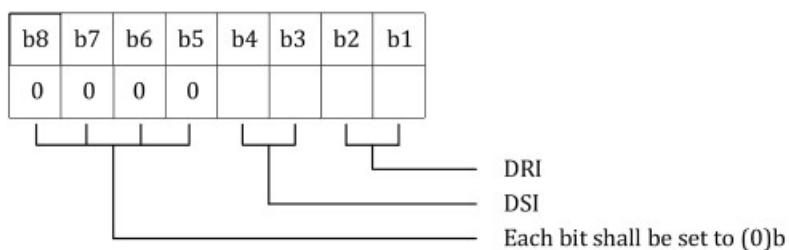
The PCD shall set (b4 to b1) = (0001)_b and (b8 to b6) = (000)_b.

**Figure 11 — Coding of PPS0**

5.4.3 Parameter 1

PPS1 consists of three parts (see [Figure 12](#)).

- b8 to b5 shall each be (0)b; a PICC receiving any bit b8 to b5 set to (1)b shall apply [5.7.2.2 b](#).
- The two-bit value field (b4, b3) is called DSI and codes the selected divisor integer from PICC to PCD.
- The two-bit value field (b2, b1) is called DRI and codes the selected divisor integer from PCD to PICC.

**Figure 12 — Coding of PPS1**

DS and DR are specified in [5.3.4](#).

The coding of D is given in [Table 2](#).

Table 2 — DRI, DSI to D conversion

DRI, DSI	(00)b	(01)b	(10)b	(11)b
D	1	2	4	8

5.5 Protocol and parameter selection response

The PPS response acknowledges the received PPS request (see [Figure 13](#)) and it contains only the start byte (see [5.4.1](#)).

The new bit rates shall become effective in the PICC immediately after it has sent the PPS response. The PCD shall not change the bit rate when the PPS response is missing or invalid or when the PPSS returned by the PICC is not identical with the PPSS sent by the PCD.

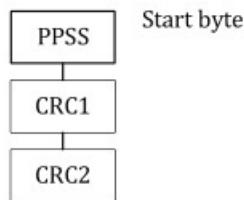


Figure 13 — Protocol and parameter selection response

5.6 Activation frame waiting time

The activation frame waiting time defines the maximum time for a PICC to start sending its response frame after the end of a frame received from the PCD and has a value of $65\ 536/f_c$ ($\sim 4\ 833\ \mu s$).

NOTE The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

5.7 Error detection and recovery

5.7.1 Handling of RATS and ATS

5.7.1.1 PCD rules

When the PCD has sent the RATS and receives a valid ATS the PCD shall continue operation.

In any other case, the PCD may retransmit the RATS before it shall use the deactivation sequence as defined in [Clause 8](#). In case of failure of this deactivation sequence, it may use the HLTA command as defined in ISO/IEC 14443-3.

5.7.1.2 PICC rules

When the PICC has been selected with the last command and

- a) receives a valid RATS, the PICC
 - shall send back its ATS, and
 - shall disable the RATS (stop responding to received RATS);
- b) receives a valid HLTA, the PICC
 - shall process the command and shall enter HALT state;
- c) receives an invalid command, an error or a RATS command with CID = 15, the PICC
 - shall not respond and shall enter IDLE state or HALT state as specified in ISO/IEC 14443-3:2018, Figure 7.

5.7.2 Handling of PPS request and PPS response

5.7.2.1 PCD rules

When the PCD has sent a PPS request and received a valid PPS response, the PCD shall activate the selected parameters and continue operation. In any other case, the PCD may retransmit a PPS request and continue operation.

5.7.2.2 PICC rules

When the PICC has received a RATS, sent its ATS and

- a) received a valid PPS request, the PICC
 - shall send the PPS response,
 - shall disable the PPS request (stop responding to received PPS requests), and
 - shall activate the received parameter;
- b) received an invalid block, the PICC
 - shall disable the PPS request (stop responding to received PPS requests), and
 - shall remain in receive mode;
- c) received a valid block, except a PPS request, the PICC
 - shall disable the PPS request (stop responding to received PPS requests), and
 - shall continue operation.

5.7.3 Handling of the CID during activation

When the PCD has sent a RATS containing a CID = n not equal to 0 and

- received an ATS indicating CID is supported, the PCD
 - shall send blocks containing CID = n to this PICC, and
 - shall not use the CID = n for further RATS while this PICC is in ACTIVE state;
- received an ATS indicating CID is not supported, the PCD
 - shall send blocks containing no CID to this PICC, and
 - shall not activate any other PICC while this PICC is in ACTIVE state.

When the PCD has sent a RATS containing a CID equal to 0 and

- received an ATS indicating CID is supported, the PCD
 - shall send blocks containing either CID equal to 0 or no CID to this PICC, not toggling between these two options until protocol deactivation, and
 - shall not activate any other PICC while this PICC is in ACTIVE state;
- received an ATS indicating CID is not supported, the PCD
 - shall send blocks containing no CID to this PICC, and
 - shall not activate any other PICC while this PICC is in ACTIVE state.

6 Protocol activation of PICC Type B

The activation sequence for a PICC Type B is described in ISO/IEC 14443-3.

7 Half-duplex block transmission protocol

7.1 Elements and mechanisms

The half-duplex block transmission protocol addresses the special needs of contactless card environments and uses the frame format as defined in ISO/IEC 14443-3.

Other relevant elements of the frame format are

- block format;
- maximum frame waiting time;
- power indication; and
- protocol operation.

A mechanism is provided in order to introduce additional protocol functions that may be defined from time to time in this document or in other standards that use this document as their foundation.

This protocol is designed according to the principle layering of the OSI reference model, with particular attention to the minimization of interactions across boundaries. Four layers are defined as follows:

- physical layer exchanges bytes according to ISO/IEC 14443-3;
- data link layer exchanges blocks as defined in this clause;
- session layer combined with the data link layer for a minimum overhead;
- application layer processing commands, which involve the exchange of at least one block or chain of blocks in either direction.

Application selection may be used as defined in ISO/IEC 7816-4. Implicit application selection is not recommended to be used with multi-application PICCs.

The RFU handling specified in ISO/IEC 14443-3:2018, 5.3 applies for [Clauses 7, 8, 9](#) and [10](#).

7.2 Block format

The block format depends on the frame format used for its transmission.

The standard block format as specified in [Figure 14](#) shall be used in standard frames as defined in ISO/IEC 14443-3 and consists of the following:

- a prologue field (mandatory);
- an information field (optional);
- a two-byte epilogue field (mandatory).

The enhanced block format specified in [Figure 15](#) shall be used in frames with error correction as defined in [Clause 10](#) and consists of the following:

- a length field (mandatory);
- a prologue field (mandatory);
- an information field (optional);
- a four-byte epilogue field (mandatory).

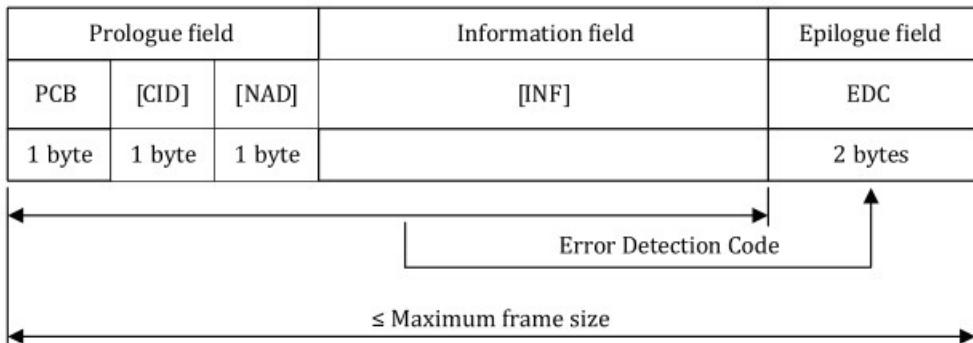


Figure 14 — Standard block format

NOTE 1 The items in brackets indicate optional requirements.

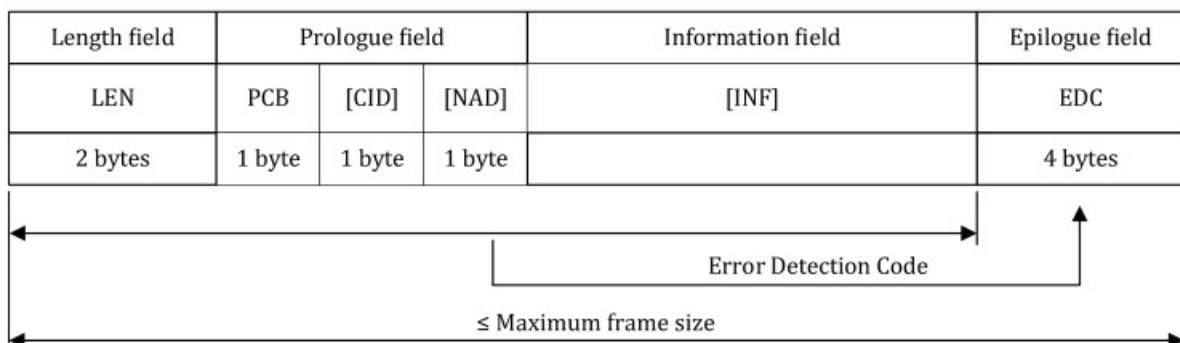


Figure 15 — Enhanced block format

NOTE 2 The items in brackets indicate optional requirements.

7.2.1 Length field

The two-byte length field shall contain the total number of bytes contained in the following fields:

- length field;
- prologue field;
- information field.

Least significant byte is transmitted first, then most significant byte.

7.2.2 Prologue field

The prologue field is mandatory and may be 1, 2, or 3 bytes with PCB mandatory and CID and NAD optional.

7.2.2.1 Protocol control byte field

The PCB is used to convey the information required to control the data transmission.

The protocol defines three fundamental types of blocks.

- I-block used to convey information for use by the application layer.
- R-block used to convey positive or negative acknowledgements. An R-block never contains an INF field. The acknowledgement relates to the last received block.

- S-block used to exchange control information between the PCD and the PICC. The support of the S(PARAMETERS) block is optional for PCDs and PICCs. Three different types of S-blocks are defined.
 - 1) "Waiting time extension" containing a 1 byte long INF field;
 - 2) "DESELECT" containing no INF field;
 - 3) "PARAMETERS" containing a n-byte long INF field with $n \geq 0$.

The coding of the PCB depends on its type and is defined by the following figures. The setting of (b8,b7) is used to identify its block type as defined in [Table 3](#).

A PICC or PCD receiving (b8,b7) = (01)b shall treat it as a protocol error.

Table 3 — Coding of block type

(b8,b7)	Block type
(00)b	I-block
(01)b	RFU
(10)b	R-block
(11)b	S-block

The coding of I-block PCB is shown in [Figure 16](#).

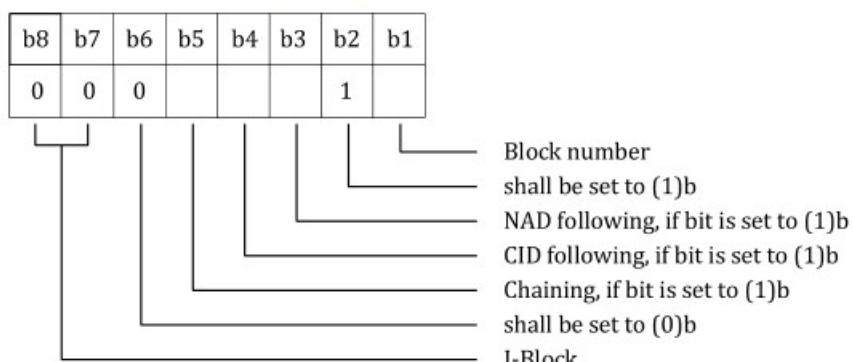


Figure 16 — Coding of I-block PCB

A PICC or PCD receiving an I-Block with b2 = (0)b shall treat it as a protocol error.

A PICC or PCD receiving an I-Block with b6 = (1)b should treat it as a protocol error.

The coding of R-block PCB is shown in [Figure 17](#).

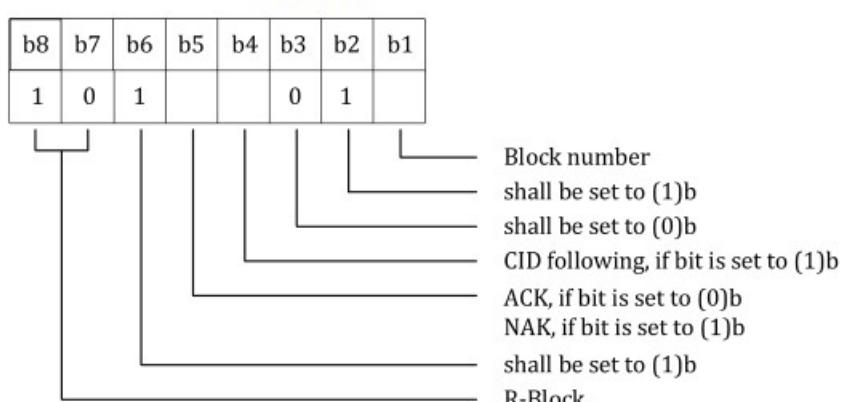


Figure 17 — Coding of R-block PCB

A PICC or PCD receiving an R-Block with $b_6 = (0)b$ or $b_3 = (1)b$ shall treat it as a protocol error.

A PICC or PCD receiving an R-Block with $b_2 = (0)b$ should treat it as a protocol error.

The coding of S-block PCB is shown in [Figure 18](#).

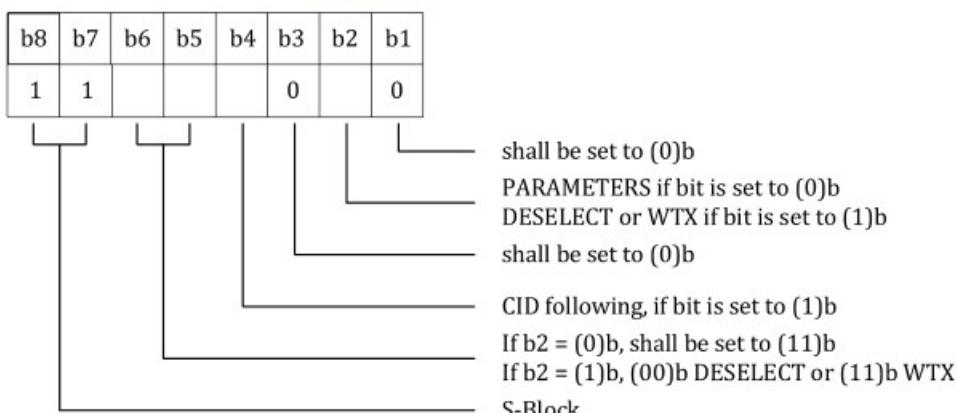


Figure 18 — Coding of S-block PCB

A PICC or PCD receiving S-Block with $b_3 = (1)b$ shall treat it as a protocol error.

A PICC or PCD receiving S-Block with $b_1 = (1)b$ should treat it as a protocol error.

A PICC or PCD receiving S-Block with $b_2 = (0)b$ and $(b_6, b_5) <> (11)b$ shall treat it as a protocol error.

A PICC or PCD receiving S-Block with $b_2 = (1)b$ and $(b_6, b_5) = (01)b$ or $(10)b$ shall treat it as a protocol error.

7.2.2.2 Card identifier field

The CID field is used to identify a specific PICC and consists of three parts (see [Figure 19](#)).

- b_8 and b_7 are used to indicate the power level indication received by a PICC from a PCD. These two bits shall be set to (00)b for PCD to PICC communication. For a definition of power level indication see [7.5](#).
- (b_6, b_5) shall be set to (00)b.
- A PICC or PCD receiving $(b_6, b_5) = (01)b$ or $(10)b$ or $(11)b$ shall treat it as a protocol error.
- b_4 to b_1 code the CID.

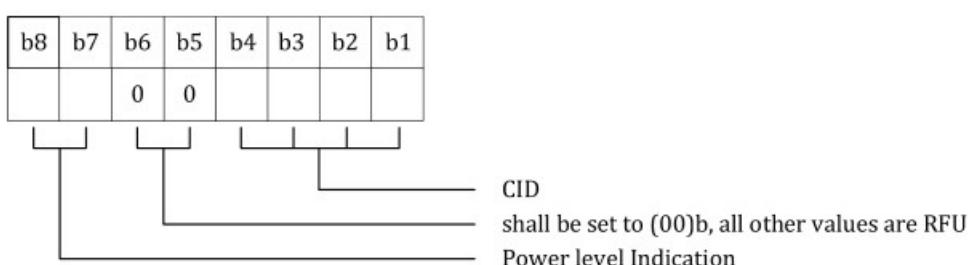


Figure 19 — Coding of card identifier

The coding of CID is given in [5.2](#) for Type A and ISO/IEC 14443-3 for Type B.

The handling of the CID by a PICC is described below:

If the PICC does not support a CID, it shall ignore any block containing a CID.

If the PICC does support a CID, it

- shall respond to blocks containing its CID by using its CID,
- shall ignore blocks containing other CIDs, and
- shall, in case its CID is 0, respond also to blocks containing no CID by using no CID.

7.2.2.3 Node address field

The usage of the NAD shall be in accordance with the definition of NAD in ISO/IEC 7816-3.

The following definitions shall apply for the usage of the NAD:

- a) the NAD field shall only be used for I-blocks;
- b) when the PCD uses the NAD, the PICC shall also use the NAD;
- c) during chaining the NAD shall only be transmitted in the first block of chain;
- d) the PCD shall never use the NAD to address different PICCs (The CID shall be used to address different PICCs);
- e) when the PICC does not support the NAD, it shall ignore any block containing the NAD.

7.2.3 Information field

The INF field is optional. When present, the INF field conveys either application data in I-blocks or non-application data and status information in S-blocks. The length of the information field is calculated by counting the number of bytes of the whole block minus length of prologue and epilogue field.

7.2.4 Epilogue field

The epilogue field contains the EDC of the transmitted block. A transmitted block shall be considered correct if it is received with a valid EDC value.

The EDC of standard blocks shall be the CRC defined in ISO/IEC 14443-3. Type A PICCs shall use CRC_A and Type B PICCs shall use CRC_B in both directions.

The EDC of enhanced blocks shall be CRC_32 as defined below.

The CRC_32 uses polynomial = '04C11DB7' with initial value = 'FFFFFFF' and reflected bit order (LSB first). The final CRC value is bit-inverted before transmission and the least significant byte is transmitted first. Refer to ISO/IEC 13239 for further details. A code sample and an example are given in [Annex E](#).

7.3 Frame waiting time

The FWT defines the time within which the PICC shall start its response frame after the end of a PCD frame (see [Figure 20](#)).

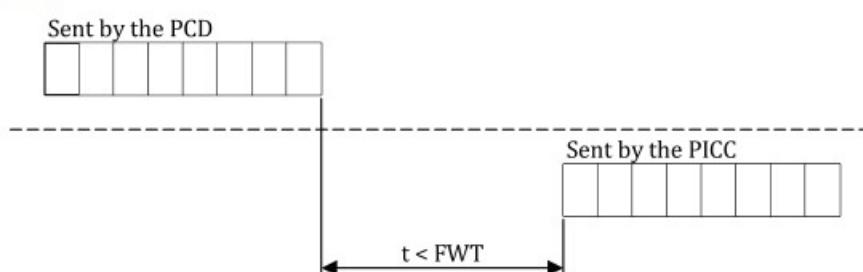


Figure 20 — Frame waiting time

NOTE The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

FWT is calculated by the following formula:

$$\text{FWT} = (256 \times 16 / fc) \times 2^{\text{FWI}}$$

where the value of FWI has the range from 0 to 14 and the value of 15 is RFU.

The default value of FWI is 4 (which gives a FWT value of ~4,8 ms) in the following cases:

- for Type A, if TB(1) is omitted, and
- for S(PARAMETERS) and S(DESELECT) blocks.

For FWI = 0, FWT = FWT_{MIN} (~302 µs).

For FWI = 14, FWT = FWT_{MAX} (~4 949 ms).

The FWT value shall be used by the PCD to detect a protocol error or an unresponsive PICC. The PCD obtains the right to re-transmit if the start of a response from the PICC is not received within FWT.

The FWI field for Type B is located in ATQB as defined in ISO/IEC 14443-3. The FWI field for Type A is located in the ATS (see [5.3.5](#)).

7.4 Frame waiting time extension

When the PICC needs more time than the defined FWT to process the received block, it shall use an S(WTX) request for a waiting time extension. An S(WTX) request contains a 1 byte long INF field that consists of two parts (see [Figure 21](#)).

- b8 and b7 are RFU.
- b6 to b1 code the WTXM. The WTXM is coded in the range from 1 to 59. The values 0 and 60 to 63 are RFU.
- When receiving WTXM = 0 or WTXM = 60-63 the PCD shall treat it as a protocol error.

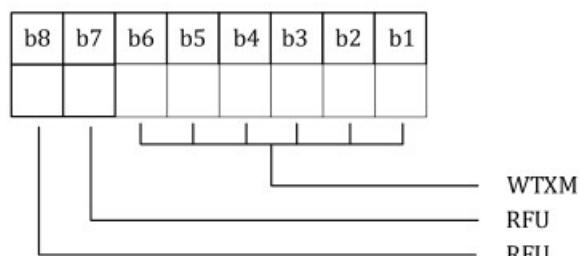
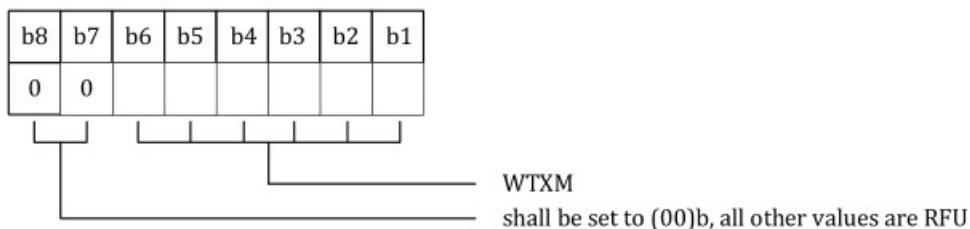


Figure 21 — Coding of INF field of an S(WTX) request

The PCD shall acknowledge by sending an S(WTX) response containing also a 1 byte long INF field that consists of two parts (see [Figure 22](#)) and contains the same WTXM as received in the request.

- b8 and b7 shall be (00)b and all other values are RFU.
- b6 to b1 code the acknowledged WTXM value used to define a temporary FWT.
- If the PICC receives a WTXM value which does not match the WTXM sent by the PICC, the PICC shall treat it as a protocol error.

**Figure 22 — Coding of INF field of an S(WTX) response**

The corresponding temporary value of FWT is calculated by the following formula:

$$FWT_{TEMP} = FWT \times WTXM$$

The time FWT_{TEMP} requested by the PICC, starts after the PCD has sent the S(WTX) response.

FWT_{MAX} shall be used, when the formula results in a value higher than FWT_{MAX} .

The temporary FWT applies only until the next block has been received by the PCD.

7.5 Power level indication

The power level indication is coded as shown in [Table 4](#) using two bits embedded in the CID field (when present) sent by the PICC (see [7.2.2.2](#)).

Table 4 — Coding of power level indication

(00)b	PICC does not support the power level indication
(01)b	Insufficient power for full functionality
(10)b	Sufficient power for full functionality
(11)b	More than sufficient power for full functionality

NOTE Interpretation of the power level indication by the PCD is optional.

7.6 Protocol operation

After the activation sequence the PICC shall wait for a block as only the PCD has the right to send. After sending a block, the PCD shall switch to receive mode and wait for a block before switching back to transmit mode. The PICC may transmit blocks only in response to received blocks (it is insensitive to time delays). After responding, the PICC shall return to the receive mode.

The PCD shall not initiate a new pair of command/response until the current pair of command/response has been completed or if the frame waiting time is exceeded with no response.

7.6.1 S(PARAMETERS) blocks

After the activation sequence, the PCD may send at any time a first S(PARAMETERS) block with or without INF field to check if S(PARAMETERS) blocks are supported by the PICC.

The PCD may initiate an exchange of S(PARAMETERS) at any time also during PCD chaining or PICC chaining or error handling, before it continues block exchange with the new parameters settings (if modified by the PCD and acknowledged by the PICC).

NOTE 1 Typically this is used to change air interface parameters for better communication stability.

The PCD should determine whether the PICC supports S(PARAMETERS) prior to using S(PARAMETERS) during chaining.

The content of the S(PARAMETERS) INF field is defined in this document and shall comply with the BER-TLV encoding rules as per ISO/IEC 7816-4:2013, Annex E. The tag field indicates context-specific class, the length field shall be short form encoded.

For each S(PARAMETERS) block the PCD and PICC shall not:

- use the same tag more than once,
- activate the same parameter with different tags using different values (e.g. "Bit rates Activation" + "Selected framing options from PICC to PCD" and "Frame Format Activation" + "Selected framing options from PICC to PCD"),

otherwise, the behavior of the receiver is indeterminate.

On reception of an S(PARAMETERS) block containing at least one unknown tag, not supported or wrong parameter, or if any activation combined with previous activations results in an invalid context, the PICC supporting S(PARAMETERS) should ignore all tags and parameters in this S(PARAMETERS) block. The PICC supporting S(PARAMETERS) should not change its current state and protocol parameters, and should respond with an S(PARAMETERS) block which may contain supported indication tags and/or S(PARAMETERS) error.

NOTE 2 All of these recommendations will become mandatory requirements in the next edition of this document.

Both the PCD and the PICC shall send only relevant BER-TLV objects. If the INF field is present, it shall contain the S(PARAMETERS) block information tag, either empty (i.e. 'A0 00'), or with nested function tags. It is also valid to send an empty S(PARAMETERS) block (i.e. no INF field and consequently no tags present).

PCDs and PICCs supporting S(PARAMETERS) shall accept frames of at least 48 bytes in length.

The PICC and PCD maximum frame sizes need to be large enough to contain the largest possible S(PARAMETERS) block; the required minimum sizes may be increased in the future when new S(PARAMETERS) options are added.

On reception of an S(PARAMETERS) block without INF field or with INF field containing only 'A0 00', a PICC supporting S(PARAMETERS) shall respond with an S(PARAMETERS) block which may contain supported indication tags.

If the PICC does not support S(PARAMETERS), it shall stay mute on reception of any S(PARAMETERS) block.

S(PARAMETERS) blocks are used to negotiate communication parameters in PROTOCOL state. The information field shall contain tags and values as defined in [Table 5](#).

The following rules shall be applied to negotiate those parameters.

- The PCD shall send an S(PARAMETERS) block to request parameters.
- If the PICC supports S(PARAMETERS) blocks, the PICC shall respond with an S(PARAMETERS) block containing values for all supported parameters. If the same communication parameters can be requested via different tags, the PICC shall always respond with the same indication independent of which tag is used.

After the PICC has sent its response and has indicated its parameters, the PCD may activate the desired options for each communication direction with the following rules.

- The PCD shall send an S(PARAMETERS) block to activate selected parameters.
- The PICC shall acknowledge the activated parameters with an S(PARAMETERS) block and then shall activate the negotiated parameters.
- The PCD shall activate the negotiated parameters.

Table 5 — S(PARAMETERS) tag definition

Tag (Hex)	Description	Length (Hex)	Value
'A0'	S(PARAMETERS) block information	L	Not present or nested function tags as specified by Tables 6 and 7 , in any order
'BE'	S(PARAMETERS) error	'01'	'00': no further error indication

7.6.2 Multi-Activation

The Multi-Activation feature allows the PCD to hold several PICCs in ACTIVE state simultaneously. It allows switching directly between several PICCs without needing additional time for deactivation of a PICC and activation of another PICC.

For an example of Multi-Activation, see [Annex A](#).

The PCD needs to handle a separate block number for each activated PICC.

7.6.3 Chaining

The chaining feature allows the PCD or PICC to transmit information that does not fit in a single block, as defined by FSC or FSD respectively, by dividing the information into several blocks. Each of those blocks shall have a length less than or equal to FSC or FSD respectively.

The chaining bit in the PCB of an I-block controls the chaining of blocks. Each I-block with the chaining bit set shall be acknowledged by an R-block.

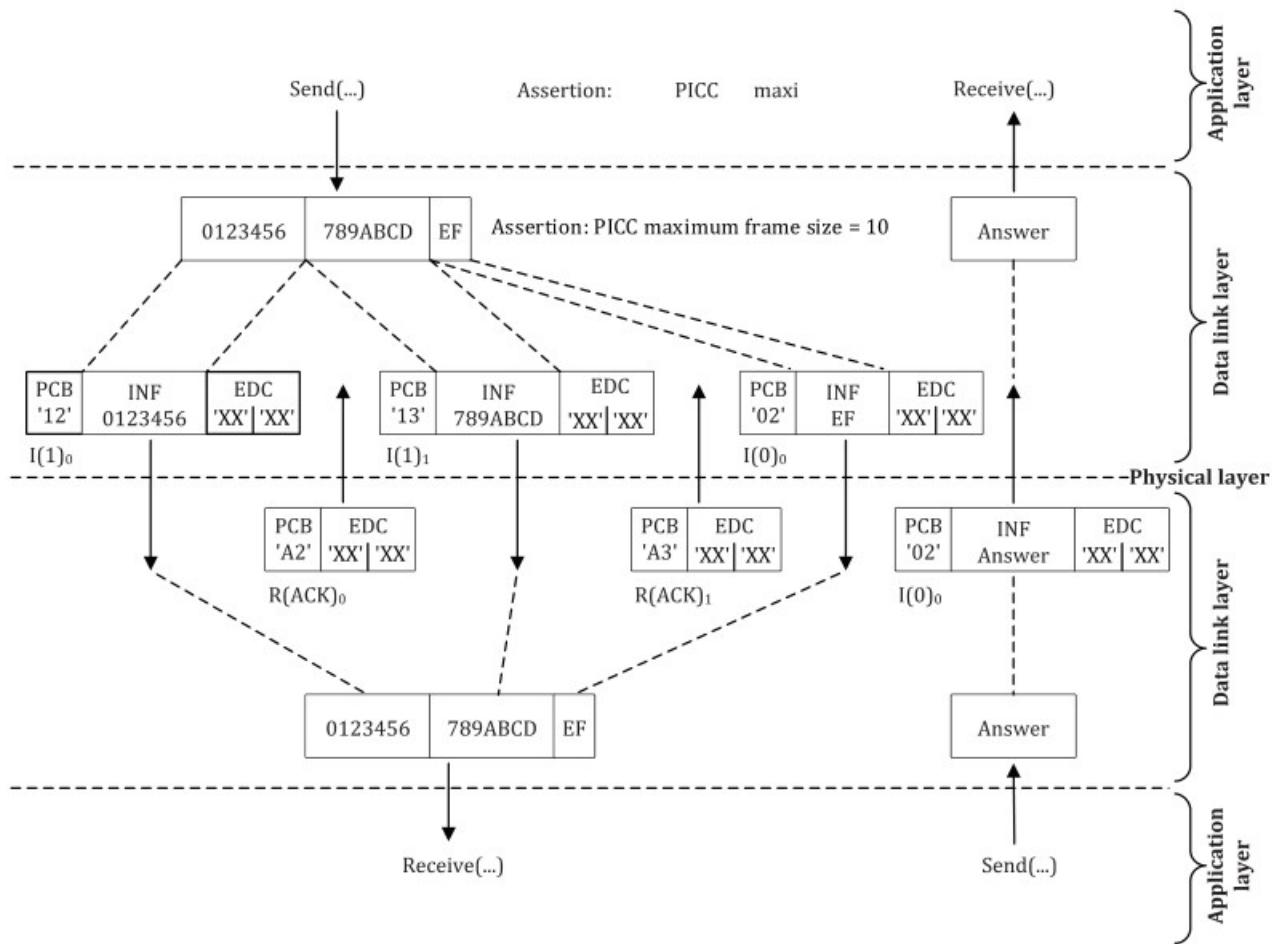
The chaining feature is shown in [Figure 23](#), using a 16 byte long string transmitted in three blocks.

Notation:

I(1)x I-block with chaining bit set and block number x;

I(0)x I-block with chaining bit not set (last block of chain) and block number x;

R(ACK)x R-block that indicates a positive acknowledge.



NOTE This example does not use the optional fields NAD and CID.

Figure 23 — Chaining

7.6.4 Block numbering rules

7.6.4.1 PCD rules

- Rule A. The PCD block number shall be initialized to 0 for each activated PICC.
- Rule B. When an I-block or an R(ACK) block with a block number equal to the current block number is received, the PCD shall toggle the current block number for that PICC before optionally sending a block.

7.6.4.2 PICC rules

- Rule C. The PICC block number shall be initialized to 1 at activation.
- Rule D. When an I-block is received, the PICC shall toggle its block number before sending a block.

NOTE 1 The PICC can check if the received block number is not in compliance with PCD rules to decide neither to toggle its internal block number nor to send a response block.

Rule E. When an R(ACK) block with a block number not equal to the current PICC's block number is received, the PICC shall toggle its block number before sending a block.

NOTE 2 There is no block number toggling when an R(NAK) block is received.

7.6.5 Block handling rules

7.6.5.1 General rules

- Rule 1. The first block shall be sent by the PCD.
- Rule 2. When an I-block indicating chaining is received, the block shall be acknowledged by an R(ACK) block.
- Rule 3. S-blocks are only used in pairs. An S(..) request block shall always be followed by an S(..) response block (see [7.4](#) and [Clause 8](#)).

7.6.5.2 PCD rules

- Rule 4. When an invalid block is received or a FWT time-out occurs, an R(NAK) block shall be sent [except in the case of PICC chaining or S(DESELECT) or S(PARAMETERS)].
- Rule 5. In the case of PICC chaining, when an invalid block is received or a FWT time-out occurs, an R(ACK) block shall be sent.

NOTE 1 An R(ACK) block can be sent by the PCD only in case of PICC chaining, as the PICC response when receiving an R(ACK) block in other cases is not defined.

Rule 6. When an R(ACK) block is received, if its block number is not equal to the PCD's current block number, the last I-block shall be re-transmitted.

NOTE 2 The last I-block re-transmission is not required out of PCD chaining. The PCD can determine the presence of a PICC by sending R(NAK) blocks at any time out of chaining (including before sending any I-block) and receiving R(ACK) from the PICC if present.

- Rule 7. When an R(ACK) block is received, if its block number is equal to the PCD's current block number, chaining shall be continued.
- Rule 8. If the S(DESELECT)/S(PARAMETERS) request is not answered by an error-free S(DESELECT)/S(PARAMETERS) response the S(DESELECT)/S(PARAMETERS) request may be re-transmitted.

In case of not receiving an S(DESELECT) response after an S(DESELECT) request, the PICC may be ignored.

7.6.5.3 PICC rules

- Rule 9. The PICC is allowed to send an S(WTX) block instead of an I-block or an R(ACK) block.
- Rule 10. When an I-block not indicating chaining is received, the block shall be acknowledged by an I-block.

NOTE If the I-block received is empty then the mandatory I-block sent can either be empty or contain any applicative information (e.g. error code).

- Rule 11. When an R(ACK) or an R(NAK) block is received, if its block number is equal to the PICC's current block number, the last block shall be re-transmitted.
- Rule 12. When an R(NAK) block is received, if its block number is not equal to the PICC's current block number, an R(ACK) block shall be sent.
- Rule 13. When an R(ACK) block is received, if its block number is not equal to the PICC's current block number, and the PICC is in chaining, chaining shall be continued.

7.6.6 PICC presence check

The following methods may be used to check the presence of a PICC at any time including before any I-block exchange.

The PCD shall not check the presence of a PICC until the current pair of command/response has been completed or when the frame waiting time is exceeded with no response.

7.6.6.1 Method 1

The PCD may send an empty I-block and expect to receive an I-block from the PICC.

7.6.6.2 Method 2

Before the first I-block exchange, the PCD may send an R(NAK) block (with block number 0) and expect to receive an R(ACK) (with block number 1) block from the PICC (rule 12).

After the first I-block exchange, the PCD may either

- send an R(NAK) block (with current block number) and expect to receive an R(ACK) block from the PICC (rule 12), in which case the PCD should not retransmit its last I-block as mentioned in the note in rule 6, or
- toggle its block number then send an R(NAK) block and expect to receive the last I-block from the PICC (rule 11).

7.6.7 Error detection and recovery

When at least one error is detected (after the optional error recovery mechanism, see [10.4.7](#)) the following recovery rules shall be applied. The definitions made in this clause overrule the block handling rules (see [7.6.5](#)).

7.6.7.1 Errors detected by the PCD

The following errors shall be detected by the PCD.

- Transmission error (Frame error or EDC error) or FWT time-out

The PCD shall attempt error recovery by the following techniques in the order shown:

- application of PCD rules (see [7.6.5.2](#));
- optionally apply PCD rules (see [7.6.5.2](#)) once more;
- use of S(DESELECT) request;
- optionally use of S(DESELECT) request once more (as specified in [8.2](#));
- ignore the PICC.

- b) Protocol error (infringement of PCB coding or infringement of protocol rules)

The PCD shall attempt error recovery by the following techniques in the order shown:

- use of S(DESELECT) request;
- ignore the PICC.

7.6.7.2 Errors detected by the PICC

The following errors shall be detected by the PICC:

- a) transmission error (Frame error or EDC error);
- b) protocol error (infringement of PCB coding or infringement of the protocol rules).

The PICC shall attempt no error recovery. The PICC shall always return to receive mode, when a transmission error or a protocol error occurs and it shall accept an S(DESELECT) request at any time.

NOTE An R(NAK) block is never sent by the PICC.

8 Protocol deactivation of PICC Type A and Type B

The PICC shall be set to HALT state, after the transactions between PCD and PICC have been completed.

The deactivation of a PICC is done by using a DESELECT Command.

The DESELECT command is coded as an S-block of the protocol and consists of an S(DESELECT) request block sent by the PCD and an S(DESELECT) response sent as acknowledge by the PICC.

8.1 Deactivation frame waiting time

The deactivation frame waiting time defines the maximum time for a PICC to start sending its S(DESELECT) response frame after the end of the S(DESELECT) request frame received from the PCD and has a value of 65 536/fc ($\sim 4,8$ ms).

NOTE The minimum time between frames in any direction is defined in ISO/IEC 14443-3.

8.2 Error detection and recovery

When the PCD has sent an S(DESELECT) request and has received an S(DESELECT) response, the PICC has been set successfully to HALT state and the CID assigned to it is released.

When the PCD fails to receive an S(DESELECT) response the PCD may retry the deactivation sequence.

9 Activation of bit rates and framing options in PROTOCOL state

S(PARAMETERS) blocks shall be used to negotiate the used bit rates and communication parameters when the PICC is in PROTOCOL state. The information field shall contain tags and values as defined in [Tables 5](#) and [6](#).

Table 6 — Bit rates function tags identifier definition

Tag (Hex)	Description	Length (Hex)	Value		
'A1'	Bit rates request (sent by the PCD)	'0'	—		
'A2'	Bit rates indication (sent by the PICC in response to tag 'A1')	L	Tag (Hex)	Length (Hex)	Value
			'80'	'02'	Supported bit rates from PCD to PICC 1 st byte is specified in Figure 24 2 nd byte is specified in Figure 25
			'81'	'02'	Supported bit rates from PICC to PCD 1 st byte is specified in Figure 24 2 nd byte set to '00', other values are RFU
			'82'	'01'	Supported framing options from PICC to PCD (see Figure 26) ^a
'A3'	Bit rates activation (sent by the PCD)	L	Tag (Hex)	Length (Hex)	Value
			'83'	'02'	Selected bit rate from PCD to PICC ^b 1 st byte is specified in Figure 24 2 nd byte is specified in Figure 25
			'84'	'02'	Selected bit rate from PICC to PCD ^b 1 st byte is specified in Figure 24 2 nd byte set to '00', other values are RFU
			'85'	'01'	Selected framing options from PICC to PCD (see Figure 26) ^{a,c,d}
'A4'	Bit rates acknowledgement (sent by the PICC in response to tag 'A3')	'0'	—		

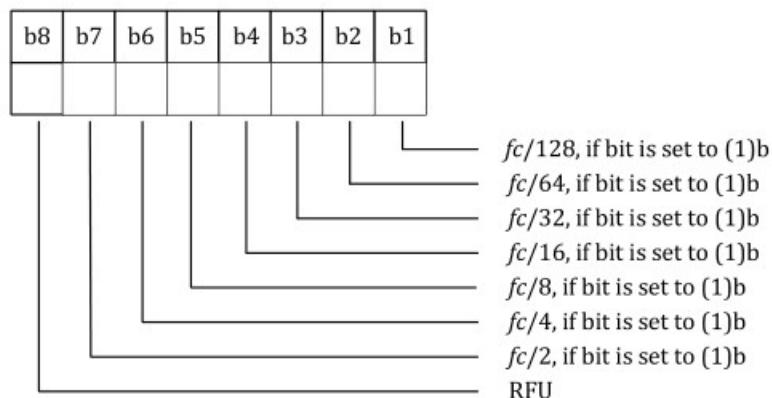
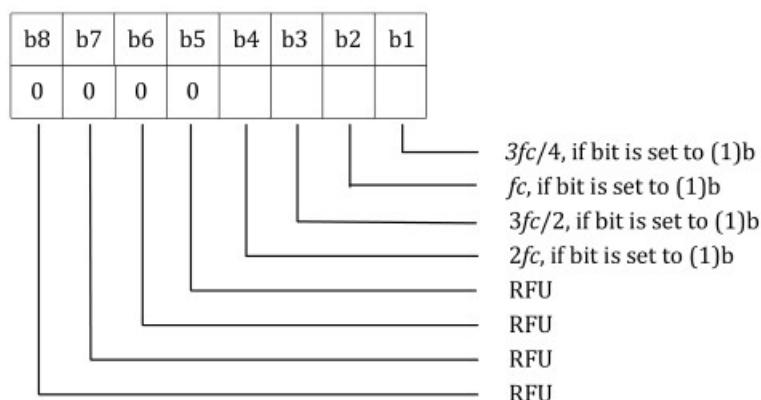
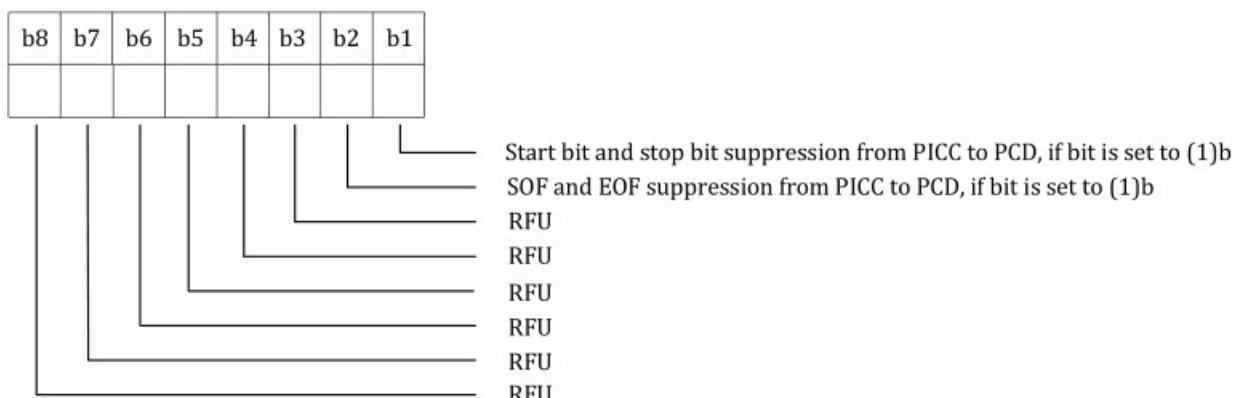
^a Shall be omitted for Type A PICCs.

^b The PCD shall set only one bit. The PCD shall not activate simultaneously a bit rate higher than $fc/16$ for PCD to PICC communication and a bit rate of $fc/128$ for PICC to PCD communication in Type A.

^c The PCD shall not set both b1 (start bit and stop bit suppression) and b2 (SOF and EOF suppression). When the PCD sets b1 (start bit and stop bit suppression):

- The PICC shall use a SOF low time of 10 etu and a SOF high time of 2 etu.
- The PICC shall use an EOF low time of 10 etu.
- The PICC shall apply no character separation.

^d Any setting overrules the ATTRIB indications (see ISO/IEC 14443-3:2018, 7.10.3.3.)

**Figure 24 — Coding of bit rates up to $fc/2$** **Figure 25 — Coding of bit rates of $3fc/4$, fc , $3fc/2$ and $2fc$ from PCD to PICC****Figure 26 — Framing options from PICC to PCD**

As an example, the sequence for an activation of the bit rate

- $fc/8$ from PCD to PICC, and
 - $fc/2$ from PICC to PCD
- with a PICC indicating to support
- bit rates $fc/128$, $fc/16$ and $fc/8$ for PCD to PICC communication,
 - bit rates $fc/128$, $fc/16$ and $fc/2$ for PICC to PCD communication, and

- no framing options,

is illustrated in [Figure 27](#).

Step	PCD	PICC
1	S(PARAMETERS)('A0 02 A1 00' CRC)	→
2		← S(PARAMETERS) (‘A0 0A’ ‘A2 08’ ‘80 02 19 00’ ‘81 02 49 00’ CRC)
3	S(PARAMETERS) (‘A0 0A’ ‘A3 08’ ‘83 02 10 00’ ‘84 02 40 00’ CRC)	→
4		← S(PARAMETERS)('A0 02 A4 00' CRC)

Figure 27 — Bit rates activation example

10 Frame with error correction

10.1 General

Frames with error correction as specified in [10.2](#) and [10.3](#) shall be used after their activation as specified in 11.5. An example is given in [Annex F](#).

10.2 Type A PCD frame format for bit rates up to $fc/16$ and higher than $fc/2$ and Type A PICC frame format for all bit rates

Frames with error correction, as defined in [Figure 28](#), shall be used for data exchange and consist of, in the following order:

- start of communication;
- SYNC;
- enhanced block with error correction (see [10.4](#));
- end of communication.

SYNC consists of six dedicated bytes with the values ‘55’, ‘55’, ‘74’, ‘74’, ‘74’ and ‘74’ transmitted in this order.

SYNC and enhanced blocks with error correction shall be transmitted as bytes consisting of 8 bits.

NOTE Parity bits (see ISO/IEC 14443-3:2018, 6.2.3.2) are not used.



Figure 28 — Frame with error correction

10.3 Type A PCD frame format for bit rates of $fc/8$, $fc/4$ and $fc/2$ and Type B PCD and PICC frame format for all bit rates

Frames with error correction, as defined in [Figure 29](#), shall be used for data exchange and consist of, in the following order:

- SOF as defined in ISO/IEC 14443-3:2018, 7.1.4;
- SYNC as defined in [10.2](#), transmitted as characters as defined in ISO/IEC 14443-3:2018, 7.1.1;
- enhanced block with error correction (see [10.4](#)) transmitted as characters as defined in ISO/IEC 14443-3:2018, 7.1.1;
- EOF as defined in ISO/IEC 14443-3:2018, 7.1.5.

No character separation shall be applied in frames with error correction.

SOF, EOF, start bit, stop bit and SYNC may be suppressed in accordance with [10.5](#).

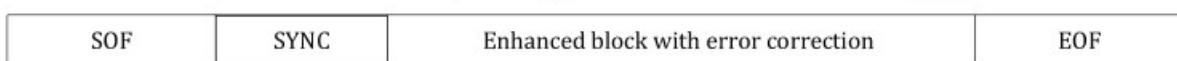


Figure 29 — Frame with error correction

10.4 Enhanced block with error correction

10.4.1 General

Enhanced block with error correction shall be composed of one or several 8-byte modified Hamming sub-blocks, each of them being calculated from 7-byte sub-blocks from enhanced block (see [Figure 30](#)).

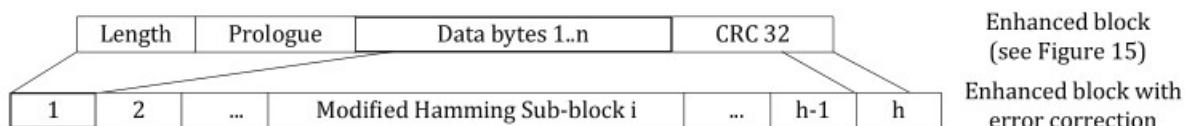


Figure 30 — Enhanced block with error correction

10.4.2 Modified Hamming sub-block format

Each modified Hamming sub-block shall consist of 7 bytes from enhanced block, followed by one Hamming control byte used to correct one single-bit error on the Hamming sub-block.

Modified Hamming sub-blocks shall always be complete. If necessary, 'FF' bytes shall be added to complete the last bytes from enhanced block to get 7 bytes.

10.4.3 Hamming control byte

The Hamming control byte shall contain the Hamming control bits c_n and logical "1" padding bits in the following order:

- one logical "1" padding bit;
- six Hamming control bits c_n in the order $c_1, c_2, c_3, c_4, c_5, c_6$;
- one logical "1" padding bit.

An example for Hamming control byte calculation in ANSI C language is given in [E.2](#).

10.4.4 Hamming control generation matrix A

Hamming control bits generation matrix A (see [Figure 33](#)) shall be generated by following steps:

- generate Matrix H' (see [Figure 32](#)) using the formula in [Figure 31](#);
- remove column vectors \underline{h}'_n , with $n = 1, 2, 4, 8, 16$ and 32 , of H' .

$$h'_{m,n} = \begin{cases} 1 & \text{for } (n \wedge 2^{m-1}) \neq 0 \\ 0 & \text{otherwise} \end{cases} \text{ with } 1 \leq m \leq 6 \text{ and } 1 \leq n \leq 62$$

Figure 31 — Matrix H' generation

NOTE \wedge stands for a bitwise AND operation.

$$H' = \begin{pmatrix} 1 & 0 & 1 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix}$$

Figure 32 — Matrix H'

$$A = \begin{pmatrix} 1 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix}$$

Figure 33 — Hamming control generation matrix A

10.4.5 Hamming control bits calculation

Hamming control bits c_m ($m = 1..6$) shall be calculated over data d_n ($n = 1..56$) using the formula in [Figure 34](#). d_1 is bit b1 of the first byte and d_{56} is bit b8 of the seventh byte of any 7-byte sub-block from enhanced block.

$$\underline{c} = A \times \underline{d}$$

Figure 34 — Hamming control bits generation

10.4.6 Hamming control check matrix H

The Hamming control check matrix H (illustrated in [Figure 35](#)) is a concatenation of matrix A and matrix $I_{6 \times 6}$.

$$H = A | I_{6,6} = \begin{pmatrix} 1 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 1 & 1 & & 0 & 0 & 1 & 0 & 1 & & 0 & 0 \\ 0 & 1 & 1 & 1 & & 1 & 1 & 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 0 & 0 & & 0 & 0 \\ 0 & 0 & 0 & 0 & & 1 & 1 & 1 & 0 & 0 & & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

Figure 35 — Hamming control check matrix H

10.4.7 Error correction

Hamming control bits shall be used to detect and correct any single bit error in modified Hamming sub-blocks.

The so called syndrome \underline{s} shall be calculated using formula in [Figure 36](#). To get \underline{y} from \underline{y}' , the padding bits of the received data y'_n on position 57 and 64 shall be removed. y'_1 is bit b1 of the first received byte and y'_{64} is bit b8 of the eighth received byte of any 8-byte sub-block from enhanced block with error correction.

$$\underline{s} = H \times \underline{y}$$

Figure 36 — Syndrome calculation

The numerical interpretation s' of the syndrome \underline{s} shall be used for error correction:

- if $s' = 0, 1, 2, 4, 8, 16, 32$ or 63 no change in received bits y'_1 to y'_{56} ;
- else
 - calculate error position s by reducing s' by the amount of powers of 2 ($1, 2, 4, 8, 16, 32$) which are smaller than s' ;
 - invert the received bit y'_s .

NOTE More than one bit error cannot be corrected by this method. EDC will detect these multiple errors with very high probability.

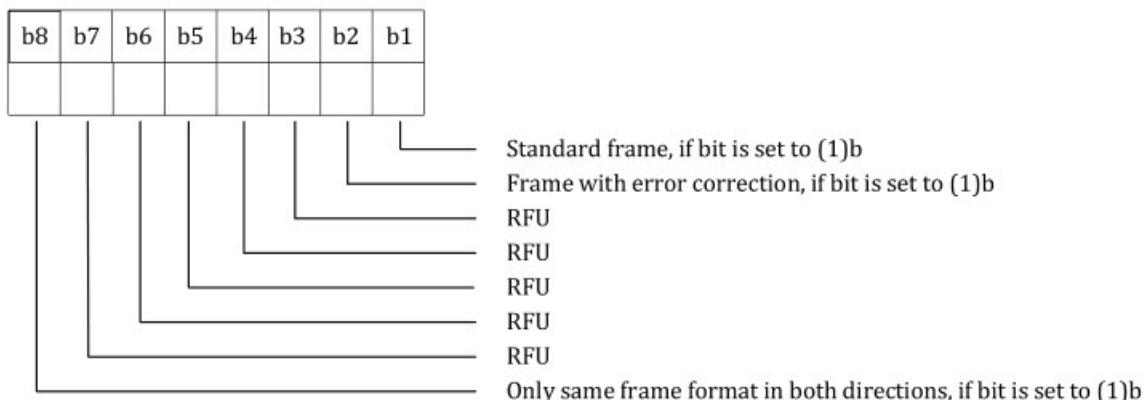
10.5 Activation of frame with error correction in PROTOCOL state

S(PARAMETERS) blocks shall be used to negotiate the used frame and communication parameters when the PICC is in PROTOCOL state. The information field shall contain tags and values as defined in [Table 5](#) and [Table 7](#).

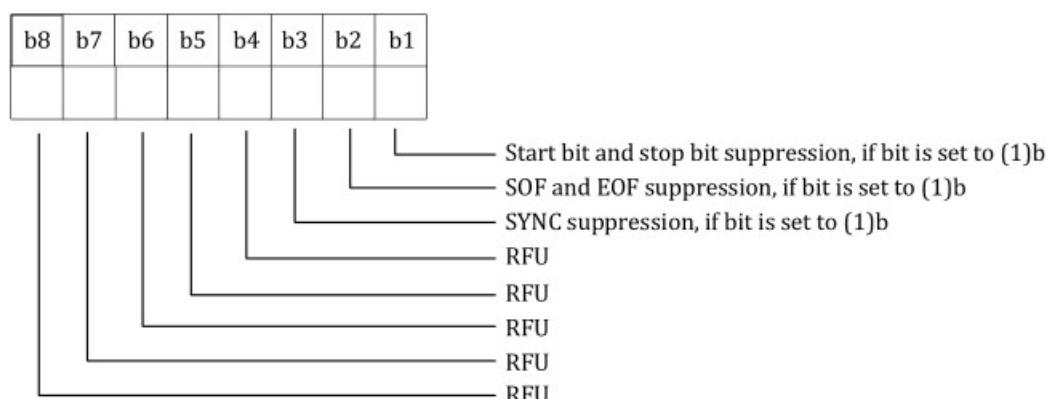
Table 7 — Frame format function tags identifier definition

Tags (Hex)	Description	Length (Hex)	Value		
'A5'	Frame format request (sent by the PCD)	0	—		
'A6'	Frame format indication (sent by the PICC in response to tag 'A5')	L	Tags (Hex)	Length (Hex)	Value
			'80'	'01'	Supported frame formats from PCD to PICC (see Figure 37) ^a
			'81'	'01'	Supported frame formats from PICC to PCD (see Figure 37) ^a
			'82'	'01'	Supported framing options from PCD to PICC (see Figure 38) ^b
			'83'	'01'	Supported framing options from PICC to PCD (see Figure 38) ^c
'A7'	Frame format activation (sent by the PCD)	L	Tags (Hex)	Length (Hex)	Value
			'84'	'01'	Selected frame formats from PCD to PICC (see Figure 37) ^d
			'85'	'01'	Selected frame formats from PICC to PCD (see Figure 37) ^d
			'86'	'01'	Selected framing options from PCD to PICC (see Figure 38) ^{b,e,f}
			'87'	'01'	Selected framing options from PICC to PCD (see Figure 38) ^{c,e,g,h}
'A8'	Frame format acknowledgement (sent by the PICC in response to tag 'A7')	0	—		

^a b8 shall be set to the same value in both communication directions.
^b Shall be omitted for Type A PICCs not supporting bit rates from PCD to PICC of $fc/8$, $fc/4$ and $fc/2$.
^c Shall be omitted for Type A PICCs.
^d b8 shall be set to (0)b and only one bit out of b1 and b2 shall be set to (1)b.
^e When SYNC is suppressed the PCD shall not select both start bit and stop bit suppression and SOF and EOF suppression. If start bit and stop bit suppression is selected, SOF and EOF low time of 10 etu and SOF high time of 2 etu shall be used.
^f The PCD shall not activate framing options for PCD to PICC standard frames.
^g When tag '87' is used, the corresponding tag of bit rates $> fc/16$ shall not be used and vice versa.
^h Any setting overrules the ATTRIB indications (see ISO/IEC 14443-3:2018, 7.10.3.3).

**Figure 37 — Frame formats**

Standard frame support is mandatory.

**Figure 38 — Framing options**

NOTE 1 Framing options depend on PICC Type, frame format, bit rate and communication direction. An overview is given in [Annex G](#).

As an example the sequence for an activation of frame with error correction in both directions with

- no SYNC suppression in both directions,
- no SOF and EOF suppression in both directions,
- start bit and stop bit suppression in both directions, and

with a PICC indicating to support

- standard frame and frame with error correction in both directions independent of each direction,
- SYNC suppression in both directions,
- SOF and EOF suppression in both directions, and
- start bit and stop bit suppression in both directions is illustrated in [Figure 39](#).

An overview on allowed suppressions of framing options is given in [Annex G](#).

Step	PCD	PICC
1	S(PARAMETERS)('A0 02 A5 00' CRC)	→
2		S(PARAMETERS) ('A0 0E' 'A6 0C' '80 01 03' '81 01 03' '82 01 07' '83 01 07' CRC)
3	S(PARAMETERS) ('A0 0E' 'A7 0C' '84 01 02' '85 01 02' '86 01 01' '87 01 01' CRC)	→
4		← S(PARAMETERS)('A0 02 A8 00' CRC)

Figure 39 — Frame activation example

NOTE 2 For Type A PICCs, tags '82' and '86' are omitted in certain cases (see table footnote b in [Table 7](#)) and tags '83' and '87' are always omitted (see table footnote c in [Table 7](#)).

Annex A (informative)

Multi-Activation example

Table A.1 describes an example of the usage of Multi-Activation for three PICCs.

Table A.1 — Multi-Activation

PCD Action	PICC 1 state	PICC 2 state	PICC 3 state
Power On field			
Three PICCs enter the field.	IDLE	IDLE	IDLE
Activate PICC with CID = 1	PROTOCOL(1)	IDLE	IDLE
Any data transmission with CID = 1	PROTOCOL(1)	IDLE	IDLE
...			
Activate PICC with CID = 2	PROTOCOL(1)	PROTOCOL(2)	IDLE
Any data transmission with CID = 1,2	PROTOCOL(1)	PROTOCOL(2)	IDLE
...			
Activate PICC with CID = 3	PROTOCOL(1)	PROTOCOL(2)	PROTOCOL(3)
Any data transmission with CID = 1,2,3	PROTOCOL(1)	PROTOCOL(2)	PROTOCOL(3)
...			
S(DESELECT) command with CID = 3	PROTOCOL(1)	PROTOCOL(2)	HALT
S(DESELECT) command with CID = 2	PROTOCOL(1)	HALT	HALT
S(DESELECT) command with CID = 1	HALT	HALT	HALT
...			
NOTE The number <i>n</i> in PROTOCOL(<i>n</i>) represents the CID.			

ISO/IEC 14443-4:2018(E) © IEC 2018 – All rights reserved

Annex B (informative)

Protocol scenarios

B.1 General

This Annex gives some scenarios in [Tables B.1](#) to [B.26](#) for an error-free operation, as well as for error handling. These scenarios may be used to build test cases for compliance tests.

B.2 Notation

Any block	$\equiv\equiv\Rightarrow$	correctly received
Any block	$\equiv\neq\Rightarrow$	erroneously received
Any block	$\equiv\Rightarrow$	nothing received (FWT time-out)
Separator line	—	end of the smallest protocol operation
$I(1)_x$		I-block with chaining bit set and block number x
$I(0)_x$		I-block with chaining bit not set (last block of chain) and block number x
$R(ACK)_x$		R-block indicating a positive acknowledge
$R(NAK)_x$		R-block indicating a negative acknowledge
$S(...)$		S-block

The block numbering in a scenario always starts with the PCD's current block number for the destination PICC. For ease of presentation, scenarios start after the PICC activation sequence and hence, the current block numbers start with 0 for the PCD and with one for the PICC.

B.3 Error-free operation

B.3.1 Exchange of I-blocks

Table B.1 — Scenario 1 Exchange of I-blocks

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		$I(0)_0$	$\equiv\equiv\Rightarrow$	0	rule D
2. rule B	1		$<\equiv\equiv$	$I(0)_0$	rule 10
3.		$I(0)_1$	$\equiv\equiv\Rightarrow$	1	rule D
4. rule B	0		$<\equiv\equiv$	$I(0)_1$	rule 10

B.3.2 Request for waiting time extension

Table B.2 — Scenario 2 Waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2.			<==== S(WTX) request		rule 9
3. rule 3		S(WTX) response	====>		
4. rule B	1		<==== I(0) ₀		rule 10
5.		I(0) ₁	====>	1	rule D
6. rule B	0		<==== I(0) ₁		rule 10

B.3.3 DESELECT

Table B.3 — Scenario 3 DESELECT

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<==== I(0) ₀		rule 10
3.		S(DESELECT) request	====>		
4.			<==== S(DESELECT) response		rule 3

B.3.4 Chaining

Table B.4 — Scenario 4 PCD uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(1) ₀	====>	0	rule D
2. rule B	1		<==== R(ACK) ₀		rule 2
3. rule 7		I(0) ₁	====>	1	rule D
4. rule B	0		<==== I(0) ₁		rule 10
5.		I(0) ₀	====>	0	rule D
6. rule B	1		<==== I(0) ₀		rule 10

Table B.5 — Scenario 5 PICC uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(1) ₀	====>	0	rule D
2. rule B	1		<==== I(1) ₀		rule 10
3. rule 2		R(ACK) ₁	====>	1	rule E
4. rule B	0		<==== I(0) ₁		rule 13
5.		I(0) ₀	====>	0	rule D
6. rule B	1		<==== I(0) ₀		rule 10

B.3.5 PICC Presence check

Table B.6 — Scenario 6 PICC presence check using method 1

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1 and method 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10 note

Table B.7 — Scenario 7 PICC presence check using method 2 (before the first I-block exchange)

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1 and method 2		R(NAK) ₀	====>		rule E note
2. no change			<====	R(ACK) ₁	rule 12
3. rule 6 note and method 2	R(NAK) ₀		====>		rule E note
4. rule 6 note	no change		<====	R(ACK) ₁	rule 12
5.	I(0) ₀	====>		0	rule D
6. rule B	1	<====	I(0) ₀		rule 10

Table B.8 — Scenario 8 PICC presence check using method 2-a (after the first I-block exchange)

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10
3. method 2-a		R(NAK) ₁	====>		rule E note
4. rule 6 note	no change		<====	R(ACK) ₀	rule 12
5.	I(0) ₁	====>		1	rule D
6. rule B	0	<====	I(0) ₁		rule 10

Table B.9 — Scenario 9 PICC presence check using method 2-b (after the first I-block exchange)

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10
3. method 2-b	0	R(NAK) ₀	====>		
4. rule B	1		<====	I(0) ₀	rule 11
5.	I(0) ₁	====>		1	rule D
6. rule B	0	<====	I(0) ₁		rule 10

B.3.6 Exchange of additional parameters

Table B.10 — Scenario 25 Exchange of additional parameters

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10
3.		S(PARAMETERS) request	====>		
4.	1		<====	S(PARAMETERS) response	rule 3
5.		I(0) ₁	====>	1	rule D
6. rule B	0		<====	I(0) ₁	rule 10

B.4 Error handling

B.4.1 Exchange of I-blocks

Table B.11 — Scenario 10 Start of protocol

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	=≠=>		
2. time-out			<==	-	
3. rule 4		R(NAK) ₀	====>		
4. no change			<====	R(ACK) ₁	rule 12
5. rule 6		I(0) ₀	====>	0	rule D
6. rule B	1		<====	I(0) ₀	rule 10
7.		I(0) ₁	====>	1	rule D
8. rule B	0		<====	I(0) ₁	rule 10

Table B.12 — Scenario 11 Exchange of I-blocks

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10
3.		I(0) ₁	=≠=>		
4. time-out			<==	-	
5. rule 4		R(NAK) ₁	====>		
6. no change			<====	R(ACK) ₀	rule 12
7. rule 6		I(0) ₁	====>	1	rule D
8. rule B	0		<====	I(0) ₁	rule 10
9.		I(0) ₀	====>	0	rule D
10. rule B	1		<====	I(0) ₀	rule 10

Table B.13 — Scenario 12 Exchange of I-blocks

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2.			<==#=	I(0) ₀	rule 10
3. rule 4		R(NAK) ₀	====>		
4. rule B	1		<====	I(0) ₀	rule 11
5.		I(0) ₁	====>	1	rule D
6. rule B	0		<====	I(0) ₁	rule 10

Table B.14 — Scenario 13 Exchange of I-blocks

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2.			<==#=	I(0) ₀	rule 10
3. rule 4		R(NAK) ₀	=#==>		
4. time-out			<==	-	
5. rule 4		R(NAK) ₀	====>		
6. rule B	1		<====	I(0) ₀	rule 11
7.		I(0) ₁	====>	1	rule D
8. rule B	0		<====	I(0) ₁	rule 10

Table B.15 — Scenario 26 Exchange of I-blocks

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B	1		<====	I(0) ₀	rule 10
3.	S(PARAMETERS) request		=#==>		
4. time-out			<==	-	
5. rule 8		S(PARAMETERS) request	====>		
6.			<====	S(PARAMETERS) response	rule 3
7.		I(0) ₁	====>	1	rule D
8. rule B	0		<====	I(0) ₁	rule 10

B.4.2 Request for waiting time extension

Table B.16 — Scenario 14 Request for waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	==>	0	rule D
2.			<== S(WTX) request		rule 9
3. rule 4		R(NAK) ₀	==>		
4.			<== S(WTX) request		rule 11
5. rule 3		S(WTX) response	==>		
6. rule B	1		<== I(0) ₀		rule 10
7.		I(0) ₁	==>	1	rule D
8. rule B	0		<== I(0) ₁		rule 10

Table B.17 — Scenario 15 Request for waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	==>	0	rule D
2.			<== S(WTX) request		rule 9
3. rule 4		R(NAK) ₀	==>		
4. time-out			<== -		
5. rule 4		R(NAK) ₀	==>		
6.			<== S(WTX) request		rule 11
7. rule 3		S(WTX) response	==>		
8. rule B	1		<== I(0) ₀		rule 10
9.		I(0) ₁	==>	1	rule D
10. rule B	0		<== I(0) ₁		rule 10

Table B.18 — Scenario 16 Request for waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	==>	0	rule D
2.			<== S(WTX) request		rule 9
3. rule 3		S(WTX) response	==>		
4. time-out			<== -		
5. rule 4		R(NAK) ₀	==>		
6.			<== S(WTX) request		rule 11
7. rule 3		S(WTX) response	==>		
8. rule B	1		<== I(0) ₀		rule 10
9.		I(0) ₁	==>	1	rule D
10. rule B	0		<== I(0) ₁		rule 10

Table B.19 — Scenario 17 Request for waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2.			<==== S(WTX) request		rule 9
3. rule 3		S(WTX) response	====>		
4.			<==#= I(0) ₀		rule 10
5. rule 4		R(NAK) ₀	====>		
6. rule B	1		<==== I(0) ₀		rule 11
7.		I(0) ₁	====>	1	rule D
8. rule B	0		<==== I(0) ₁		rule 10

Table B.20 — Scenario 18 Request for waiting time extension

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2.			<==== S(WTX) request		rule 9
3. rule 3		S(WTX) response	====>		
4.			<==#= I(0) ₀		rule 10
5. rule 4		R(NAK) ₀	=#=>		
6. time-out			<== -		
7. rule 4		R(NAK) ₀	====>		
8. rule B	1		<==== I(0) ₀		rule 11
9.		I(0) ₁	====>	1	rule D
10. rule B	0		<==== I(0) ₁		rule 10

B.4.3 DESELECT

Table B.21 — Scenario 19 DESELECT

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0) ₀	====>	0	rule D
2. rule B			<==== I(0) ₀		rule 10
3.		S(DESELECT) request	=#=>		
4. time-out			<== -		rule 10
5. rule 8		S(DESELECT) request	====>		
6.			<==== S(DESELECT) response		rule 3

B.4.4 Chaining

Table B.22 — Scenario 20 PCD uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(1) ₀	====>	0	rule D
2.			<==#=	R(ACK) ₀	rule 2
3. rule 4		R(NAK) ₀	====>		
4. rule B	1		<====	R(ACK) ₀	rule 11
5. rule 7		I(1) ₁	====>	1	rule D
6. rule B	0		<====	R(ACK) ₁	rule 2
7. rule 7		I(0) ₀	====>	0	rule D
8. rule B	1		<====	I(0) ₀	rule 10
9.		I(0) ₁	====>	1	rule D
10. rule B	0		<====	I(0) ₁	rule 10

Table B.23 — Scenario 21 PCD uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(1) ₀	====>	0	rule D
2. rule B	1		<====	R(ACK) ₀	rule 2
3. rule 7		I(1) ₁	=#=>		
4. time-out			<==	-	
5. rule 4		R(NAK) ₁	====>		
6. no change			<====	R(ACK) ₀	rule 12
7. rule 6		I(1) ₁	====>	1	rule D
8. rule B	0		<====	R(ACK) ₁	rule 2
9. rule 7		I(0) ₀	====>	0	rule D
10. rule B	1		<====	I(0) ₀	rule 10
11.		I(0) ₁	====>	1	rule D
12. rule B	0		<====	I(0) ₁	rule 10

Table B.24 — Scenario 22 PCD uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(1)_0	====>	0	rule D
2.			<==#=	R(ACK)_0	rule 2
3. rule 4		R(NAK)_0	=#=>		
4. time-out			<==	-	
5. rule 4		R(NAK)_0	====>		
6. rule B	1		<====	R(ACK)_0	rule 11
7. rule 7		I(1)_1	====>	1	rule D
8. rule B	0		<====	R(ACK)_1	rule 2
9. rule 7		I(0)_0	====>	0	rule D
10. rule B	1		<====	I(0)_0	rule 10
11.		I(0)_1	====>	1	rule D
12. rule B	0		<====	I(0)_1	rule 10

Table B.25 — Scenario 23 PICC uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0)_0	====>	0	rule D
2. rule B	1		<====	I(1)_0	rule 10
3. rule 2		R(ACK)_1	=#=>		
4. time-out			<==	-	
5. rule 5		R(ACK)_1	====>	1	rule E
6. rule B	0		<====	I(1)_1	rule 13
7. rule 2		R(ACK)_0	====>	0	rule E
8. rule B	1		<====	I(0)_0	rule 13
9.		I(0)_1	====>	1	rule D
10. rule B	0		<====	I(0)_1	rule 10

Table B.26 — Scenario 24 PICC uses chaining

Comment	Block No. (0)	PCD	PICC	Block No. (1)	Comment
1. rule 1		I(0)_0	====>	0	rule D
2. rule B	1		<====	I(1)_0	rule 10
3. rule 2		R(ACK)_1	====>	1	rule E
4.			<==#=	I(1)_1	rule 13
5. rule 5		R(ACK)_1	====>	no change	
6. rule B	0		<====	I(1)_1	rule 11
7. rule 2		R(ACK)_0	====>	0	rule E
8. rule B	1		<====	I(0)_0	rule 13
9.		I(0)_1	====>	1	rule D
10. rule B	0		<====	I(0)_1	rule 10

Annex C (informative)

Block and frame coding overview

This Annex gives an overview of the different block and frame coding sent by the PCD.

Definitions made in ISO/IEC 14443-3:

REQA	(0100110)b (7 bit)
WUPA	(1010010)b (7 bit)
REQB/WUPB	(00000101)b
Slot-MARKER (Type B only)	(xxxx0101)b
SELECT (Type A only)	(1001xxxx)b
ATTRIB (Type B only)	(00011101)b
HLTA	(01010000)b
HLTB	(01010000)b

Definitions made in this document:

RATS	(11100000)b
PPS	(1101xxxx)b
I-block PCB	(00xxxxxxxx)b (not (00xxx101)b)
R-block PCB	(10xxxxxxxx)b (not (1001xxxx)b)
S-block PCB	(11xxxxxxxx)b (not (1110xxxx)b and not (1101xxxx)b)

[Table C.1](#) describes the first byte of the defined block and frame coding.

Table C.1 — Block and frame coding

Bit	I-block PCB	R-block PCB	S-block PCB			REQB/ WUPB	Slot- MARKER	SE- LECT	AT- TRIB	HLTA	HLTB	RATS	PPS		
			De- se- lect	WTX	Pa- ram- eters										
b8	0	1	1			0	x	1	0	0	0	1	1		
b7	0	0	1			0	x	0	0	1	1	1	1		
b6	0	1	0	1		0	x	0	0	0	0	1	0		
b5	Chaining	ACK/NAK	0			0	x	1	1	1	1	0	1		
b4	CID	CID	CID			0	0	x	1	0	0	0	x		
b3	NAD	0 (no NAD)	0 (no NAD)			1	1	x	1	0	0	0	x		
b2	1	1	1	0		0	0	x	0	0	0	0	x		
b1	Block number	Block number	0			1	1	x	1	0	0	0	x		

Annex D

(deliberately left blank)

Annex E (informative)

CRC_32 encoding

E.1 CRC_32 encoding

This Annex is provided for explanatory purposes and indicates the bit patterns in enhanced block. It is included for the purpose of checking a CRC_32 encoding.

Initial Value = 'FFFFFFFF'.

EXAMPLE In Figure E.1, transmission of first byte = '06', second byte = '00', third byte = '0A', fourth byte = '01', fifth byte = '01', sixth byte = '02', and appended CRC_32 is illustrated. Calculated CRC_32 = '8098F1FE'.

1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	CRC_32			
06	00	0A	01	01	02	80	98	F1	FE

Figure E.1 — Enhanced frame including CRC_32

E.2 Code sample written in ANSI C language for CRC_32 calculation

```
// ComputeCrc32.cpp : Defines the entry point for the console application
#include <stdio.h>

unsigned long ComputeCrc32(unsigned char *Data)
{
    unsigned long c;
    unsigned char d, e, f;

    unsigned int i;

    unsigned int Length = 6;
    // initial value
    c = 0xffffffff;
    // compute CRC
    for (i = 0; i < Length; i++) {
        d = Data[i];

        e = c ^ d;
        f = e ^ (e << 6);
        c = (c >> 8) ^ (f << 24) ^ (f << 23) ^ (f << 22) ^
            (f << 20) ^ (f << 19) ^ (f << 17) ^ (f << 16) ^
            (f << 14) ^ (f << 13) ^ (f << 12) ^ (f << 8) ^
            (f << 2) ^ (f << 1) ^ (f >> 2);
    }
    // invert result
    c = c ^ 0xffffffff;
}

return c;
}

int main(void)
{
    unsigned char BuffCRC_32[6] = {0x06, 0x00, 0x0A, 0x01, 0x01, 0x02};
    unsigned int Crc32;
    int i;
```

```
printf("\n");
printf("CRC-32 reference results ISO/IEC 14443-4\n");
printf("Crc-32 G(x) = x^32 + x^26 + x^23 + x^22 + x^16 + x^12 + x^11 +
       x^10 + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1\n\n");
printf("CRC_32 of [ ");
for(i=0; i<6; i++) printf("%02X ",BuffCRC_32[i]);
Crc32 = ComputeCrc32(BuffCRC_32);

// revert byte order to get least significant byte first
printf("] Transmitted: %02X then %02X then %02X then %02X.\n",
       (Crc32 & 0xFF), ((Crc32 >> 8) & 0xFF), ((Crc32 >> 16) & 0xFF), ((Crc32 >> 24) &
0xFF));

return 0;
}
```

Annex F (informative)

Frame with error correction

F.1 Example

This Annex is provided for explanatory purposes and shows how an I-block in using enhanced block format is transmitted. The example in Figure F.1 uses CID = '01' and INF-field = '0102'.

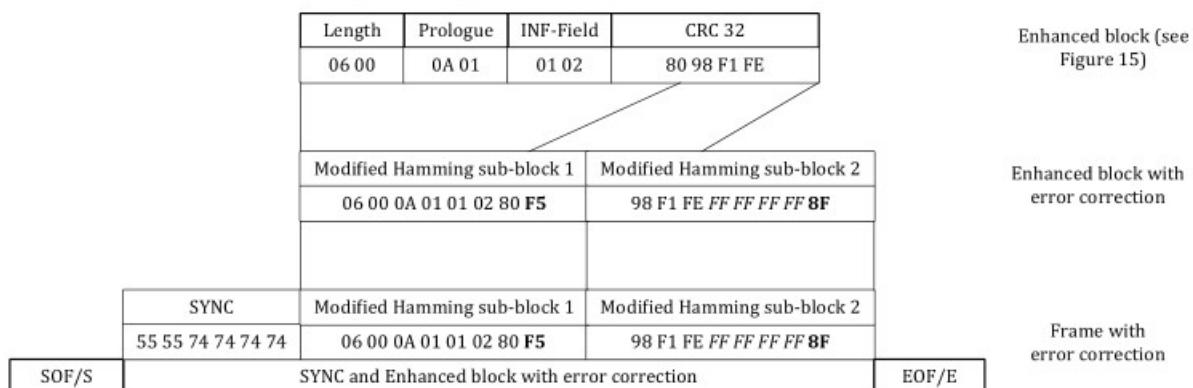


Figure F.1 — Composition of frame with error correction

The enhanced block is divided into 7-byte blocks. In our example, there are two blocks. The second block needs to be padded with 4 'FF' bytes, illustrated in italic letters in Figure F.1. Each block has appended the Hamming control byte (bold letters in Figure F.1). These two blocks are called enhanced block with error correction. Before this enhanced block with error correction SYNC is prepended. Up to here everything is the same for all communication types.

When using the frame format defined in [10.2](#), the frame with error correction starts with S and ends with E as specified in ISO/IEC 14443-3:2018, Clause 4. SYNC (which cannot be suppressed) and the enhanced block with error correction are sent without parity bits of ISO/IEC 14443-3:2018, 6.2.3.2.

When using the frame format defined in [10.3](#), the frame with error correction starts with SOF and ends with EOF. SYNC and the enhanced block with error correction are sent in the character transmission format defined in ISO/IEC 14443-3:2018, 7.1.1. Start bit, stop bit, SOF, EOF and SYNC may be suppressed in accordance with [10.5](#).

F.2 Code sample written in ANSI C language for Hamming control byte calculation

```
#include <stdio.h>

unsigned char ComputeHammingByte(unsigned char* pEnhancedSubBlock)
{
    unsigned char currentByte;
    unsigned char currentBit;
    unsigned char c = 0x00; // Hamming control byte
    unsigned char A[] = { // Hamming control generation matrix A
        0x03, 0x05, 0x06, 0x07, 0x09, 0x0A, 0x0B, 0x0C,
        0x0D, 0x0E, 0x0F, 0x11, 0x12, 0x13, 0x14, 0x15,
        0x16, 0x17, 0x18, 0x19, 0x1A, 0x1B, 0x1C, 0x1D,
        0x1E, 0x1F, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26,
    };
    for (currentByte = 0; currentByte < 26; currentByte++)
    {
        for (currentBit = 0; currentBit < 8; currentBit++)
        {
            if ((currentByte & (1 << currentBit)) != 0)
                c ^= A[currentBit];
        }
    }
    return c;
}
```

```

        0x27, 0x28, 0x29, 0x2A, 0x2B, 0x2C, 0x2D, 0x2E,
        0x2F, 0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36,
        0x37, 0x38, 0x39, 0x3A, 0x3B, 0x3C, 0x3D, 0x3E,
    };

    // iterate through all bits of the 7-byte sub-block from enhanced block
    for (currentByte = 0; currentByte < 7; currentByte++) // 7 bytes
    {
        for (currentBit = 0; currentBit < 8; currentBit++) // 8 bits each
        {
            // if bit is equal 1 XOR with corresponding column of matrix A
            if ((pEnhancedSubBlock[currentByte] >> currentBit) & 0x01)
            {
                c = c ^ A[currentByte*8 + currentBit];
            }
        }
    }
    // add padding bits
    c = (c << 1) | 0x81;
    return c;
}

int main(void)
{
    unsigned char EnhancedSubBlock[7] = {0x06, 0x00, 0x0A, 0x01, 0x01, 0x02, 0x80};
    unsigned char HammingByte;
    int i;
    HammingByte = ComputeHammingByte(EnhancedSubBlock);
    printf("\n");
    printf("Hamming Control Byte reference results ISO/IEC 14443-4\n");
    printf("Hamming Control Byte of enhanced sub-block [ ");
    for(i=0; i<7; i++) printf("%02X ",EnhancedSubBlock[i]);
    printf("] is %02X \n\n", HammingByte);
    return 0;
}

```

Annex G

(informative)

Framing options

This Annex illustrates all possible framing options dependent on communication type, bit rate, frame format and communication direction.

In Table G.1, "yes" means that framing options may be applied and "no" means that framing options shall not be applied.

Table G.1 — Allowed application of framing options

	Standard frame		Frame with error correction	
	PCD to PICC	PICC to PCD	PCD to PICC	PICC to PCD
Type A, bit rate $\leq fc/16$	no ^a	no ^a	no ^a	no ^a
Type A, $fc/16 < \text{bit rate} \leq fc/2$	no ^b	no ^a	yes	no ^a
Type A, bit rate $> fc/2$	no ^a	N/A	no ^a	N/A
Type B, bit rate $\leq fc/2$	no ^b	yes	yes	yes
Type B, bit rate $> fc/2$	no ^a	N/A	yes	N/A

^a There are no start bits, stop bits, SOF and EOF in the frame format.

^b Start bits and stop bits are needed in the frame format in case of long series of '00' or 'FF' bytes.

When activating framing options more than once, the new settings override the previous one, e.g. first activating SYNC suppression leads to SYNC suppression after the acknowledgement from the PICC. Then activating for instance Start/Stop bit suppression, only Start/Stop bit suppression is active after the acknowledgement from the PICC.

Bibliography

- [1] ISO/IEC 7810, *Identification cards — Physical characteristics*
- [2] ISO/IEC 10536 (all parts), *Identification cards — Contactless integrated circuit(s) cards — Close-coupled cards*
- [3] ISO/IEC 13239, *Information technology — Telecommunications and information exchange between systems — High-level data link control (HDLC) procedures*
- [4] ISO/IEC 15693 (all parts)²⁾, *Cards and security devices for personal identification — Contactless vicinity objects*
- [5] ISO/IEC 18092, *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1)*
- [6] ISO/IEC 21481, *Information technology — Telecommunications and information exchange between systems — Near Field Communication Interface and Protocol -2 (NFCIP-2)*

2) Third edition to be published.