Digital Protocol

Technical Specification

Version 2.3

2021-08-03

[DIGITAL]

NFC Forum™

Contents

Figures

Tables

Requirements

1 Introduction

1.1 Scope

This specification covers the digital interface and the half-duplex transmission protocol of the NFC Forum Device.

The digital interface and transmission protocol include the modulation schemes, bit-level coding, bit rates, frame formats, exchange protocols and command sets.

1.2 Applicable Documents or References

The following documents contain provisions that are referenced in this specification. The latest version including all published amendments applies unless a publication date is explicitly stated.

[ACTIVITY]	Activity Technical Specification, NFC Forum
[ANALOG]	Analog Technical Specification, NFC Forum
[EMV_CLESS]	EMV Contactless Communication Protocol Specification, EMVCo
[ISO/IEC_7816-4]	ISO/IEC 7816-4, Identification cards – Integrated circuit cards – Organization, security and commands for interchange, ISO/IEC
[ISO/IEC_7816-6]	ISO/IEC 7816-6, Identification cards – Integrated circuit cards – Interindustry data elements for interchange, ISO/IEC
[ISO/IEC_13239]	ISO/IEC 13239, Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures, ISO/IEC

[ISO/IEC_14443] ISO/IEC 14443,
Identification cards – Contactless integrated circuit cards – Proximity cards. Includes:

- [ISO/IEC 14443-1], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 1: Physical characteristics
- [ISO/IEC 14443-2], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 2: Radio frequency power and signal balance
- [ISO/IEC 14443-3], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 3: Initialization and anticollision
- [ISO/IEC 14443-4], Identification cards – Contactless integrated circuit cards – Proximity cards – Part 4: Transmission protocol

ISO/IEC

[ISO/IEC_15693] ISO/IEC 15693,
Identification cards – Contactless integrated circuit cards – Vicinity cards

Includes:

- [ISO/IEC 15693-2], Identification cards – Contactless integrated circuit cards – Vicinity cards – Part 2: Air interface and initialization
- [ISO/IEC 15693-3], Identification cards – Contactless integrated circuit cards – Vicinity cards – Part 3: Anticollision and transmission protocol,

ISO/IEC

[ISO/IEC_18092] ISO/IEC 18092,
Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)
ISO/IEC

[JIS_X_6319-4] JIS X 6319-4,
Specification of implementation for integrated circuit(s) cards – Part 4: High speed proximity cards
JIS

[RFC2119] Key words for use in RFCs to Indicate Requirement Levels, RFC 2119,
S. Bradner,
March 1997
Internet Engineering Task Force

[T2T] Type 2 Tag Technical Specification,
NFC Forum

[T3T] Type 3 Tag Technical Specification,
NFC Forum

[T4T] Type 4 Tag Technical Specification,
NFC Forum

[T5T] Type 5 Tag Technical Specification,
NFC Forum

1.3 Administration

The NFC Forum Digital Protocol Technical Specification is an open specification supported by the Near Field Communication Forum, Inc., located at:

401 Edgewater Place, Suite 600
Wakefield, MA, 01880

Tel.: +1 781-876-8955
Fax: +1 781-610-9864

http://www.nfc-forum.org/

The NFC Forum, Inc. maintains this specification.

1.4 Name and Logo Usage

The Near Field Communication Forum's policy regarding the use of the trademarks *NFC Forum* and the NFC Forum logo is as follows:

- Any company MAY claim compatibility with NFC Forum specifications, whether a member of the NFC Forum or not.
- Permission to use the NFC Forum logos is automatically granted to designated members only as stipulated on the most recent Membership Privileges document, during the period of time for which their membership dues are paid.
- Member's distributors and sales representatives MAY use the NFC Forum logo in promoting member's products sold under the name of the member.
- The logo SHALL be printed in black or in color, as illustrated on the Logo Page that is available from the NFC Forum at the address above. The aspect ratio of the logo SHALL be maintained, but the size MAY be varied. Nothing MAY be added to or deleted from the logos.
- Since the NFC Forum name is a trademark of the Near Field Communication Forum, the following statement SHALL be included in all published literature and advertising material in which the name or logo appears:

  NFC Forum and the NFC Forum logo are trademarks of the Near Field Communication Forum.

1.5 Intellectual Property

The Digital Protocol Specification conforms to the Intellectual Property guidelines specified in the NFC Forum's *Intellectual Property Rights Policy*, as outlined in the NFC Forum *Rules of Procedure*. These documents are available on the NFC Forum website.

1.6 Acknowledgements

Some information has been derived from [ISO/IEC_14443] and [ISO/IEC_18092].

British Standards can be obtained in PDF or hardcopy formats from the BSI online shop at www.bsigroup.com/Shop or by contacting BSI Customer Services for hardcopies only at +44 (0)20 8996 9001; email: cservices@bsigroup.com.

1.7 Special Word Usage

The key words “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT” and “MAY” in this document are to be interpreted as described in [RFC2119].

1.8 Requirement Numbering

Requirements in this document are uniquely numbered with the number appearing next to each requirement. For example:

Table 1: Sample Requirement

1.8.1.1	A car SHALL have four wheels.

A requirement can have different numbers in different versions of the specifications. Hence, all references to a requirement need to include the version of the document as well as the requirement’s number.

1.9 Notational Conventions

1.9.1 Notations

The notations shown in Table 2 apply in this document.

Table 2: Notational Conventions

Notation	Description
XYh	Hexadecimal notation. Hexadecimal numbers are represented using the numbers 0 - 9 and the characters A – F. An "h" is added at the end. The most significant byte (MSB) is shown on the left; the least significant byte (LSB) on the right. Example: F5h
xyb	Binary notation. Binary numbers are represented by strings of the digits 0 and 1, shown with the most significant bit (msb) on the left and the least significant bit (lsb) on the right. A "b" is added at the end. Example: 11110101b
xy	Decimal notation Decimal numbers are represented without any tailing character. Example: 245
⌈..⌉	A roundup integer function is expressed by the brackets ⌈..⌉ Example: ⌈7/8⌉ = 1, ⌈8/8⌉ = 1, ⌈9/8⌉ = 2
Specially Defined Names	Terms defined in the Glossary or other NFC Technical Specification Glossaries are written with initial capital letters.
STATE	Names of defined States are written in bold all-capital **COURIER FONT** letters.
COMMAND and RESPONSE	The defined Command and Response names are written in non-bold all-capital letters.
PARAMETER	Parameter names are written in non-bold all-capital letters. Parameter names start with one of the following prefixes: CON_ Prefix for Configuration Parameters (e.g., CON_DEVICES_LIMIT_A). INT_ Prefix for variables used in the Activities (e.g., INT_COLL_PEND). GRE_ Prefix for variables used in the Greedy Collection (e.g., GRE_POLL_A).

1.9.2 Values of Parameters

Throughout the document, symbols are used to identify the values of parameters. The actual values of the parameters are listed in Appendix B. Symbols referenced in Appendix B are written in **Arial bold** to distinguish them in the text.

1.9.3 RFU, Fixed Values, Not Allowed Values, Undefined Values

The following tables list global definitions and related (default) requirements for RFU, Fixed Values, Not Allowed Values and Undefined Values as being used throughout this document.

Requirements 1: RFU

Poll and Listen Mode	
1.9.3.1	Fields defined as RFU SHALL be sent by an NFC Forum Device with value 0, unless otherwise indicated. Values defined as "RFU" in this specification SHALL NOT be sent by an NFC Forum Device.
1.9.3.2	Unless explicitly stated otherwise, the NFC Forum Device that receives values defined as RFU SHALL disregard these values and SHALL keep the same interpretation of all other fields.

Requirements 2: Fixed Value

Poll and Listen Mode	
1.9.3.3	Fields that are defined as containing Fixed Values SHALL be sent by an NFC Forum Device with the values indicated.
1.9.3.4	When the NFC Forum Device receives fields defined as containing Fixed Values, it SHALL verify those values and SHALL treat any difference as a Syntax Error.

Requirements 3: Not Allowed Value

Poll and Listen Mode	
1.9.3.5	Values defined as Not Allowed Values SHALL NOT be sent by an NFC Forum Device.
1.9.3.6	Unless explicitly stated otherwise, when the NFC Forum Device receives a field containing one or more values that are defined as Not Allowed Values, it SHALL treat that receipt as a Syntax Error.

Requirements 4: Undefined Value

Poll and Listen Mode	
1.9.3.7	The NFC Forum Device that transmits fields containing Undefined Values MAY set these fields to any value.
1.9.3.8	The NFC Forum Device SHALL NOT change the processing of defined fields based on Undefined Values being present in other fields.

1.10 Abbreviations

Table 3 contains the definitions of the abbreviations and acronyms used in this specification.

Table 3: Abbreviations

Abbreviation	Description
ADC	Application Data Coding
ASK	Amplitude Shift Keying
ATN	Attention
ATS	Answer to Select
ATTRIB	Attribute
BCC	NFCID1 CLn check byte for NFC-A
bd	Bit Duration
BPSK	Binary Phase Shift Keying
CL*n*	Cascade Level n ($1 \leq n \leq 3$)
CRC	Cyclic Redundancy Check
CRC_A	CRC error detection code for NFC-A Technology
CRC_B	CRC error detection code for NFC-B Technology
CRC_F	CRC error detection code for NFC-F Technology
CRC_V	CRC error detection code for NFC-V Technology
CT	Cascade Tag
DEP	Data Exchange Protocol
DID	Device Identification number
$\mathbf{D}_{\text{LISTEN}\rightarrow\text{POLL}}$	Divisor for communication direction Listen→Poll
$\mathbf{D}_{\text{POLL}\rightarrow\text{LISTEN}}$	Divisor for communication direction Poll→Listen
Dri	Data rate Received by initiator
DSFID	Data Storage Format Identifier
Dsi	Data rate Sent by initiator

Abbreviation	Description
EGT	Extra Guard Time
EMD	ElectroMagnetic Disturbance
EoD	End of Data
EoF	End of Frame
EoS	End of Sequence
$\mathbf{f}_C$	Carrier Frequency
$\mathbf{f}_S$	Subcarrier Frequency
FO	Frame Options
FSC	Frame Size for proximity Card
FSCI	Frame Size for proximity Card Integer
FWI	Frame Waiting time Integer
FWT	Frame Waiting Time
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
JIS	Japanese Industrial Standard
LR	Length Reduction
LSB	Least Significant Byte
lsb	Least Significant Bit
Max	Maximum
MBL	Maximum Buffer Length
MBLI	Maximum Buffer Length Index
Min	Minimum
MRT	Maximum Response Time
MRTI	Maximum Response Time Information
MSB	Most Significant Byte
msb	Most Significant Bit
n.a.	Not Applicable
NAD	Node Address
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NFC-ACM	Near Field Communication – Active Communication Mode (based on either NFC-A or NFC-F Technology)

Abbreviation	Description
NFC-A	Near Field Communication – Type A Technology
NFC-B	Near Field Communication – Type B Technology
NFC-F	Near Field Communication – Type F Technology
NFC-V	Near Field Communication – Type V Technology
NFCID0	NFC-B identifier of the NFC Forum Device. NFCID0 is always 4 bytes long.
NFCID1	NFC-A identifier of the NFC Forum Device in the Passive Communication Mode. NFCID1 can be 4, 7, or 10 bytes long (simple, double, or triple size).
NFCID1 CL*n*	Contains the portion of the NFCID1 relative to the Cascade Level *n*. NFCID1 CL*n* is always 4 bytes long.
NFCID2	NFC-F identifier of the NFC Forum Device in the Passive Communication Mode. NFCID2 is always 8 bytes long.
NFCID3	NFCIP-1 identifier of the NFC Forum Device. NFCID3 is always 10 bytes long.
NFCIP-1	Near Field Communication Interface and Protocol, as specified in [ISO/IEC_18092].
NRZ-L	Non-Return to Zero (L for Level)
OOK	On-Off Keying
PDU	Protocol Data Unit
RATS	Request for Answer to Select
RC	Request Code
RD	Request Data
RF	Radio Frequency
RFU	Reserved for Future Use
RRDD	Reader-Reader Data Delay
RTOX	Response Timeout Extension
RWT	Response Waiting Time
SC	System Code
SDD	Single Device Detection
SFGI	Start-up Frame Guard time Integer
SFGT	Start-up Frame Guard Time

Abbreviation	Description
SoD	Start of Data
SoF	Start of Frame
SoS	Start of Sequence
TSN	Time Slot Number
UID	NFC-V Unique identifier
WT	Waiting Time, parameter to code Response Waiting Time

1.11 Glossary

Active Communication Mode

A communication mode in which each device generates an Operating Field when it has to send a frame to a peer device.

Activity

A process within an NFC Forum Device.

Block

Unit of data bytes transmitted as part of the ISO-DEP Protocol.

Command

An instruction transmitted from one device to another device in order to move the other device through a state machine.

Correct Frame

A frame without Transmission Error.

Cyclic Redundancy Check (CRC)

A checksum appended within the data segment before transmission, and verified afterwards by the recipient to detect Transmission Errors.

Initiator

Role of a Poller when it has gone through a number of Activities. In this mode, the NFC Forum Device communicates using the NFC-DEP Protocol.

ISO-DEP Protocol

Half-duplex block transmission protocol defined in Section 16 and based on [ISO/IEC_14443] and [EMV_CLESS].

Listen Frame

A frame sent by a Listener.

Listen Mode

The mode of an NFC Forum Device where it receives Commands and sends Responses.

Listener

An NFC Forum Device in Listen Mode.

NFC-DEP Protocol

Half-duplex block transmission protocol defined in Section 17 and based on [ISO/IEC_18092].

NFC Forum Device

A device that supports at least one communication protocol for at least one communication mode defined by the NFC Forum specifications. Currently the following NFC Forum Devices are defined:
NFC Universal Device, NFC Tag Device, and NFC Reader Device.

NFC Reader Device

An NFC Forum Device that supports the following Modus Operandi: Reader/Writer. It can also support Initiator.

NFC Tag Device

An NFC Forum Device that supports at least one communication protocol for Card Emulator and NDEF.

NFC Universal Device

An NFC Forum Device that supports the following Modus Operandi: Initiator, Target, and Reader/Writer. It can also support Card Emulator.

NFCIDx

The identifiers NFCID0, NFCID1, NFCID2, NFCID3 and UID for NFC-B, NFC-A, NFC-F, NFC-DEP and NFC-V respectively. Identifiers subsumed under the term NFCIDx always belong to the same Technology.

Operating Field

The radio frequency field created by the NFC Forum Device.

Passive Communication Mode

A communication mode in which one device generates an Operating Field and sends Commands to a second device. To respond, this second device uses load modulation, that is, it does not generate an Operating Field but it draws power from a Remote Field.

Peer

A role either equal to the role of an Initiator or to the role of a Target.

Poll Command

A Command to probe a Listener:

- ALL_REQ Command or SENS_REQ Command for NFC-A
- ALLB_REQ Command or SENSB_REQ Command for NFC-B
- SENSF_REQ Command for NFC-F
- INVENTORY_REQ Command for NFC-V
- ATR_REQ Command for NFC-ACM

Poll Frame

A frame sent by a Poller.

Poll Mode

The mode of an NFC Forum Device where it sends Commands and receives Responses.

Poller

An NFC Forum Device in Poll Mode.

Protocol Error

A Semantic Error or Syntax Error.

Read-Alike Command

Any Command that does not change the persistent State of the Type 5 Tag.

Reader/Writer

Role of a Poller when it has gone through a number of Activities. In this mode the Poller communicates with, Type 2 Tag, Type 3 Tags, Type 4 Tags or Type 5 Tags.

Response

Information sent from one device to another device upon receipt of a Command. The information received by the other device allows it to continue the data exchange.

Semantic Error

A Correct Frame that contains no Syntax Error is received when it is not expected.

State

A state of the Listener.

Syntax Error

A Correct Frame is received with invalid content. In this case, the coding of the Command or the block within the frame is not consistent with this specification.

Target

Role of a Listener when it has gone through a number of States. In this mode the Listener communicates using the NFC-DEP Protocol.

Technology

A group of transmission parameters defined by the NFC Forum specifications that make a complete communication protocol. A non-exhaustive list of transmission parameters is: RF carrier, communication mode, bit rate, modulation scheme, bit-level coding, frame format, protocol, and command set. NFC Forum defines four groups and therefore four Technologies: NFC-A, NFC-B, NFC-F, and NFC-V. The four Technologies use the same RF carrier (13.56 MHz). Each Technology uses its own modulation scheme, bit-level coding, and frame format, but can have the same protocol and command set.

Technology Subset

A legacy platform supporting a subset of a Technology. A Technology Subset supports at least the Poll Command of the Technology. The Technology Subsets are:

- Type 2 Tag Platform, which uses a particular subset of NFC-A, including anti-collision.
- Type 3 Tag Platform, which uses a particular subset of NFC-F.
- Type 4 Tag Platform, which uses a particular subset of NFC-A or NFC-B, including anti-collision.
- Type 5 Tag Platform, which uses a particular subset of NFC-V, including anti-collision.

Timeout Error

No Response has been received within the Response Waiting Time (RWT).

Transmission Error

An incorrect frame is received. In this case the signal modulation, the bit coding, the frame format, the timing, or the checksum is not consistent with this specification.

Valid Block, Valid PDU

A block or PDU without Protocol Error within a Correct Frame.

Valid Command, Valid Response

A Command or Response without Protocol Error within a Correct Frame.

Write-Alike Command

Any Command that both changes the persistent State of the Type 5 Tag platform and allows a long response time.

2 Overview

This specification covers the digital protocol of NFC Forum Devices.

An NFC Forum Device exchanges information, referred to as “payload” in this specification, by transmitting data packages. A data package optionally begins with a Start of Data (SoD) field consisting of a start byte and/or a length byte, followed by the payload field, and optionally ends with an End of Data (EoD) field. If present, the EoD consists of a two-byte checksum.

Data packages, which are digital information composed of the values Logic “0” and Logic “1”, are embedded in frames. These frames are optionally delimited by a Start of Frame (SoF) and an End of Frame (EoF). SoF, EoF, “0”, and “1” are coded digitally. This means that these elements are coded by using combinations of only two different physical characteristics (e.g., a high/low signal level or a 0°/180° signal phase).

Frames are embedded in sequences to synchronize and calibrate the sending and receiving devices and to trigger and stop the signal-decoding process in the receiving device. A sequence contains a frame with a particular leading signal pattern called Start of Sequence (SoS) and a particular closing signal pattern called End of Sequence (EoS).

Table 4 summarizes the structure of this document. The upper row contains the different Activities. The second column from the left indicates if Passive or Active Communication Mode is used. The third column from the left shows that the behavior of the NFC Forum Device during the RF Collision Avoidance, the Technology Detection Activity and the Collision Resolution Activity depends on its Technology only. The fourth through sixth columns from the left show that the behavior of the NFC Forum Device during the Device Activation Activity, the Data Exchange Activity, and the Device Deactivation Activity also depends on its Technology Subset.

Table 4: Activities versus Technology / Device Platform

<table>
<tr><th></th><th>Communication Mode</th><th>Listen, RF Collision Avoidance, Technology Detection, Collision Resolution</th><th>Device Activation</th><th>Data Exchange</th><th>Device Deactivation</th></tr>
<tr><td rowspan="6">Technology / Tag Platform / Protocol</td><td>Passive or Active Communication Mode</td><td rowspan="3">NFC-A Section 6</td><td colspan="3">NFC-DEP Protocol Section 17</td></tr>
<tr><td rowspan="3">Passive Communication Mode</td><td colspan="2">Type 2 Tag Platform Section 12</td><td>NFC-A Section 6</td></tr>
<tr><td>Type 4A Tag Platform Section 14</td><td colspan="2" rowspan="2">ISO-DEP Protocol Section 16</td></tr>
<tr><td>NFC-B Section 7</td><td>Type 4B Tag Platform Section 15</td></tr>
<tr><td>NFC-F Section 8</td><td colspan="2">Type 3 Tag Platform Section 13</td><td>N.A</td></tr>
<tr><td>Passive or Active Communication Mode</td><td colspan="3">NFC-DEP Protocol Section 17</td></tr>
<tr><td>Passive Communication Mode</td><td>NFC-V Section 9</td><td colspan="2">Type 5 Tag Platform Section 18</td><td>N.A</td></tr>
</table>

NOTE Half-duplex Protocol Section 10 is applicable to all technologies and platforms related to Device Activation, Data Exchange and Device Deactivation.

3 Bit Duration

The bit duration (bd) indicates the timing of a digital signal. In this specification the bd is the time it takes to transmit one unit of information.

3.1 Bit Duration NFC-A, NFC-B, and NFC-F

For communication direction Poll→Listen, the bd is defined as follows:

$1\ bd = 128 / (\mathbf{f}_C \times \mathbf{D}_{POLL \rightarrow LISTEN})$

For communication direction Listen→Poll, the bd is defined as follows:

$1\ bd = 128 / (\mathbf{f}_C \times \mathbf{D}_{LISTEN \rightarrow POLL})$

where $\mathbf{f}_C$ is the carrier frequency generated by the Poller ([ANALOG] defines $\mathbf{f}_C$), and the integers $\mathbf{D}_{POLL \rightarrow LISTEN}$ and $\mathbf{D}_{LISTEN \rightarrow POLL}$ are divisors whose values are technology dependent.

3.2 Bit Duration NFC-V

For communication direction Poll→Listen, the bd is defined:

$1\ bd = 512 / \mathbf{f}_C$

For communication direction Listen→Poll, the bd is defined:

$1\ bd = 512 / \mathbf{f}_C$.

4 EMD Handling

Electromagnetic disturbance (EMD) handling enhances the robustness of the contactless communication between NFC Forum Devices against electromagnetic disturbance.

It is important for a Poller to distinguish between EMD and frame reception errors when it interacts with a device that implements the Type 4 Tag Platform.

NOTE Definitions in Section 4 deviate from definitions in [ISO/IEC_14443].

Requirements 5: EMD Handling

Poll Mode	
4.1.1.1	When the Poller receives a frame it SHALL check for Transmission Errors when it is expecting any of the following: • RATS Response, as defined in Section 14 • ATTRIB Response, as defined in Section 15 • ISO-DEP Block, as defined in Section 16. If a Transmission Error is detected and the number of received bytes in a frame is less than 4, then the Poller SHALL ignore the frame without taking any action other than meeting the requirement that the Poller SHALL be ready to receive another frame no later than $\mathbf{t}_{RECOVERY,MAX}$ after the end of the modulation of the received bytes that contain the Transmission Error. Appendix B.1 lists the value of $\mathbf{t}_{RECOVERY,MAX}$.

NOTE The EMD handling defined here might overrule Transmission Error handling rules in cases in which a Transmission Error occurs in frames less than 4 bytes long. In these circumstances Timeout Errors might be triggered even when no EMD is involved. The developer needs to be careful to avoid the possibility that EMD will trigger an unnecessary error recovery process. When such error recovery is triggered by EMD (and before the device in Listen Mode has responded with any frame) it could possibly trigger a complete breakdown of the communication link.

5 Exception Processing

When unrecoverable errors are detected, so called “unrecoverable exceptions” are raised. All Technologies operate under the same overall requirements.

Requirements 6: Exception Processing – Unrecoverable Exceptions

Poll Mode	
5.1.1.1	The Poller SHALL stop the error handling when an unrecoverable protocol exception is raised. The recovery procedure after the Unrecoverable Protocol Exception is not defined by this version of the specification.
5.1.1.2	The Poller SHALL stop the error handling when an unrecoverable transmission exception is raised. The recovery procedure after the Unrecoverable Transmission Exception is not defined by this version of the specification.
5.1.1.3	The Poller SHALL stop the error handling when an unrecoverable timeout exception is raised. The recovery procedure after the Unrecoverable Timeout Exception is not defined by this version of the specification.

NOTE Future versions of the specification might define specific recovery procedures per individual exception.

6 NFC-A Technology

This section specifies the NFC-A Technology-related features of the NFC Forum Device.

This section is derived from [ISO/IEC_14443] and [ISO/IEC_18092].

6.1 Sequence Format

This section describes the sequence format for NFC-A Technology.

6.1.1 Poll→Listen Modulation

In Poll Mode the analog signal is modulated using Modified Miller coding with Amplitude Shift Keying (ASK) 100% modulation, as illustrated in Figure 1.

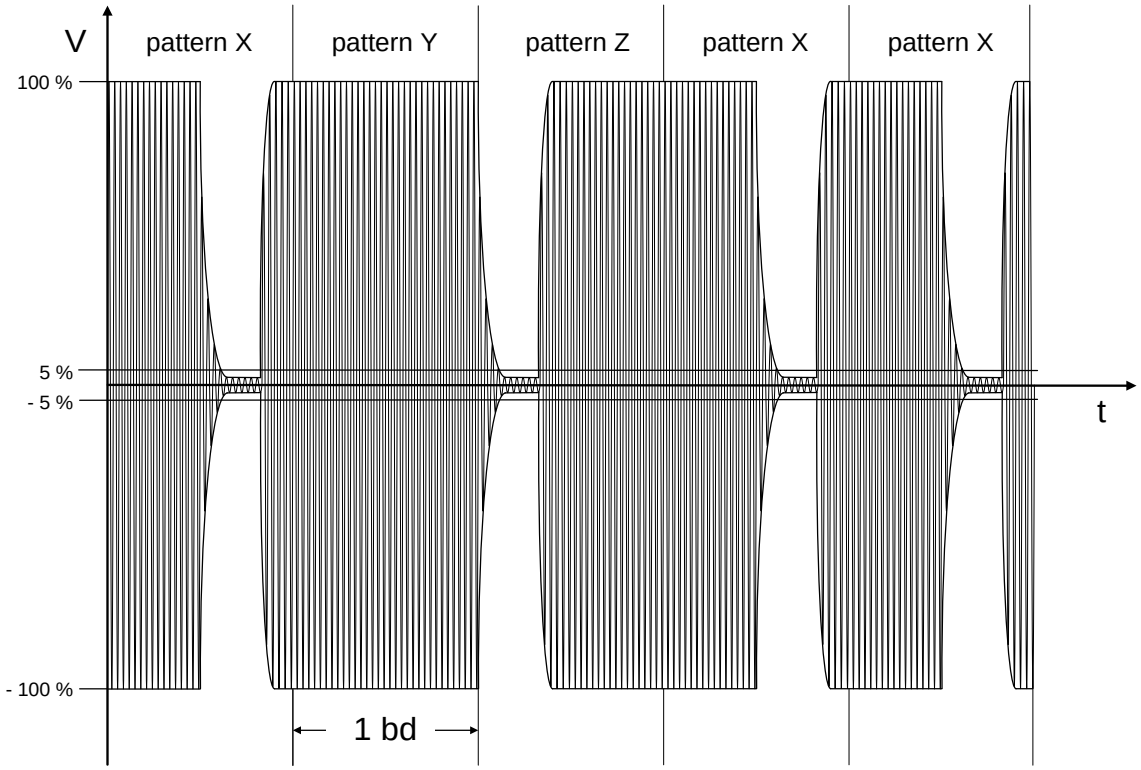


Figure 1: Modified Miller Coding with ASK 100%

The modulation principle of ASK 100% of the Operating Field is used to create $\mathbf{V}_2$. [ANALOG] defines $\mathbf{V}_2$.

Modified Miller coding uses these $\mathbf{V}_2$ to define three particular patterns: X, Y, and Z.

Requirements 7: NFC-A Signal Patterns Poll→Listen

Poll Mode		Listen Mode	
6.1.1.1	The Poller SHALL build signal patterns X, Y, and Z, as follows: • Pattern X: at the beginning of the bd, no modulation SHALL occur for a period of 1/2 bd. After the period of 1/2 bd, , a $\mathbf{V}_2$ SHALL occur. • Pattern Y: for 1 bd no modulation SHALL occur. • Pattern Z: at the beginning of the bd, a $\mathbf{V}_2$ SHALL occur. For the rest of the bd, no modulation SHALL occur.	6.1.1.2	The Listener SHALL read signal patterns X, Y, and Z, as follows: • If the Listener detects no modulation for a period of 1/2 of the bd and a $\mathbf{V}_2$ after that period, the Listener SHALL read this as pattern X. • If the Listener detects a $\mathbf{V}_2$ at the beginning of the bd and no modulation for the rest of the bd, the Listener SHALL read this as pattern Z. • If the Listener does not detect any modulation during the full bd, the Listener SHALL read this as pattern Y. • All other patterns SHALL be treated as invalid patterns.

NOTE The term “Pulse” used in [ISO/IEC_18092] is referred to as $\mathbf{V}_2$ in this document.

NOTE The processing of invalid patterns is not defined by this version of the specification.

6.1.2 Listen→Poll Modulation

In Listen Mode the analog signal is modulated using Manchester coding with On-Off Keying (OOK) subcarrier modulation (see Figure 2).

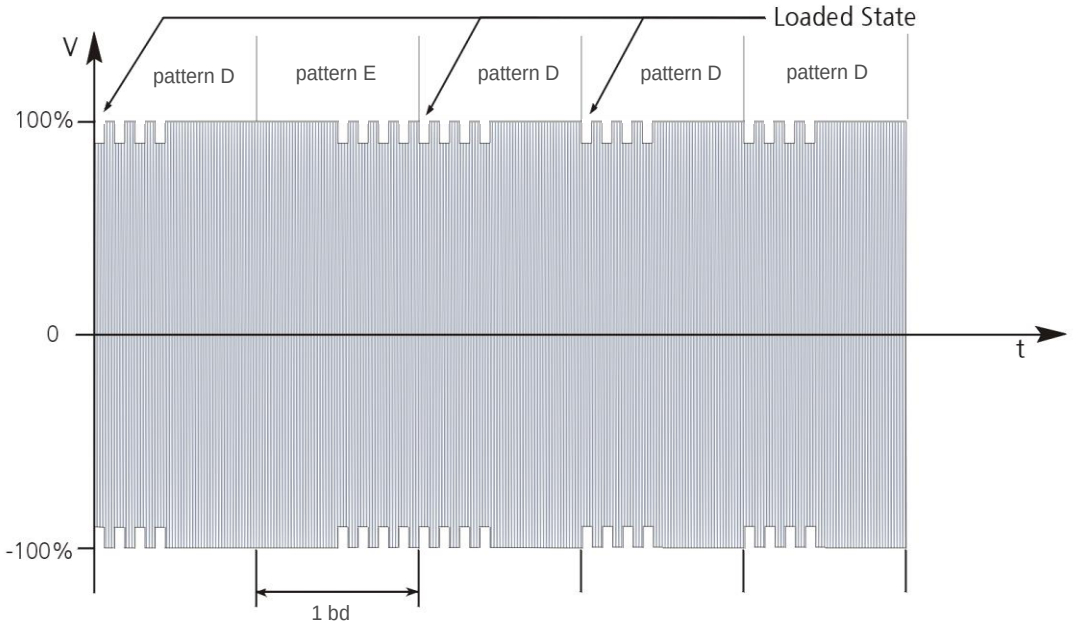


Figure 2: Manchester Coding with OOK

Manchester coding uses OOK subcarrier modulation to define three particular patterns: E, D, and F.

Requirements 8: NFC-A Modulation Listen→Poll

Listen Mode	
6.1.2.1	The Listener SHALL modulate the carrier with the subcarrier such that the bit period starts with the loaded state of the subcarrier (see Figure 2).

Requirements 9: NFC-A Signal Patterns Listen→Poll

Poll Mode		Listen Mode	
6.1.2.2	The Poller SHALL read the patterns D, E, and F as follows: • If the carrier is modulated with the subcarrier for the first 1/2 bd and is not modulated for the second 1/2bd, the Poller SHALL read that as pattern D. • If the carrier is not modulated with the subcarrier for the first 1/2 bd and is modulated for the second 1/2 bd, the Poller SHALL read that as pattern E. • If the carrier is not modulated with the subcarrier during 1 bd, the Poller SHALL read this as pattern F. • All other patterns SHALL be treated as invalid patterns.	6.1.2.3	The Listener SHALL build the patterns D, E, and F as follows: • Pattern D: the carrier SHALL be modulated with the subcarrier for the first 1/2 bd and SHALL NOT be modulated for the remainder of the bd. • Pattern E: the carrier SHALL NOT be modulated with the subcarrier for the first 1/2 bd and SHALL be modulated for the second 1/2 bd. • Pattern F: the carrier SHALL NOT be modulated with the subcarrier for 1 bd.

NOTE The processing of invalid patterns is not defined by this version of the specification.

6.1.3 Synchronization

NFC-A Technology does not require signal synchronization, and therefore SoS is not present.

6.1.4 De-synchronization

NFC-A Technology does not require signal de-synchronization, and therefore EoS is not present.

6.2 Bit Level Coding

6.2.1 Poll→Listen Coding Scheme

The patterns X, Y, and Z are used to code the digital Logic “0” and Logic “1”. Logic “0”s and Logic “1”s are the basic components of frames.

Requirements 10: NFC-A Bit Level Coding Poll→Listen

Poll Mode		Listen Mode	
6.2.1.1	The Poller SHALL code Logic “0” and Logic “1” as follows: • Logic “1”: pattern X • Logic “0”: pattern Y with the following exceptions: • Pattern Z SHALL be used to code the first Logic “0” (SoF). • If there are two or more contiguous Logic “0”s, pattern Z SHALL be used from the second Logic “0” onward.	6.2.1.2	The Listener SHALL decode Logic “0” and Logic “1” as follows: • The first pattern Z SHALL be decoded as Logic “0”. • If the Listener detects pattern X, then it SHALL decode this as Logic “1”. • If the Listener detects pattern Y after pattern X, then it SHALL decode pattern Y as Logic “0”. • If the Listener detects pattern Z after pattern Y, then it SHALL decode pattern Z as Logic “0”. • If the Listener detects pattern Z after pattern Z, then it SHALL decode the last pattern Z as Logic “0”.

6.2.2 Listen→Poll Coding Scheme

The patterns E and D are used to code the digital Logic “0” and Logic “1”. Logic “0”s and Logic “1”s, referred to as “data bits” in the following, are the basic components of frames.

Requirements 11: NFC-A Bit Level Coding Listen→Poll

Poll Mode		Listen Mode	
6.2.2.1	The Poller SHALL decode Logic “0” and Logic “1” as follows: • If the Poller detects pattern D, then it SHALL decode this as Logic “1”. • If the Poller detects pattern E, then it SHALL decode this as Logic “0”.	6.2.2.2	The Listener SHALL code Logic “0” and Logic “1” as follows: • Logic “1”: pattern D • Logic “0”: pattern E

6.3 Frame Format

This section defines the frames of NFC Forum Devices configured for NFC-A Technology. NFC-A Technology uses three types of frames: Short Frame, Standard Frame, and Bit Oriented Single Device Detection (SDD) Frame. The Short Frame is used to initiate communication. The Standard Frame is used for data exchange. The Bit Oriented SDD Frame is used for collision resolution.

NOTE Definitions in Section 6.3 deviate from definitions in [ISO/IEC_18092].

6.3.1 Common requirements

Data bits, when transmitted between NFC Forum Devices, are grouped within frames. NFC-A Technology groups the data bits together in a frame by adding an SoF and an EoF.

Requirements 12: NFC-A Frame Format

Poll and Listen Mode	
6.3.1.1	A frame SHALL start with a SoF. For Poll→Listen communication, the SoF SHALL be a Logic “0”. For Listen→Poll communication, the SoF SHALL be a Logic “1”.
6.3.1.2	A frame SHALL end with an EoF. For Poll→Listen communication, EoF SHALL be a Logic “0” followed by pattern Y. For Listen→Poll communication, EoF SHALL be a pattern F (see Requirement 6.1.2.3).

6.3.2 Short Frame

A Short Frame is used to initiate communication and consists of the following (see Figure 3):

- SoF
- Up to 7 data bits
- EoF.

No parity is added.

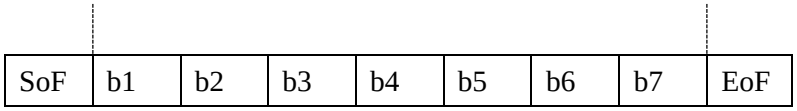


Figure 3: NFC-A Short Frame Format with 7 Data Bits

Requirements 13: NFC-A Short Frame Format

Poll and Listen Mode	
6.3.2.1	Following the SoF and preceding the EoF, the Short Frame SHALL contain less than 8 data bits with lsb first.

6.3.3 Standard Frame

Standard Frames are used for data exchange and consist of the following (see Figure 4):

- SoF
- n * (8 data bits + odd parity bit), with n ≥ 1
- EoF.

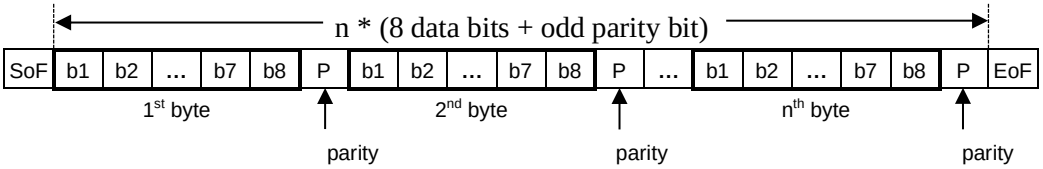


Figure 4: NFC-A Standard Frame Format

Requirements 14: NFC-A Standard Frame Format

Poll and Listen Mode	
6.3.3.1	Each 8 data bits, with lsb first, in a Standard Frame SHALL be followed by an odd parity bit. The parity bit P SHALL be set such that the number of Logic "1"s is odd in the set consisting of b1 to b8 and P.

6.3.4 Bit Oriented SDD Frame

Bit Oriented SDD frames are used for collision resolution and result from a Standard Frame with a length of 7 bytes that is split into two parts. The split can occur after any data bit. Figure 5 shows an example with the split after the first bit of the second byte.

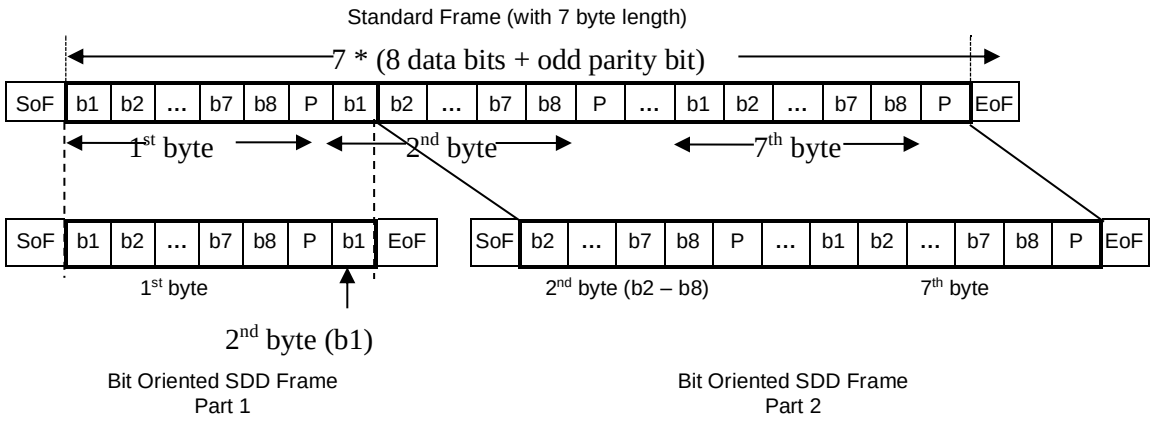


Figure 5: Bit Oriented SDD Frame (with Split after the First Bit of the Second Byte)

Bit Oriented SDD Frames are characterized as follows:

- Bit Oriented SDD Frames include a SoF and an EoF.
- Part 1 of a Bit Oriented SDD Frame is always sent by a Poller.
- Part 2 of a Bit Oriented SDD Frame is always sent by a Listener in response to part 1.
- The sum of data bits of part 1 and part 2 is 56 (7 bytes with 8 data bits each).
- The minimum length of part 1 is set to 16 data bits.
- The maximum length of part 1 is set to 48 data bits.
- As a result, the minimum length of part 2 is 8 data bits, and the maximum length of part 2 is 40 data bits.
- If the split occurs after the eighth data bit of a byte, then the related parity bit is added after the last data bit of part 1.
- If the split occurs at another position within a byte, then no parity bit is added after the last data bit of part 1, and the first parity bit of part 2 is undefined.

NOTE If the split occurs after the eighth (i.e., the last) data bit of a byte, then part 1 and part 2 of the Bit Oriented SDD frame have the same format as Standard Frames.

Requirements 15: Bit Oriented SDD Frame Format

Poll and Listen Mode	
6.3.4.1	Each 8 data bits in a Bit Oriented SDD frame SHALL be followed by an odd parity bit. The parity bit P SHALL be set such that the number of Logic “1”s is odd in the set consisting of b1 to b8 and P.

Requirements 16: Bit Oriented SDD Frame Format (Continued)

Poll Mode		Listen Mode	
6.3.4.2	The minimum length of part 1 SHALL be 16 data bits. The maximum length of part 1 SHALL be 48 data bits.	6.3.4.3	The sum of data bits of part 1 and part 2 SHALL be 56, as illustrated in Figure 5.
6.3.4.4	If the split occurs after the eighth data bit of a byte, then the related parity bit SHALL be added after the last data bit of part 1.	6.3.4.5	If the split occurs at a position within a byte (i.e., the split does not occur after the eighth data bit), then the first parity bit of part 2 is undefined. Thus, even though this bit SHALL be included, it MAY be set to any value.

6.4 Data and Payload Format

Data transmitted in an NFC-A Technology Standard Frame (i.e., the bytes between SoF and EoF), have the following substructure. They consist of the payload and, depending on the particular payload, of an End of Data (EoD) field (the Start of Data (SoD) field is not used).

If present, the EoD contains a two-byte checksum referred to as CRC_A. The CRC_A is defined as a function of k data bits. The number of bits k is a multiple of eight. The input for the CRC_A calculation is the payload.

Figure 6 illustrates the NFC-A data and payload format for a Standard Frame. Table 5 shows the cases in which the EoD is present.

Data					
Payload (Command or Response)				**[EoD]**	
Byte 1	Byte 2	…	Byte n	[CRC_A1]	[CRC_A2]

Figure 6: NFC-A Data and Payload Format

Data embedded in Short Frames or Bit Oriented SDD frames do not include an EoD.

Requirements 17: NFC-A Data and Payload Format

Poll and Listen Mode	
6.4.1.1	Data SHALL be transmitted in NFC-A Standard Frames, as defined in 6.3.3.
6.4.1.2	If needed according to Table 5, the payload SHALL be followed by an EoD at the position, as indicated in Figure 6. The EoD SHALL contain a CRC_A.
6.4.1.3	The CRC_A SHALL be calculated as defined in [ISO/IEC_13239], but the initial register content SHALL be 6363h and the register content SHALL NOT be inverted after calculation. CRC_A1 is the LSB and CRC_A2 is the MSB. The input for the CRC_A calculation SHALL be the payload.
6.4.1.4	The NFC Forum Device SHALL compare the received CRC_A with the CRC_A calculated according to Requirement 6.4.1.3. If different, then the NFC Forum Device SHALL treat the received data as a Transmission Error.

NOTE Transmission Errors on the Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

6.5 Command Set

A payload exchanged between NFC Forum Devices consists of Commands and Responses. Table 5 lists the Commands that are available to the NFC Forum Device configured for NFC-A Technology. For each Command the corresponding Response is indicated. Furthermore, Table 5 shows the frame type that a specific Command is embedded in and whether an EoD is present. Section 6.3 defines the NFC-A frames. Section 6.4 defines the data and payload format for the Standard Frame.

Table 5: NFC-A Command Set

Command	Response	EoD Present	Frame Type
ALL_REQ, SENS_REQ		No	Short Frame
	SENS_RES	No	Standard Frame
SDD_REQ		No	Bit Oriented SDD Frame
	SDD_RES	No	Bit Oriented SDD Frame
SEL_REQ		Yes	Standard Frame
	SEL_RES	Yes	Standard Frame
SLP_REQ		Yes	Standard Frame

NOTE The Commands and Responses listed in Table 5 have different names in [ISO/IEC_14443]: ALL_REQ refers to WUPA, SENS_REQ refers to REQA, SENS_RES refers to ATQA, and SLP_REQ refers to HLTA.

Table 6 lists all Commands that use Short Frames:

Table 6: Short Frame Commands Coding

b7	b6	b5	b4	b3	b2	b1	Meaning
0	1	0	0	1	1	0	SENS_REQ
1	0	1	0	0	1	0	ALL_REQ
0	1	1	0	1	0	1	RFU
1	0	0	x	x	x	x	Undefined Value
1	1	1	1	x	x	x	Undefined Value
All other values							RFU

Requirements 18: Short Frame Commands Undefined Value Coding

Listen Mode	
6.5.1.1	A received Undefined Value MAY change processing of Commands using Short Frames listed in Table 6 of the Listener.

Requirements 19: NFC-A Command Set

Poll and Listen Mode	
6.5.1.2	Commands and Responses SHALL be transmitted in frames, as defined in Table 5. Commands SHALL be transmitted using a value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ equal to 1. Responses SHALL be transmitted using a value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ equal to $\mathbf{D}_{POLL \rightarrow LISTEN}$.

6.6 ALL_REQ and SENS_REQ

The ALL_REQ and SENS_REQ Commands are sent by the Poller to probe the Operating Field for Listeners configured for NFC-A Technology.

6.6.1 ALL_REQ Command

Table 7 shows the format of the ALL_REQ Command.

Table 7: ALL_REQ Format

b7	b6	b5	b4	b3	b2	b1	Meaning
1	0	1	0	0	1	0	52h = ALL_REQ

Requirements 20: ALL_REQ Command

Poll Mode		Listen Mode	
6.6.1.1	The Poller SHALL send the ALL_REQ Command formatted and coded as specified in Table 7.	6.6.1.2	The Listener SHALL be ready to receive an ALL_REQ Command that has the format and coding specified in Table 7.

6.6.2 SENS_REQ Command

The format of the SENS_REQ Command is specified in Table 8.

Table 8: SENS_REQ Format

b7	b6	b5	b4	b3	b2	b1	Meaning
0	1	0	0	1	1	0	26h = SENS_REQ

Requirements 21: SENS_REQ Command

Poll Mode		Listen Mode	
6.6.2.1	The Poller SHALL send the SENS_REQ Command formatted and coded as specified in Table 8.	6.6.2.2	The Listener SHALL be ready to receive a SENS_REQ Command that has the format and coding specified in Table 8.

6.6.3 SENS_RES Response

Table 9 specifies the format of the SENS_RES Response.

Table 9: SENS_RES Response Format

Byte 1	Byte 2
Anticollision Information	Platform Information

In response to an ALL_REQ and SENS_REQ Command from the Poller, a Listener returns a SENS_RES Response with a length of 2 bytes, depending on its State. (See [ACTIVITY] for more details.) Table 10 and Table 11 specify the coding of the SENS_RES Response.

Anticollision Information

Byte 1 of the SENS_RES Response codes the NFCID1 size and the Bit frame SDD.

Table 10: Byte 1 of SENS_RES (Anticollision Information)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							NFCID1 size: single (4 bytes)
0	1							NFCID1 size: double (7 bytes)
1	0							NFCID1 size: triple (10 bytes)
1	1							RFU
		0						RFU
			1	0	0	0	0	Bit frame SDD
			0	1	0	0	0	Bit frame SDD
			0	0	1	0	0	Bit frame SDD
			0	0	0	1	0	Bit frame SDD
			0	0	0	0	1	Bit frame SDD
			0	0	0	0	0	Bit frame SDD
		All other values						RFU

NOTE In [ISO/IEC_18092] the NFCID1 is specified to be only 4 bytes long. Requirement 6.6.3.4 deviates from the restrictions of [ISO/IEC_18092].

Platform Information

Byte 2 of the SENS_RES Response encoding.

Table 11: Byte 2 of SENS_RES (Platform Information)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				1	1	0	0	
		All other values						

Requirements 22: SENS_RES Response

Poll Mode		Listen Mode	
6.6.3.1	The Poller SHALL be ready to receive a SENS_RES Response that has the format specified in Table 9.	6.6.3.2	The Listener SHALL send the SENS_RES Response formatted and coded as specified in Table 9.
6.6.3.3	The Poller SHALL accept any NFCID1 size according to Table 10.	6.6.3.4	The Listener SHALL set its NFCID1 size according to Table 10.
6.6.3.5	The Poller SHALL accept all values (including RFU values) in the SENS_RES Response (as defined in Table 10 and Table 11), and the Poller SHALL continue normal operation.	6.6.3.6	The Listener SHALL set its Bit frame SDD according to Table 10.
6.6.3.7		6.6.3.8	.
6.6.3.9	The Poller SHALL treat a SENS_RES Response as a Syntax Error if b5-b1 of Byte 1 are set to 00000b, but b4-b1 of Byte 2 are set to a value different from 1100b.		

NOTE Requirement 6.6.3.9 extends [ISO/IEC_18092].

6.7 SDD_REQ

The SDD_REQ Command is used to obtain the NFCID1 of a Listener and to detect whether more than one device of the same Technology is in the Operating Field of the Poller. Furthermore, the SDD_REQ Command is used for collision resolution if there is more than one Listener in the Operating Field. [ACTIVITY] describes the collision resolution mechanism.

6.7.1 SDD_REQ Command

Table 12 specifies how the SDD_REQ Command is coded.

Table 12: SDD_REQ Command Format

Byte 1	Byte 2	n data bits
SEL_CMD	SEL_PAR	data bit 1 ... data bit n

The SEL_CMD byte indicates the cascade level (CL) of the NFCID1 that is requested by the Poller and is coded as shown in Table 13.

Table 13: SEL_CMD Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	0	0	1	0	0	1	1	93h: SDD_REQ CL1
1	0	0	1	0	1	0	1	95h: SDD_REQ CL2
1	0	0	1	0	1	1	1	97h: SDD_REQ CL3
All other values								Not Allowed

The SEL_PAR byte indicates the length of the SDD_REQ Command, including the SEL_CMD and SEL_PAR bytes, and, therefore, it also indicates the number of data bits following the SEL_PAR byte.

The upper 4 bits are called “byte count” and specify the integer part of the number of data bits transmitted by the Poller (including SEL_CMD and SEL_PAR) divided by 8. Consequently, the minimum value of “Byte count” is 2. Table 14 shows the byte count format.

The lower 4 bits are called “bit count” and specify the number of data bits transmitted by the Poller modulo 8. Table 15 shows the bit count format.

For example, if SEL_PAR equals 20h, then no data bits are following. If SEL_PAR equals 35h, then 13 data bits are following.

Table 14: SEL_PAR (Upper 4 Bits) Format

b8	b7	b6	b5	Meaning
0	0	1	0	Byte count = 2
0	0	1	1	Byte count = 3
0	1	0	0	Byte count = 4
0	1	0	1	Byte count = 5
0	1	1	0	Byte count = 6
0	1	1	1	Byte count = 7

Table 15: SEL_PAR (Lower 4 Bits) Format

b4	b3	b2	b1	Meaning
0	0	0	0	Bit count = 0
0	0	0	1	Bit count = 1
0	0	1	0	Bit count = 2
0	0	1	1	Bit count = 3
0	1	0	0	Bit count = 4
0	1	0	1	Bit count = 5
0	1	1	0	Bit count = 6
0	1	1	1	Bit count = 7

The data bits following the SEL_PAR contain the first n bits of an NFCID1 in the cascade level, as specified by the SEL_CMD byte. The value of n is calculated by the value of SEL_PAR. The SDD_REQ Command contains a maximum of 32 data bits, excluding the SEL_CMD and the SEL_PAR.

A Listener that receives the SDD_REQ Command compares its NFCID1 with the data bits received. If the first n bits of its NFCID1 in the respective cascade level are equal to the n data bits of the SDD_REQ Command, then this Listener responds to the SDD_REQ Command. Otherwise the Listener ignores the SDD_REQ Command.

Requirements 23: SDD_REQ Command Format

Poll Mode		Listen Mode	
6.7.1.1	The Poller SHALL send the SDD_REQ Command formatted as specified in Table 12.	6.7.1.2	The Listener SHALL be ready to receive an SDD_REQ Command that has the format specified in Table 12.
6.7.1.3	The Poller SHALL set SEL_CMD to a value that is compliant with Table 13.	6.7.1.4	The Listener SHALL be ready to receive a SEL_CMD value that is compliant with Table 13. The Listener SHALL treat a SEL_CMD value not compliant with Table 13 as a Syntax Error.
6.7.1.5	If the Poller does not support collision resolution, then the following applies: • The Poller SHALL set SEL_PAR to a value that is compliant with the length of the SDD_REQ Command. The SEL_PAR value SHALL be 20h. If the Poller supports collision resolution, then the following applies: • The Poller SHALL set SEL_PAR to a value that is compliant with the length of the SDD_REQ Command defined in Table 14 and Table 15. • If the Poller sets the SEL_PAR Byte count defined in Table 14 to 6, the SEL_PAR Bit count defined in Table 15 SHALL be set to 0.	6.7.1.6	The Listener SHALL be ready to receive a SEL_PAR value that is compliant with the length of the SDD_REQ Command. The Listener SHALL be ready to receive a minimum SEL_PAR value of 20h. For collision resolution the Listener SHALL be ready to receive SEL_PAR, as defined in Table 14 and Table 15. The Listener SHALL treat as a Syntax Error any SEL_PAR value that is not allowed. If the SEL_PAR Byte count is 6 and the SEL_PAR Bit count is not equal to 0, the Listener MAY treat the SDD_REQ Command as a Syntax Error.

6.7.2 SDD_RES Response

In response to an SDD_REQ Command with a SEL_PAR value equal to 20h, all Listeners in the Operating Field transmit the requested cascade level of their NFCID1 (NFCID1 CLn, with n=1, 2 or 3). The NFCID1 of a Listener consists of 4, 7 or 10 bytes. The length of the Response that contains a complete NFCID1 cascade level (i.e., NFCID1 CL1, or NFCID1 CL2, or NFCID1 CL3) is always 5 bytes. The coding of the Response depends on the value of the SEL_CMD byte and the size of the NFCID1. Table 16 specifies how to code the SDD_RES Response that contains a complete NFCID1 cascade level.

Table 16: SDD_RES Response (NFCID1 CLn + BCC)

SEL_CMD	NFCID1 Size	SDD_RES Response
93h	4	CL1: **nfcid1**$_0$ **nfcid1**$_1$ **nfcid1**$_2$ **nfcid1**$_3$ BCC
93h	> 4	CL1: CT **nfcid1**$_0$ **nfcid1**$_1$ **nfcid1**$_2$ BCC
95h	7	CL2: **nfcid1**$_3$ **nfcid1**$_4$ **nfcid1**$_5$ **nfcid1**$_6$ BCC
95h	> 7	CL2: CT **nfcid1**$_3$ **nfcid1**$_4$ **nfcid1**$_5$ BCC
97h	10	CL3: **nfcid1**$_6$ **nfcid1**$_7$ **nfcid1**$_8$ **nfcid1**$_9$ BCC

- **nfcid1**$_n$ is the n^{th} byte of the complete NFCID1, with **nfcid1**$_0$ being the most significant byte.
- CT is the cascade tag.
- BCC (the NFCID1 CLn check byte for NFC-A) is an exclusive-OR over the first 4 bytes of the SDD_RES Response.

Table 17 specifies the content of nfcid1$_0$ in the case of a single size NFCID1.

Table 17: nfcid1$_0$ for Single-size NFCID1

nfcid1$_0$	**Description**
08h	**nfcid1**$_1$ to **nfcid1**$_3$ are dynamically generated
x0h to x7h x9h to xEh 18h, 28h, 38h, 48h, 58h, 68h, 78h, 98h, A8h, B8h, C8h, D8h, E8h	Fixed diversified number (the assignment of this number is out of scope of this specification)
xFh	Fixed number, non-unique
F8h	RFU

Requirements 24: Handling of NFCID1$_0$ values for Single-size NFCID1

Poll Mode	
6.7.2.1	The Poller SHALL treat all values (including RFU values) of the NFCID1$_0$ field as valid NFCID1 values.

Requirements 25: NFCID1

Listen Mode	
6.7.2.2	The NFCID1 can be dynamically generated by the NFC Forum Device. If it is dynamically generated by the NFC Forum Device, the length of the NFCID1 SHALL be limited to 4 bytes.
6.7.2.3	A dynamically generated NFCID1 SHALL be generated whenever the NFC Forum Device enters the **`IDLE`** State of the Listen Mode state machine (specified in [ACTIVITY]) from any of the following States: **`NO_REMOTE_FIELD`**, **`ATR_READY_A`**, **`ATR_READY_F`**, **`TARGET_A`**, **`TARGET_F`**.
6.7.2.4	The **nfcid1**$_0$ for a single-size NFCID1 SHALL be coded as specified in Table 17.
6.7.2.5	The Listener SHALL set **nfcid1**$_0$ of a single-size NFCID1 and **nfcid1**$_3$ of a double-size NFCID1 to a value not equal to 88h.

Requirements 26: SDD_RES Response (Complete and Incomplete NFCID1) Format

Poll Mode		Listen Mode	
6.7.2.6	The Poller SHALL be ready to receive an SDD_RES Response that has the format specified in Table 16.	6.7.2.7	The Listener SHALL send the SDD_RES Response, depending on the received SEL_CMD byte, formatted as specified in Table 16.
6.7.2.8	If the Poller finds that the CT value in SDD_RES is inconsistent with the NFCID1 length, it MAY treat this value as a Syntax Error.	6.7.2.9	The Listener SHALL set CT to a value of 88h.

Requirements 27: SDD_RES Response (Complete NFCID1 Only) Format

Poll Mode		Listen Mode	
6.7.2.10	The Poller SHALL verify the BCC included in the SDD_RES Response only when it receives the SDD_RES Response without any collisions. The Poller SHALL treat an incorrect BCC as a Transmission Error.	6.7.2.11	The Listener SHALL calculate a BCC as an exclusive-or over the first 4 bytes of the SDD_RES Response (NFCID1 CLn).

In response to an SDD_REQ Command with a SEL_PAR value different from 20h, only the Listeners in the Operating Field respond to the Command that fulfills the following requirement:

- The first n bits of the NFCID1 in the respective cascade level are equal to the n data bits of the SDD_REQ Command that follows the SEL_PAR.

Otherwise the SDD_REQ Command is ignored by the Listener.

Listeners that fulfill this requirement return the remaining bits of the requested cascade level of their NFCID1 (NFCID1 CLn, with n=1, 2, or 3).

Figure 7 shows an example. The Poller transmits an SDD_REQ Command with its SEL_CMD set to 93h, indicating the request for NFCID1 in cascade level 1. Its SEL_PAR, set to 36h, indicates that 14 data bits will follow (1 byte + 6 bits). Listeners that have an NFCID1 with its first 14 bits equal to the 14 data bits sent in the SDD_REQ Command respond by sending the 26 remaining bits of their NFCID1.

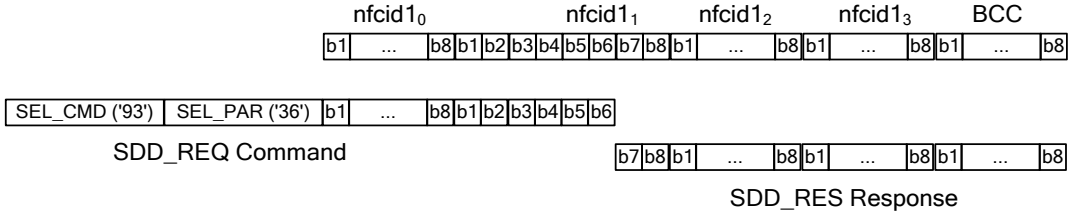


Figure 7: Example: SDD_REQ with 14 Data Bits of NFCID1 CL1 (4 Byte Size) and SDD_RES

Requirements 28: SDD_RES Response (Incomplete NFCID1 Only) Format

Poll Mode		Listen Mode	
6.7.2.12	The Poller SHALL be ready to receive an SDD_RES Response that contains (40 - n) data bits, where n is the number of data bits following the SEL_PAR in the SDD_REQ Command that was previously sent. The Poller SHALL treat an SDD_RES Response that contains a different number of data bits as a Syntax Error.	6.7.2.13	If the first n bits of the NFC Forum Device's NFCID1 in the respective cascade level are equal to the n data bits following the SEL_PAR in the preceding SDD_REQ Command, then the Listener SHALL return the remaining (40 - n bits of its NFCID1 in the respective cascade level. Otherwise the Listener SHALL ignore the SDD_REQ Command.
6.7.2.14	The Poller SHALL verify the BCC included in the SDD_RES Response only when it receives the SDD_RES Response without any collisions. Verification data SHALL constitute the first 4 bytes of the NFCID1 CLn (i.e., the concatenation of the data bits following the SEL_PAR in the preceding SDD_REQ Command and the SDD_RES Response, excluding the BCC). The Poller SHALL treat an incorrect BCC as a Transmission Error.	6.7.2.15	The Listener SHALL calculate a BCC as the exclusive-or over the first 4 bytes of the NFCID1 CLn (from which a part is returned in the SDD_RES Response).

6.8 SEL_REQ

The SEL_REQ Command is used to select the Listener by means of its NFCID1.

6.8.1 SEL_REQ Command

Table 18 specifies the code for the SEL_REQ Command.

Table 18: SEL_REQ Command Format

Byte 1	Byte 2	Bytes 3 – 6	Byte 7
SEL_CMD	70h	NFCID1 CLn	BCC

Requirements 29: SEL_REQ Command

Poll Mode		Listen Mode	
6.8.1.1	The Poller SHALL send the SEL_REQ Command formatted as specified in Table 18.	6.8.1.2	The Listener SHALL be ready to receive a SEL_REQ Command that has the format specified in Table 18.

The SEL_CMD byte is coded as shown in Table 13.

BCC is a checksum calculated as the exclusive-or over the 4 bytes of the NFCID1 CLn.

Table 19 specifies the format of NFCID1 CLn.

Table 19: NFCID1 CLn Format

SEL_CMD	NFCID1 Size	NFCID1 CLn
93h	4	CL1: **nfcid1**$_0$ **nfcid1**$_1$ **nfcid1**$_2$ **nfcid1**$_3$
93h	> 4	CL1: CT **nfcid1**$_0$ **nfcid1**$_1$ **nfcid1**$_2$
95h	7	CL2: **nfcid1**$_3$ **nfcid1**$_4$ **nfcid1**$_5$ **nfcid1**$_6$
95h	> 7	CL2: CT **nfcid1**$_3$ **nfcid1**$_4$ **nfcid1**$_5$
97h	10	CL3: **nfcid1**$_6$ **nfcid1**$_7$ **nfcid1**$_8$ **nfcid1**$_9$

Parameter **nfcid1**$_n$ indicates the n^{th} byte of the complete NFCID1, with **nfcid1**$_0$ being the most significant byte.

Requirements 30: NFCID1 CLn Format

Poll Mode		Listen Mode	
6.8.1.3	The Poller SHALL format the NFCID1 CLn of the SEL_REQ Command according to Table 19.	6.8.1.4	The Listener SHALL be ready to accept the NFCID1 CLn format of the SEL_REQ Command according to Table 19.

6.8.2 SEL_RES Response

The Listener transmits the SEL_RES Response in reply to a SEL_REQ Command. The length of the SEL_RES Response is one byte. Table 20 specifies the format of the SEL_RES Response.

Table 20: SEL_RES Response Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x								Undefined Value
	x	x						If b3 is set to 1b: • Undefined Value. If b3 is set to 0b: • 00b: Configured for Type 2 Tag Platform • 01b: Configured for Type 4A Tag Platform • 10b: Configured for the NFC-DEP Protocol • 11b: Configured for the NFC-DEP Protocol and Type 4A Tag Platform.
			x	x				Undefined Value
					x			Cascade bit • 1b: NFCID1 not complete • 0b: NFCID1 complete
						x	x	Undefined Value

NOTE The definition of SEL_RES in Table 20 extends the definition of SEL_RES in [ISO/IEC_18092].

Requirements 31: SEL_RES Response

Poll Mode		Listen Mode	
6.8.2.1	The Poller SHALL be ready to receive a SEL_RES Response that has the format specified in Table 20.	6.8.2.2	The Listener SHALL send the SEL_RES Response formatted as specified in Table 20.
6.8.2.3	The Poller SHALL treat a SEL_RES Response with b3 set to 0b and b7-b6 set to 00b as coming from a device that implements the Type 2 Tag Platform, and SHALL act as specified in [ACTIVITY].		
6.8.2.4	The Poller SHALL treat a SEL_RES Response in which b3 is 0b and b7-b6 are 01b as coming from a Listener configured for the Type 4A Tag Platform, and SHALL act as specified in [ACTIVITY].	6.8.2.5	To indicate support for the Type 4A Tag Platform, the Listener SHALL set b7-b6 of the SEL_RES Response to 01b in the last cascade level (i.e., when the NFCID1 is complete).
6.8.2.6	The Poller SHALL treat a SEL_RES Response in which b3 is 0b and b7-b6 are 10b as coming from a Listener configured for the NFC-DEP Protocol, and SHALL act as specified in [ACTIVITY].	6.8.2.7	To indicate support for the NFC-DEP Protocol, the Listener SHALL set b7-b6 of the SEL_RES Response to 10b in the last cascade level (i.e., when the NFCID1 is complete).
6.8.2.8	The Poller SHALL treat a SEL_RES Response in which b3 is 0b and b7-b6 are 11b as coming from a Listener configured for both the Type 4A Tag Platform and the NFC-DEP Protocol, and SHALL act as specified in [ACTIVITY].	6.8.2.9	To indicate support for both the Type 4A Tag Platform and the NFC-DEP Protocol, the Listener SHALL set b7-b6 of the SEL_RES Response to 11b in the last cascade level (i.e., when the NFCID1 is complete).

NOTE A Type 2 Tag Platform will set b7-b6 of the SEL_RES Response to 00b in the last cascade level (i.e., when the NFCID1 is complete).

6.9 SLP_REQ

The SLP_REQ Command is used to put the Listener in the **`SLEEP`** State (see [ACTIVITY] for the Listen Mode state machine).

6.9.1 SLP_REQ Command

The SLP_REQ Command consists of 2 bytes. Table 21 specifies the format of the SLP_REQ Command.

Table 21: SLP_REQ Command Format

Byte 1	Byte 2
50h	00h

Requirements 32: SLP_REQ Command

Poll Mode		Listen Mode	
6.9.1.1	The Poller SHALL send the SLP_REQ Command formatted and coded as specified in Table 21.	6.9.1.2	The Listener SHALL be ready to receive a SLP_REQ Command that has the format and coding specified in Table 21.

6.9.2 SLP_REQ Response

The Listener does not respond to an SLP_REQ Command.

Requirements 33: SLP_REQ Response

Poll Mode		Listen Mode	
6.9.2.1	The Poller SHALL always treat the SLP_REQ Command as acknowledged by the Listener.	6.9.2.2	The Listener SHALL NOT respond to an SLP_REQ Command.

6.10 Timing Requirements

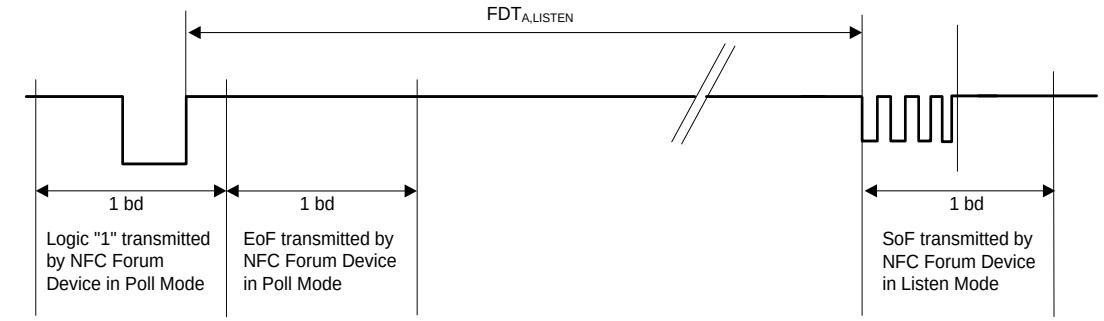
This section specifies the requirements for the guard time and Frame Delay Times for NFC-A Technology.

NOTE The term “Frame Response Time” used in [ISO/IEC_18092] is referred to as “Frame Delay Time” in this document.

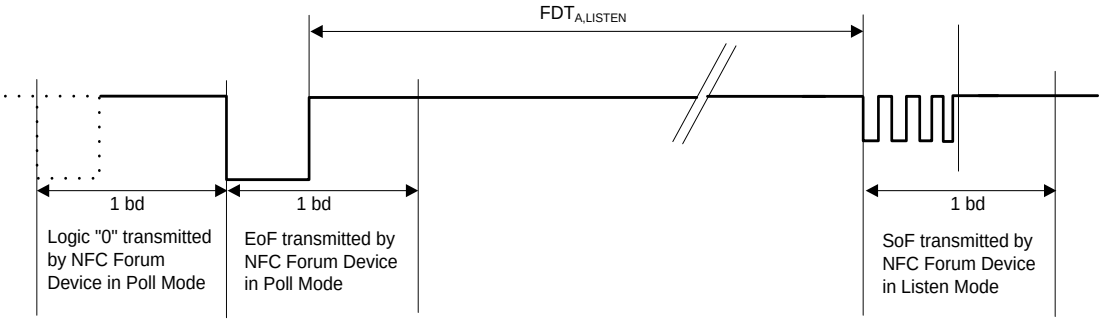
6.10.1 Frame Delay Time Poll→Listen

The Frame Delay Time in the Poll→Listen direction ($\mathbf{FDT}_{A,LISTEN}$) is the time between the end of transmission of a Poll Frame and the end of transmission of a Listen Frame. The amount of time between the end of the minimum time interval $\mathbf{FDT}_{A,LISTEN,MIN}$ and the end of the maximum time interval $\mathbf{FDT}_{A,LISTEN,MAX}$ defines the time period during which a Listen Frame is allowed to be sent by a Listener in response to a Poll Frame of a Poller.

For NFC-A Technology, $\mathbf{FDT}_{A,LISTEN}$ depends on the logic value of the last bit before the EoF transmitted by the Poller, as illustrated in Figure 8.



(a) Last bit before EoF is Logic "1"



(b) Last bit before EoF is Logic "0"

Figure 8: $FDT_{A,LISTEN}$

Table 22 shows the values that $\mathbf{FDT}_{A,LISTEN}$ can take, depending on the logic value of the last data bit transmitted by the Poller. The value of n (see Table 23) is an integer and depends on the Command type transmitted in a Poll Frame. Section 6.5 defines the Commands.

Table 22: $FDT_{A,LISTEN}$ and Logic State of Last Data Bit

Logic State	$FDT_{A,LISTEN}$
"0"	n bd + $20/\mathbf{f}_C$
"1"	n bd + $84/\mathbf{f}_C$

Table 23: Value of n and Command Type

Command Type	n
ALL_REQ SENS_REQ SDD_REQ SEL_REQ	$\mathbf{n}_{A,MIN}$
All other Commands	$\geq \mathbf{n}_{A,MIN}$

$\mathbf{FDT}_{A,LISTEN,MIN}$ follows the equations in Table 22 with a specific value of n = $\mathbf{n}_{A,MIN}$. Appendix B.2 lists the value of $\mathbf{n}_{A,MIN}$.

If the Poll Frame contains an ALL_REQ, SENS_REQ, SDD_REQ, or SEL_REQ Command, then $\mathbf{FDT}_{A,LISTEN,MAX}$ is equal to $\mathbf{FDT}_{A,LISTEN,MIN}$. In this case the Listen Frame is sent by the Listener at a specific point of time.

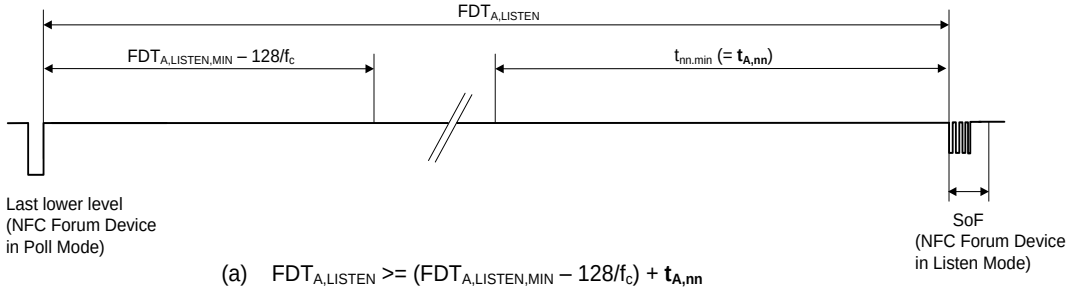
Requirements 34: $\mathbf{FDT}_{A,LISTEN}$

Poll Mode		Listen Mode	
6.10.1.1	Following the end of a Poll Frame, the Poller SHALL be ready to receive the start of a Listen Frame at a time aligned to the grid (as defined in Figure 8, Table 22 and Table 23) with a tolerance of $-1/\mathbf{f}_C$ to $\mathbf{FDT}_{A,DELTA} + 1/\mathbf{f}_C$. Appendix B.2 defines $\mathbf{FDT}_{A,DELTA}$.	6.10.1.2	Following the receipt of a Poll Frame, the Listener SHALL align the first modulation edge within the start bit of a Listen Frame to the grid (as defined in Figure 8, Table 22 and Table 23) with a tolerance of 0 to $\mathbf{FDT}_{A,DELTA}$. Appendix B.2 defines $\mathbf{FDT}_{A,DELTA}$.
6.10.1.3	For the Commands ALL_REQ, SENS_REQ, SDD_REQ and SEL_REQ, the Poller SHALL treat as a Timeout Error the receipt of a Listen Frame at a time after $\mathbf{FDT}_{A,LISTEN,MIN}$ (i.e., $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FDT}_{A,LISTEN,MIN}$).	6.10.1.4	For the Commands ALL_REQ, SENS_REQ, SDD_REQ and SEL_REQ, the Listener SHALL always respond exactly at $\mathbf{FDT}_{A,LISTEN,MIN}$, as defined in Table 22 and Table 23 (i.e., $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FDT}_{A,LISTEN,MIN}$).

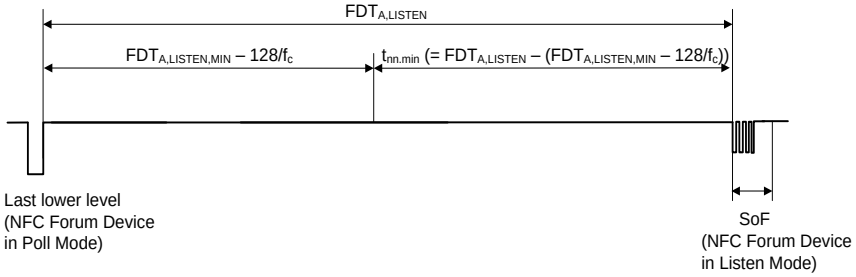
Requirements 35: $FDT_{A,LISTEN,MIN}$

Poll Mode	
6.10.1.5	Following the end of a Poll Frame, the Poller SHALL ignore any response during the time interval $\mathbf{FDT}_{A,LISTEN,MIN} - 128/\mathbf{f}_C$.

The Listener is not allowed to produce any detectable disturbance during the minimum time period $\mathbf{t}_{A,nn,min}$ before it sends a Response.



(a) $FDT_{A,LISTEN} >= (FDT_{A,LISTEN,MIN} - 128/f_c) + \mathbf{t_{A,nn}}$



(b) $FDT_{A,LISTEN} < (FDT_{A,LISTEN,MIN} - 128/f_c) + \mathbf{t_{A,nn}}$

Figure 9: NFC-A $t_{A,nn,min}$

Requirements 36: NFC-A $t_{A,nn,min}$

Listen Mode	
6.10.1.6	The Listener SHALL NOT produce any detectable disturbance during a period of at least $\mathbf{t}_{A,nn,min}$, defined as the minimum of $\mathbf{FDT}_{A,LISTEN} - (\mathbf{FDT}_{A,LISTEN,MIN} - 128/\mathbf{f}_C)$ and $\mathbf{t}_{A,nn}$, before it sends a Response. The value of $\mathbf{t}_{A,nn,min}$ SHALL be measured before the start of the SoF of the Listen Frame, as shown in Figure 9. Appendix B.2 lists the value of $\mathbf{t}_{A,nn}$.

NOTE The exact quantity of what constitutes ‘any detectable disturbance’ is not defined by this version of the specification.

6.10.2 Frame Delay Time Poll→Poll

The Frame Delay Time Poll→Poll ($\mathbf{FDT}_{A,PP}$) is the time interval between two consecutive Poll Frames during which no Listen Frame is received. If no Listen Frame is received within the expected receive window, the minimum value $\mathbf{FDT}_{A,PP,MIN}$ defines the time a Poller waits before it sends a new Poll Frame after the end of a previous Poll Frame.

If the previous Poll Frame contains an ALL_REQ, SENS_REQ, SDD_REQ, or SEL_REQ Command, then $\mathbf{FDT}_{A,PP,MIN}$ is equal to $\mathbf{FDT}_{A,POLL,MIN}$. If the previous Poll Frame contains an SLP_REQ Command, then $\mathbf{FDT}_{A,PP,MIN}$ is equal to $\mathbf{FDT}_{A,PP,SLP\_REQ,MIN}$.

No maximum Frame Delay Time Poll→Poll value ($\mathbf{FDT}_{A,PP,MAX}$) is defined.

Requirements 37: $\mathbf{FDT}_{A,PP,MIN}$

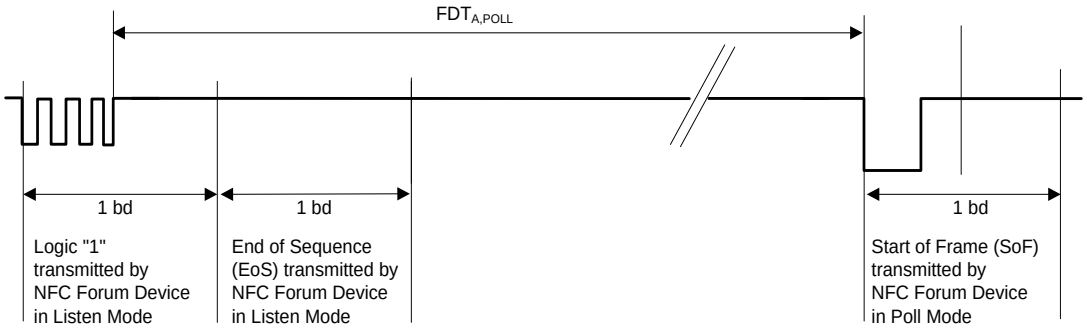
Poll Mode		Listen Mode	
6.10.2.1	Following the end of an NFC-A Poll Frame, unless an NFC-A Listen Frame is received, the Poller SHALL wait at least for $\mathbf{FDT}_{A,PP,MIN}$ before transmitting the SoF of a new Command. If an NFC-A Listen Frame is received during $\mathbf{FDT}_{A,PP,MIN}$, the Poller MAY terminate the wait for Frame Delay Time Poll→Poll.	6.10.2.2	The Listener SHALL be ready to receive the SoF of a new Command no later than $\mathbf{FDT}_{A,PP,MIN}$ after the end of an NFC-A Poll Frame if it does not send an NFC-A Listen Frame in response. If it has not sent a Response and the SoF of a new Command is received before $\mathbf{FDT}_{A,PP,MIN}$, then the Listener MAY treat this Command as a Transmission Error.

NOTE Whether or not the Poller terminates the wait for Frame Delay Time Poll→Poll, the requirements for Frame Delay Time Listen→Poll in Section 6.10.3 still apply.

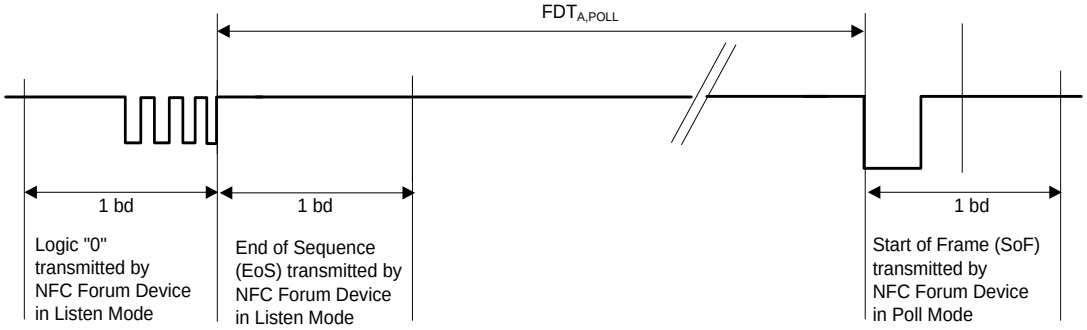
6.10.3 Frame Delay Time Listen→Poll

The Frame Delay Time Listen→Poll ($\mathbf{FDT}_{A,POLL}$) is the time interval between a Listen Frame and a Poll Frame. The minimum time interval $\mathbf{FDT}_{A,POLL,MIN}$ defines the time a Poller waits before it sends a new Poll Frame after receipt of a Listen Frame. No maximum value Frame Delay Listen→Poll ($\mathbf{FDT}_{A,POLL,MAX}$) is defined.

For NFC-A Technology the definition of $\mathbf{FDT}_{A,POLL}$ depends on the logic value of the last data bit of the Listen Frame transmitted by the NFC Forum Device, as shown in Figure 10. However, $\mathbf{FDT}_{A,POLL}$ is not restricted to certain discrete values similar to $\mathbf{FDT}_{A,LISTEN}$.



(a) Last transmitted data bit is Logic "1"



(b) Last transmitted data bit is Logic "0"

Figure 10: $FDT_{A,POLL}$

Requirements 38: $FDT_{A,POLL,MIN}$

Poll Mode		Listen Mode	
6.10.3.1	Following the end of a Listen Frame, the Poller SHALL wait at least for **$FDT_{A,POLL,MIN}$** before transmitting the start of a new Poll Frame. Appendix B.2 lists the value of **$FDT_{A,POLL,MIN}$**.	6.10.3.2	The Listener SHALL be ready to receive the start of a new Poll Frame no later than **$FDT_{A,POLL,MIN}$** after the end of a Listen Frame, and, if the start of a new Poll Frame is received before **$FDT_{A,POLL,MIN}$**, then the Listener MAY treat this Poll Frame as a Transmission Error.

A specific value for **$FDT_{A,POLL,MIN}$** is defined for the re-activation of the Listener following a DSL_RES Response (see Section 17.9.2), an RLS_RES Response (see Section 17.10.2), or an S(DESELECT) Response (see Section 16.2.7).

Requirements 39: $FDT_{A,REACTIVATION}$

Poll Mode		Listen Mode	
6.10.3.3	Following the end of a DSL_RES Response, an RLS_RES Response, or an S(DESELECT) Response, the Poller SHALL wait at least for **$FDT_{A,REACTIVATION}$** before transmitting the start of a new Poll Frame. Appendix B.2 lists the value of **$FDT_{A,REACTIVATION}$**.	6.10.3.4	The Listener SHALL be ready to receive the start of a new Poll Frame no later than **$FDT_{A,REACTIVATION}$** after the end of a DSL_RES Response, an RLS_RES Response, or an S(DESELECT) Response. If the start of a new Poll Frame is received before **$FDT_{A,REACTIVATION}$**, then the Listener MAY treat this Poll Frame as a Transmission Error.

6.10.4 Guard Time

This section specifies the guard time of an Unmodulated Carrier after which the Listener is ready to receive an ALL_REQ or SENS_REQ Command.

Requirements 40: NFC-A Guard Time

Listen Mode	
6.10.4.1	When a Listener is exposed to an Unmodulated Carrier (see [ANALOG]), the Listener SHALL be ready to receive an ALL_REQ or SENS_REQ Command after a guard time **GT_A**. Appendix B.2 lists the value of **GT_A**.

7 NFC-B Technology

This section specifies the NFC-B Technology-related features of the NFC Forum Device.

This section is derived from [ISO/IEC_14443].

7.1 Sequence Format

This section describes the sequence format for NFC-B Technology.

7.1.1 Poll→Listen Modulation

In Poll Mode the analog signal is modulated using Non-Return to Zero, L for level (NRZ-L) coding based on the ASK 10% modulation principle. [ANALOG] contains details about the Modulation Index. Figure 11 illustrates NRZ-L coding.

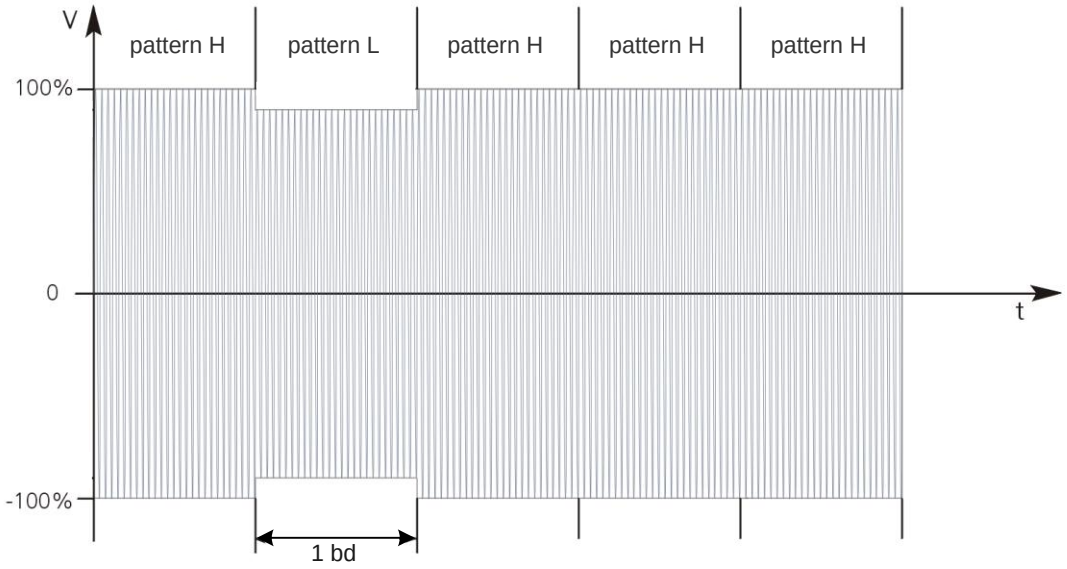


Figure 11: NRZ-L Coding

Using the modulation principle of NRZ-L coding with ASK modulation, the carrier amplitude is varied in order to define two particular patterns: L and H.

Requirements 41: NFC-B Signal Patterns Poll→Listen

Poll Mode		Listen Mode	
7.1.1.1	The Poller SHALL build pattern L and pattern H as follows: • Pattern L: carrier low (modulation applied) for the full bd • Pattern H: carrier high (no modulation applied) for the full bd.	7.1.1.2	The Listener SHALL read pattern L and pattern H as follows: • If the carrier is low (modulation applied) for the full bd, then the Listener SHALL read this pattern L. • If the carrier is high (no modulation applied) for the full bd, then the Listener SHALL read this as pattern H. • All other patterns SHALL be treated as invalid patterns.

NOTE The processing of invalid patterns is not defined by this version of the specification.

7.1.2 Listen→Poll Modulation

In Listen Mode the analog signal is modulated using NRZ-L with Binary Phase Shift Keying (BPSK) modulation (see Figure 12).

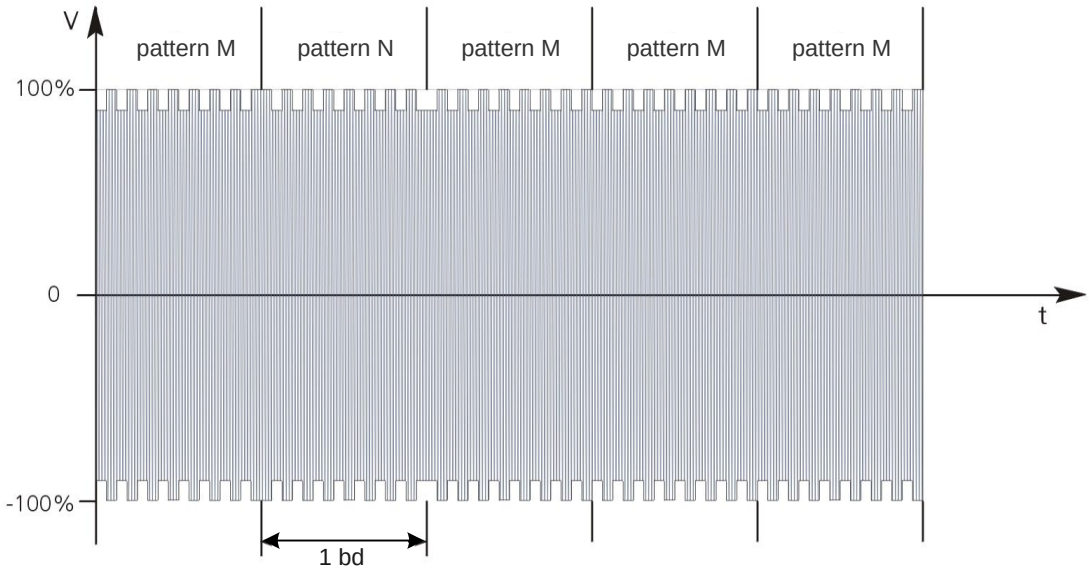


Figure 12: NRZ-L Coding with BPSK

NRZ-L with BPSK modulation uses a phase shift (180°) of the subcarrier to define two particular patterns: N and M.

Requirements 42: NFC-B Signal Patterns Listen→Poll

Poll Mode		Listen Mode	
7.1.2.1	The Poller SHALL read a subcarrier with phase Ø0 for the full bd as pattern M. If there is a phase transition, then this marks the beginning of the pattern.	7.1.2.2	The Listener SHALL build the following as pattern M: • A subcarrier with phase Ø0 for the full bd. If this requires a phase transition, the phase transition SHALL be at the beginning of the pattern.
7.1.2.3	The Poller SHALL read a subcarrier with phase Ø180° for the full bd as pattern N. If there is a phase transition, then this marks the beginning of the pattern. All patterns different from pattern M and pattern N SHALL be treated as invalid patterns.	7.1.2.4	The Listener SHALL build the following as pattern N: • A subcarrier with phase Ø180° for the full bd. If this requires a phase transition, the phase transition SHALL be at the beginning of the pattern.

NOTE The processing of invalid patterns is not defined by this version of the specification.

7.1.3 Synchronization

Figure 13 and Figure 14 illustrate the different parameters used for NFC-B Technology signal synchronization and related signal timing parameters.

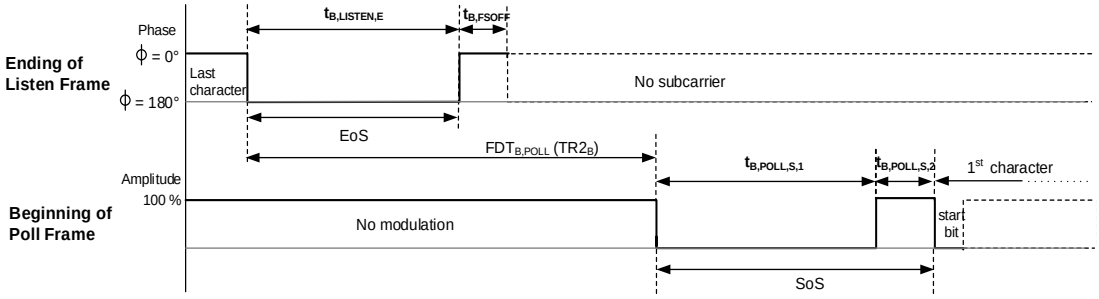


Figure 13: Synchronization and Timing Parameters between a Listen Frame and a Poll Frame

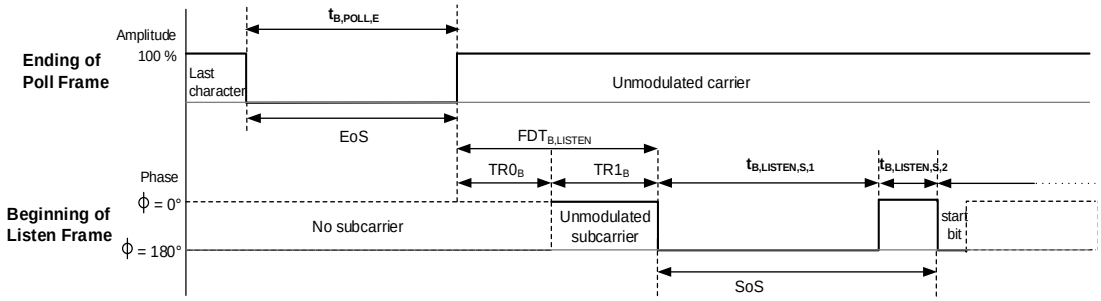


Figure 14: Synchronization and Timing Parameters between a Poll Frame and a Listen Frame

For the demodulator the beginning of a signal is indicated by the SoS.

Requirements 43: NFC-B Synchronization Poll→Listen

Poll Mode		Listen Mode	
7.1.3.1	The Poller SHALL code the SoS as follows: • SoS: $\mathbf{t}_{B,POLL,S,1}$ with carrier low (modulation applied), followed by $\mathbf{t}_{B,POLL,S,2}$ with carrier high (no modulation applied). Appendix B.3 lists the values of $\mathbf{t}_{B,POLL,S,1}$ and $\mathbf{t}_{B,POLL,S,2}$.	7.1.3.2	The Listener SHALL decode the SoS as follows: • If the carrier is low (modulation applied) for $\mathbf{t}_{B,POLL,S,1}$, followed by $\mathbf{t}_{B,POLL,S,2}$ with carrier high (no modulation applied), then the Listener SHALL decode that combination as an SoS.

Requirements 44: NFC-B Synchronization Listen→Poll

Poll Mode		Listen Mode	
7.1.3.3	For establishing phase reference Ø0, the Poller SHALL proceed as follows: • After any Command from the NFC Forum Device, it SHALL ignore any subcarrier generated by the Listener during a time $\mathbf{TR0}_{B,MIN}$. • The subcarrier, as detected during a time interval between $\mathbf{TR1}_{B,MIN}$ and $\mathbf{TR1}_{B,MAX}$, SHALL be taken as phase reference Ø0. If a phase transition is detected before $\mathbf{TR1}_{B,MIN}$, then the Poller MAY treat the transition as a Transmission Error. If at $\mathbf{TR1}_{B,MAX}$ no phase transition is detected, then the Poller MAY treat the lack of phase transition as a Transmission Error. Appendix B.3 lists the value of $\mathbf{TR1}_{B,MAX}$.	7.1.3.4	For establishing phase reference Ø0, the Listener SHALL proceed as follows: • After any Command from the Poller, a minimum guard time $\mathbf{TR0}_{B,MIN}$ applies during which the Listener SHALL NOT generate a subcarrier. $\mathbf{TR0}_{B,MIN}$ SHALL take the value $\mathbf{TR0}_{B,MIN,DEFAULT}$, unless changed by a higher layer Command. • Following the guard time $\mathbf{TR0}_{B}$ the Listener SHALL then generate a subcarrier with no phase transition for a synchronization time $\mathbf{TR1}_{B}$. This establishes a subcarrier phase reference Ø0. $\mathbf{TR1}_{B,MIN}$ SHALL take the value $\mathbf{TR1}_{B,MIN,DEFAULT}$, unless changed by a higher layer Command. Appendix B.3 lists the values of $\mathbf{TR0}_{B,MIN,DEFAULT}$ and $\mathbf{TR1}_{B,MIN,DEFAULT}$.
7.1.3.5	If, after the synchronization time $\mathbf{TR1}_{B}$, the Poller detects: • A subcarrier phase transition Ø0 to Ø0+180° • Followed by a subcarrier with phase Ø0+180° for $\mathbf{t}_{B,LISTEN,S,1}$ • Followed by a subcarrier phase transition Ø0+180° to Ø0 • Followed by the subcarrier with phase Ø0 for $\mathbf{t}_{B,LISTEN,S,2}$ then the Poller SHALL decode this combination as an SoS.	7.1.3.6	After the synchronization time $\mathbf{TR1}_{B}$, the Listener SHALL code the SoS as follows: • Subcarrier phase transition Ø0 to Ø0+180° • Followed by a subcarrier with phase Ø0+180° for $\mathbf{t}_{B,LISTEN,S,1}$ • Followed by a subcarrier phase transition Ø0+180° to Ø0 • Followed by subcarrier with phase Ø0 for $\mathbf{t}_{B,LISTEN,S,2}$. Appendix B.3 lists the values of $\mathbf{t}_{B,LISTEN,S,1}$ and $\mathbf{t}_{B,LISTEN,S,2}$.

7.1.4 Pattern Synchronization

Patterns are grouped in a pattern group that is 10 bd long. The separation between two pattern groups is defined as the Extra Guard Time (EGT). The Listen Device ensures that the established phase reference Ø0 is still valid at the end of the EGT, so that the next phase transition still occurs at nominal positions of rising or falling edges of the subcarrier.

Requirements 45: NFC-B Pattern Group Separation

Poll and Listen Mode	
7.1.4.1	The time between two consecutive pattern groups sent by the Poller to the Listener SHALL be in the range between $\mathbf{EGT}_{B,POLL,MIN}$ and $\mathbf{EGT}_{B,POLL,MAX}$. Appendix B.3 lists the values of $\mathbf{EGT}_{B,POLL,MIN}$ and $\mathbf{EGT}_{B,POLL,MAX}$.
7.1.4.2	The time between two consecutive pattern groups sent by the Listener to the Poller SHALL be in the range between $\mathbf{EGT}_{B,LISTEN,MIN}$ and $\mathbf{EGT}_{B,LISTEN,MAX}$. Appendix B.3 lists the values of $\mathbf{EGT}_{B,LISTEN,MIN}$ and $\mathbf{EGT}_{B,LISTEN,MAX}$.

The separation between two patterns within a pattern group occurs according to the following requirements.

Requirements 46: NFC-B Pattern Boundaries

Poll Mode		Listen Mode	
7.1.4.3	For Poll→Listen communication, pattern boundaries within a pattern group SHALL occur during the interval between time n bd – $8/\mathbf{f}_C$ and time n bd + $8/\mathbf{f}_C$, in which n is the number of pattern boundaries after the start pattern falling edge ($1 \leq n \leq 9$).	7.1.4.4	For Listen→Poll communication, pattern boundaries within a pattern group SHALL only occur at nominal positions of rising or falling edges of the subcarrier: n bd.

7.1.5 De-synchronization

The end of a sequence is indicated by the EoS.

Requirements 47: NFC-B De-synchronization Poll→Listen

Poll Mode		Listen Mode	
7.1.5.1	The Poller SHALL code the EoS as follows: • EoS: $\mathbf{t}_{B,POLL,E}$ with carrier low (modulation applied), followed by a transition to carrier high. Appendix B.3 lists the value of $\mathbf{t}_{B,POLL,E}$.	7.1.5.2	The Listener SHALL decode the EoS as follows: • If the carrier is low (modulation applied) for $\mathbf{t}_{B,POLL,E}$, followed by a transition to carrier high (no modulation applied), then the Listener SHALL decode this combination as EoS.

Requirements 48: NFC-B De-synchronization Listen→Poll

Poll Mode		Listen Mode	
7.1.5.3	If the Poller detects: • A subcarrier phase transition Ø0 to Ø0+180° • Followed by the subcarrier with phase Ø0+180° for $\mathbf{t}_{B,LISTEN,E}$ • Followed by a subcarrier phase transition Ø0+180° to Ø, then the Poller SHALL decode this combination as an EoS.	7.1.5.4	The Listener SHALL code the following as an EoS: • A subcarrier phase transition Ø0 to Ø0+180° • Followed by a subcarrier with phase Ø0+180° for $\mathbf{t}_{B,LISTEN,E}$ • Followed by a subcarrier phase transition Ø0+180° to Ø0. Appendix B.3 lists the value of $\mathbf{t}_{B,LISTEN,E}$.
7.1.5.5	The Poller SHALL allow for the fact that the subcarrier might be maintained on for a time $\mathbf{t}_{B,FSOFF}$ after EoS by the Listener. After EoS, if the Listener maintains the subcarrier on for a time greater than the maximum value of $\mathbf{t}_{B,FSOFF}$, then the Poller MAY treat this subcarrier as a Transmission Error.	7.1.5.6	After EoS, the Listener SHALL maintain the subcarrier on for a time $\mathbf{t}_{B,FSOFF}$ and then SHALL turn the subcarrier off. Appendix B.3 lists the value of $\mathbf{t}_{B,FSOFF}$.

7.2 Bit Level Coding

7.2.1 Poll→Listen Coding Scheme

Patterns L and H are used to code the digital Logic “0” and Logic “1”.

Requirements 49: NFC-B Bit Level Coding Poll→Listen

Poll Mode		Listen Mode	
7.2.1.1	The Poller SHALL code Logic “0” and Logic “1” as follows: • Logic “0”: pattern L • Logic “1”: pattern H.	7.2.1.2	The Listener SHALL decode Logic “0” and Logic “1” as follows: • If the Listener detects pattern L, then it SHALL decode this as Logic “0”. • If the Listener detects pattern H, then it SHALL decode this as Logic “1”.

7.2.2 Listen→Poll Coding Scheme

Patterns N and M are used to code the digital Logic “0” and Logic “1”.

Requirements 50: NFC-B Bit Level Coding Listen→Poll

Poll Mode		Listen Mode	
7.2.2.1	The Poller SHALL decode Logic “0” and Logic “1” as follows: • If the Poller detects pattern N, then it SHALL decode this pattern as Logic “0”. • If the Poller detects pattern M, then it SHALL decode this pattern as Logic “1”.	7.2.2.2	The Listener SHALL code Logic “0” and Logic “1” as follows: • Logic “0”: pattern N • Logic “1”: pattern M.

7.3 Frame Format

To transmit data, the NFC Forum Device configured for NFC-B Technology uses frames that are built from characters. This section defines the formats of characters and frames.

A character consists of a Logic “0” start bit, a Logic “1” stop bit and 8 data bits. The stop bit, start bit and the individual data bits each have a length of one bd. Figure 15 shows the character format.

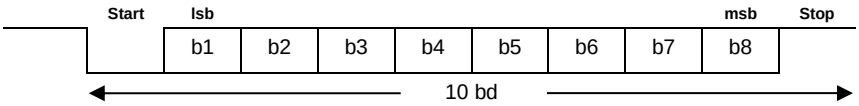


Figure 15: NFC-B Character Format

Requirements 51: NFC-B Character Format

Poll and Listen Mode	
7.3.1.1	A character SHALL consist of a start bit (Logic “0”), 8 data bits and a stop bit (Logic “1”). The stop bit, start bit and the individual data bits SHALL each be one bd long.

Characters are sent as frames, as shown in Figure 16.

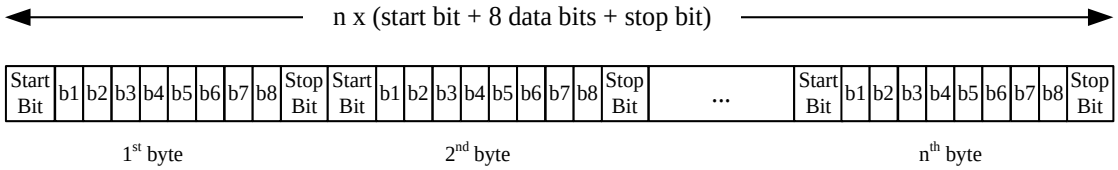


Figure 16: NFC-B Frame Format

Requirements 52: NFC-B Frame Format

Poll and Listen Mode	
7.3.1.2	A frame SHALL consist of characters that are aligned as shown in Figure 16.

NOTE The frame format defined in Figure 16 deviates from [ISO/IEC_14443]. SoF and EoF (as they are named in [ISO/IEC_14443]) are defined by this specification as SoS and EoS at the sequence layer.

7.4 Data and Payload Format

Data transmitted in an NFC-B frame are organized according to the following structure. The data consist of the payload and of an EoD (SoD is not used).

The EoD contains a 2-byte checksum referred to as CRC_B. The CRC_B is a function of k data bits, where k is a multiple of 8. The input for the CRC_B calculation is the payload.

Figure 17 illustrates the NFC-B data and payload format.

Data					
Payload (Command or Response)				EoD	
Byte 1	Byte 2	…	Byte n	CRC_B1	CRC_B2

Figure 17: NFC-B Data and Payload Format

Requirements 53: NFC-B Data and Payload Format

Poll and Listen Mode	
7.4.1.1	Data SHALL be transmitted in frames, as defined in Section 7.3.
7.4.1.2	The payload SHALL be followed by an EoD at the position indicated in Figure 17. The EoD SHALL contain a CRC_B.
7.4.1.3	The CRC_B SHALL be calculated as defined in [ISO/IEC_13239]. The initial register content SHALL be all ones (FFFFh). CRC_B1 is the LSB and CRC_B2 is the MSB. The input for the CRC_B calculation SHALL be the payload.
7.4.1.4	The NFC Forum Device SHALL compare the received CRC_B with the CRC_B calculated according to Requirement 7.4.1.3. If the two are different, then the NFC Forum Device SHALL treat the received data as a Transmission Error.

NOTE Transmission Errors on the Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

7.5 Command Set

The payload exchanged between NFC Forum Devices consists of Commands and Responses. Table 24 lists the Commands that are supported by the NFC Forum Device configured for NFC-B Technology. For each Command the corresponding Response is indicated.

Table 24: NFC-B Command Set

Command	Response
ALLB_REQ, SENSB_REQ	SENSB_RES
SLOT_MARKER	SENSB_RES
SLPB_REQ	SLPB_RES

NOTE The Commands and Responses listed in Table 24 have different names in [ISO/IEC_14443]: ALLB_REQ refers to WUPB, SENSB_REQ refers to REQB, SENSB_RES refers to ATQB, and SLPB_REQ and SLPB_RES refer to HLTB and its Response.

Requirements 54: NFC-B Command Set

Poll and Listen Mode	
7.5.1.1	Commands and Responses SHALL be transmitted as the payload of the data and payload format defined in Section 7.4. Commands SHALL be transmitted using a value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ equal to 1. Responses SHALL be transmitted using a value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ equal to $\mathbf{D}_{POLL \rightarrow LISTEN}$.

7.6 ALLB_REQ and SENSB_REQ

The ALLB_REQ and SENSB_REQ Commands are used by a Poller to probe the Operating Field for Listeners.

7.6.1 ALLB_REQ and SENSB_REQ Command

Table 25 defines the format of the ALLB_REQ and SENSB_REQ Commands.

Table 25: ALLB_REQ and SENSB_REQ Command Formats

Byte 1	Byte 2	Byte 3
05h	AFI	PARAM

Requirements 55: ALLB_REQ and SENSB_REQ Command Formats

Poll Mode		Listen Mode	
7.6.1.1	The Poller SHALL send ALLB_REQ and SENSB_REQ Commands formatted as specified in Table 25.	7.6.1.2	The Listener SHALL be ready to receive ALLB_REQ and SENSB_REQ Commands that has the format specified in Table 25.

The components of this Command are defined as follows.

AFI

The AFI indicates the application family being selected.

Requirements 56: AFI

Poll Mode		Listen Mode	
7.6.1.3	The AFI SHALL be set to 00h. This selects all application families.	7.6.1.4	The Listener SHALL support ALLB_REQ and SENSB_REQ with AFI equal to 00h. It MAY support an ALLB_REQ and a SENSB_REQ with AFI different from 00h.

PARAM

Table 26 specifies the format of PARAM.

Table 26: Format of PARAM Byte Included in ALLB_REQ and SENSB_REQ Command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			x					0b: Extended SENSB_RES not supported 1b: Extended SENSB_RES supported
				x				0b: SENSB_REQ 1b: ALLB_REQ
					x	x	x	NI

Requirements 57: PARAM Byte Format

Poll Mode		Listen Mode	
7.6.1.5	The Poller SHALL format the PARAM byte according to Table 26.	7.6.1.6	The Listener SHALL be ready to receive the PARAM byte that has the format specified in Table 26.

The Number of slots Identifier (NI) is used by the anticollision scheme defined in [ISO/IEC_14443]. The anticollision scheme is based on the definition of time slots in which Listeners are invited to respond. Table 27 specifies the coding of NI.

Table 27: NI Coding

b3	b2	b1	Meaning
0	0	0	Number of Slots=1
0	0	1	Number of Slots=2
0	1	0	Number of Slots=4
0	1	1	Number of Slots=8
1	0	0	Number of Slots=16
1	0	1	RFU
1	1	x	RFU

Requirements 58: NI

Poll Mode		Listen Mode	
7.6.1.7	If the Poller does not support collision resolution, it SHALL set NI to 000b. If the Poller supports collision resolution, it SHALL set NI to a value specified in Table 27.	7.6.1.8	The Listener SHALL be ready to receive a value of NI set to 000b. For collision resolution it SHALL be ready to receive a value of NI, as specified in Table 27.

Requirements 59: Listen Mode Handling of NI with an RFU Value

Listen Mode	
7.6.1.9	A received NI value of RFU SHALL be treated by the Listener as NI=100b (Number of Slots=16).

Bit 5 indicates whether the Poller supports the extended SENSB_RES byte. The extended SENSB_RES byte is an optional byte included in the SENSB_RES Response, coding the Start-up Frame Guard time Integer (SFGI) used by the Listener.

Requirements 60: Support for Extended SENSB_RES

Poll Mode		Listen Mode	
7.6.1.10	The Poller SHALL be ready to receive a SENSB_RES Response without an extended SENSB_RES byte included. If the Poller set b5 of its SENSB_REQ to 1b, it SHALL be ready to receive a SENSB_RES Response with or without an extended SENSB_RES byte included.	7.6.1.11	When answering an ALLB_REQ or SENSB_REQ Command with bit b5 set to 0b, indicating that the extended SENSB_RES is not supported, the Listener SHALL NOT include the extended SENSB_RES byte in its SENSB_RES Response. When answering an ALLB_REQ or SENSB_REQ Command with b5 set to 1b, indicating that the extended SENSB_RES is supported, the Listener MAY include the extended SENSB_RES byte in its SENSB_RES Response.

NOTE The Poller is allowed to receive a SENSB_RES Response with an extended SENSB_RES byte included even if it set b5 of its SENSB_REQ to 0b.

7.6.2 SENSB_RES Response

Table 28 defines the SENSB_RES format.

Table 28: SENSB_RES Response Format

Byte 1	Byte 2 – 5	Byte 6 – 9	Byte 10 – 12 or 13
50h	NFCID0	Application Data	Protocol Info

Byte 13 (i.e., the extended SENSB_RES byte of the SENSB_RES Response) is optional. If it is not used, then the SENSB_RES consists of only 12 bytes.

Requirements 61: SENSB_RES Response

Poll Mode		Listen Mode	
7.6.2.1	The Poller SHALL be ready to receive a SENSB_RES Response that has the format specified in Table 28.	7.6.2.2	The Listener SHALL send the SENSB_RES Response with the 12-byte format specified in Table 28.
	The Poller SHALL be ready to receive the SENSB_RES Response with the 13-byte format specified in Table 28 if the support of extended SENSB_RES is indicated in the PARAM byte of the SENSB_REQ or ALLB_REQ Command.		The Listener MAY send the extended SENSB_RES Response with the 13-byte format specified in Table 28 if the Poller has indicated support of the extended SENSB_RES in the PARAM byte of its SENSB_REQ or ALLB_REQ Command.

NFCID0

The NFCID0 is used to distinguish Listeners during the Collision Resolution Activity.

Requirements 62: NFCID0 in SENSB_RES

Poll Mode		Listen Mode	
7.6.2.3	The Poller SHALL accept any NFCID0.	7.6.2.4	The NFCID0 SHALL have a length of 4 bytes. The value of the NFCID0 SHALL be fixed or dynamically generated by the NFC Forum Device.
		7.6.2.5	A dynamically generated NFCID0 SHALL be generated only on state transition from **`NO_REMOTE_FIELD`** to **`IDLE`** State (see [ACTIVITY] for the Listen Mode state machine).

Application Data

The Application Data field is used to inform the Poller which applications are installed on the Listener.

Requirements 63: Application Data Field

Poll Mode	
7.6.2.6	The Poller SHALL accept any value in an Application Data field.

The Application Data is defined according to the ADC in the Protocol Info, which defines whether the CRC_B compressing method (described below) or a coding not defined by this version of the specification is used.

When the CRC_B compressing coding is used, the Application Data field contains the information shown in Table 29.

Table 29: Application Data Format

Byte 1	Byte 2 and 3	Byte 4
AFI	CRC_B (AID)	Number of applications

Requirements 64: Application Data Format

Poll Mode		Listen Mode	
7.6.2.7	If the ADC of Protocol Info is equal to 01b, the Poller SHALL be ready to receive Application Data that has the format specified in Table 29. Otherwise the Poller SHALL be ready to receive any value in the Application Data field (bytes 6 – 9 of the SENSB_RES Response).	7.6.2.8	If the ADC of Protocol Info is set to 01b, the Listener SHALL format the Application Data according to Table 29. Otherwise the Listener MAY set the Application Data field to any value.

- AFI

  When there is a single application in the Listener, AFI provides the family of the application (see AFI coding defined in [ISO/IEC_14443]).

  When there are multiple applications in the Listener, AFI provides the family of the application described in CRC_B (AID).

- CRC_B (AID)

  CRC_B(AID) is the result of the calculation of CRC_B of the AID ([ISO/IEC_7816-4] contains the definition of AID) of an application in the NFC Form Device in Listen Mode, matching the AFI given in the SENSB_REQ/ALLB_REQ Command.

- Number of applications

  Indicates the presence of other applications in the Listener.

  The most significant half-byte value provides the number of applications corresponding to the AFI given in Application Data, with ‘0’ meaning no application and ‘F’ meaning 15 applications or more.

  The least significant half-byte value provides the total number of applications in the Listener, with ‘0’ meaning no application and ‘F’ meaning 15 applications or more.

Protocol Info

The Protocol Info field indicates the parameters supported by the Listener, as defined in Table 30.

Table 30: Protocol Info Format

Byte 1	Byte 2		Byte 3			Byte 4 (optional)	
Bit_Rate_Capability (8 bits)	FSCI (4 bits)	Protocol_Type (4 bits)	FWI (4 bits)	ADC (2 bits)	FO (2 bits)	SFGI (4 bits)	RFU (4 bits)

Requirements 65: Protocol Info Format

Poll Mode		Listen Mode	
7.6.2.9	The Poller SHALL be ready to receive Protocol Info that has the format specified in Table 30.	7.6.2.10	The Listener SHALL format the Protocol Info according to Table 30.

- Bit_Rate_Capability

  Table 31 specifies the bit rates supported by the Listener. Bits b7 to b5 code the bit rate capability of the Listener for the direction from Listener to Poller ($\mathbf{D}_{LISTEN \rightarrow POLL}$). The value 000b corresponds with $\mathbf{D}_{LISTEN \rightarrow POLL} = 1$. The bits b3 through b1 code the bit rate capability of the Listener for the direction from Poller to Listener ($\mathbf{D}_{POLL \rightarrow LISTEN}$). The value 000b corresponds with $\mathbf{D}_{POLL \rightarrow LISTEN} = 1$.

Table 31: Bit Rates Supported by the Listener

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x								If b8 = 1b, then only the same bit rate divisor for both directions is supported ($\mathbf{D}_{LISTEN \rightarrow POLL} = \mathbf{D}_{POLL \rightarrow LISTEN}$). If b8 = 0b, then a different bit rate divisor for each direction is supported.
	x							$\mathbf{D}_{LISTEN \rightarrow POLL}$ = 8 supported, if bit is set to 1b.
		x						$\mathbf{D}_{LISTEN \rightarrow POLL}$ =4 supported, if bit is set to 1b.
			x					$\mathbf{D}_{LISTEN \rightarrow POLL}$ =2 supported, if bit is set to 1b.
				0				RFU
					x			$\mathbf{D}_{POLL \rightarrow LISTEN}$ =8 supported, if bit is set to 1b.
						x		$\mathbf{D}_{POLL \rightarrow LISTEN}$ =4 supported, if bit is set to 1b.
							x	$\mathbf{D}_{POLL \rightarrow LISTEN}$ =2 supported, if bit is set to 1b.

Requirements 66: Bit Rates Supported by the Listener

Poll Mode		Listen Mode	
7.6.2.11	The Poller SHALL be ready to receive any Bit_Rate_Capability value (see Table 31).	7.6.2.13	The Listener MAY set bits b7 through b5 and b3 through b1 of the Bit_Rate_Capability to any value specified in Table 31.
7.6.2.12	The Poller SHALL support a bit rate of 106 kbits/s in both directions. The Poller MAY support higher bit rates, as indicated by Bit_Rate_Capability.		

NOTE Bit rates above 106 kbits/s for NFC-B are not defined by this version of the specification.

Requirements 67: Poll Mode Handling of b4 in Bit_Rate_Capability with Value RFU

Poll Mode	
7.6.2.14	A received b4 value of RFU SHALL be interpreted by the Poller as if b8 through b1 = 00000000b (only 106 kbits/s in both directions).

- FSCI

  The Frame Size for proximity Card Integer (FSCI) identifies the Frame Size for proximity Card (FSC), coded as indicated in Table 32. The FSC defines the maximum size of a frame (in bytes) accepted by the Listener.

Table 32: FSCI to FSC Conversion

FSCI	FSC (bytes)
0h	16
1h	24
2h	32
3h	40
4h	48
5h	64
6h	96
7h	128
8h	256
9h	512
Ah	1024
Bh	2048
Ch	4096
Dh – Fh	RFU

Requirements 68: FSC

Poll Mode		Listen Mode	
7.6.2.15	The Poller SHALL send frames with a number of data bytes less than or equal to FSC.	7.6.2.16	The Listener SHALL accept frames with a number of data bytes less than or equal to FSC. If it receives a frame with more than FSC data bytes, the Listener MAY treat the frame as a Syntax Error .
7.6.2.17	The Poller SHALL be capable of sending frames in accordance with an FSC greater than or equal to $\mathbf{FSC}_{B,MIN}$.	7.6.2.18	The FSC supported by the Listener SHALL be at least $\mathbf{FSC}_{B,MIN}$. Appendix B.3 lists the value of $\mathbf{FSC}_{B,MIN}$.

NOTE If blocks that contain more than FSC bytes are accepted according to requirement 7.6.2.16, they need to be handled correctly.

Requirements 69: FSCI

Listen Mode	
7.6.2.19	The NFC Forum Device SHALL set FSCI greater than or equal to the value corresponding to $\mathbf{FSC}_{B,MIN}$. Appendix B.3 lists the value of $\mathbf{FSC}_{B,MIN}$. Table 32 describes the FSCI to FSC conversion.

Requirements 70: Poll Mode Handling of FSCI with Value RFU

Poll Mode	
7.6.2.20	A received FSCI value of RFU SHALL be treated by the Poller as FSCI=Ch.

- Protocol_Type

  Protocol_Type encodes the Platform Type support and the minimum $TR2_B$ supported by the Listener, as shown in Table 33.

Table 33: Protocol_Type

b4	b3	b2	b1	Meaning
0				Fixed Value
	x	x		Minimum $TR2_B$
			x	Platform Type 0b: Not configured for Type 4B Tag Platform 1b: Configured for Type 4B Tag Platform; ISO-DEP Protocol supported

Requirements 71: Platform Type Supported by the Listener

Poll Mode		Listen Mode	
7.6.2.21	The Poller SHALL be ready to receive Protocol_Type that has the format specified in Table 33. The Poller SHALL treat a SENSB_RES with Protocol_Type bit b1 equal to 1b as coming from a device that implements the Type 4B Tag Platform and SHALL act as defined in [ACTIVITY]. The behavior of the receiving Protocol_Type bit b1 equal to 0b is out of scope of this specification.	7.6.2.22	The Listener SHALL set the Protocol_Type according to Table 33. If the Listener is configured for the Type 4B Tag Platform, it SHALL set bit b1 of Protocol Type to 1b, as defined in Table 33.

Protocol_Type bits b3 and b2 define Minimum $TR2_B$ supported by the Listener, as specified in Table 34.

Table 34: Minimum $TR2_B$ Coding

b3	b2	Meaning
0	0	$1792/\mathbf{f}_C$ (10 bd + $32/\mathbf{f}_S$)
0	1	$3328/\mathbf{f}_C$ (10 bd + $128/\mathbf{f}_S$)
1	0	$5376/\mathbf{f}_C$ (10 bd + $256/\mathbf{f}_S$)
1	1	$9472/\mathbf{f}_C$ (10 bd + $512/\mathbf{f}_S$)

NOTE [ANALOG] specifies $\mathbf{f}_S$.

Requirements 72: Minimum $TR2_B$ Coding

Poll Mode		Listen Mode	
7.6.2.23	The Poller SHALL set its **$TR2_{B,MIN}$** either to **$TR2_{B,MIN,DEFAULT}$** or to Minimum **$TR2_B$** , as received and indicated in Table 34 if it is larger than **$TR2_{B,MIN,DEFAULT}$** . Appendix B.3 lists the value of **$TR2_{B,MIN,DEFAULT}$** .	7.6.2.24	The Listener MAY set bits b3 and b2 to any value specified by Table 34.

NOTE Definitions in Requirements 7.6.2.23 deviate from definitions in [ISO/IEC_14443].

Requirements 73: Poll Mode Handling of b4 in Protocol_Type with Value RFU

Poll Mode	
7.6.2.25	The Poller SHALL NOT continue communicating with a Listener that sets b4 to 1b.

- FWI – Frame Waiting Time Integer (4 bits)

  The FWI assigns a set of integer values that are used to define the Frame Waiting Time (FWT). The FWT is the maximum time a Listener is can wait before it sends the Listen Frame after the end of a Poll Frame. The FWT is calculated as specified in Section 7.9.1. FWI has a value in the range from 0 to 14. The value 15 is RFU.

Requirements 74: Maximum Value of FWI

Poll Mode		Listen Mode	
7.6.2.26	The Poller SHALL support an FWI less than or equal to **$FWI_{B,MAX}$**.	7.6.2.27	The Listener SHALL set FWI less than or equal to **$FWI_{B,MAX}$**. Appendix B.3 lists the value of **$FWI_{B,MAX}$**.

Requirements 75: Poll Mode Handling of FWI Value of RFU

Poll Mode	
7.6.2.28	A received FWI value of RFU SHALL be treated by the Poller as FWI=4.

- ADC

  The ADC represents the Application Data Coding supported by the Listener and is coded as specified in Table 35.

Table 35: ADC Coding

b4	b3	Meaning
0		RFU
	x	0b: Application is not defined by this version of the specification 1b: Application is coded as described in Table 29.

Requirements 76: ADC Coding

Poll Mode		Listen Mode	
7.6.2.29	The Poller SHALL be ready to receive ADC specified in Table 35. The Poller SHALL accept any value in b3 of ADC.	7.6.2.30	The Listener SHALL set ADC according to Table 35.

- FO

  Table 36 indicates the Frame Options (FO) supported by the Listener.

  The bits b2 and b1 are used by the Listener to indicate which optional fields in the SoD of ISO-DEP block it supports. Section 16.1 specifies the SoD. The default value for b2 is 1b (Device Identification number (DID) supported). The default value for b1 is 0b (NAD not supported).

Table 36: FO Coding

b2	b1	Meaning
x		NAD supported, if bit is set to 1b.
	x	DID supported, if bit is set to 1b.

Requirements 77: FO Coding

Poll Mode		Listen Mode	
7.6.2.31	The Poller SHALL be ready to receive FO set to any value specified in Table 36.	7.6.2.32	The Listener SHALL set FO as specified in Table 36.

- SFGI – Start-up Frame Guard Time Integer (4 bits)

  The most significant nibble b8 to b5 of the optional Byte 4 contains SFGI and is used by the Listener to define the Start-up Frame Guard Time (SFGT). Section 15.8.3 defines SFGT. SFGI has a value in the range from 0 to 14. The value 15 is RFU. The default value of SFGI is 0.

Requirements 78: SFGI

Poll Mode		Listen Mode	
7.6.2.33	The Poller SHALL support an SFGI less than or equal to $\mathbf{SFGI}_{T4BT,MAX}$.	7.6.2.34	The Listener SHALL set SFGI less than or equal to $\mathbf{SFGI}_{T4BT,MAX}$. Appendix B.8 lists the value of $\mathbf{SFGI}_{T4BT,MAX}$.

NOTE The use of extended SENSB_RES and the SFGI field in the optional Byte 4 for a Listener is only meaningful when the Type 4B Tag Platform is supported – as SFGT is the Special Frame Guard Time, defined for the next Command after the ATTRIB Response (See Chapter 15).

Requirements 79: Poll Mode Handling of SFGI Value of RFU

Poll Mode	
7.6.2.35	A received SFGI value of RFU SHALL be treated by the Poller as SFGI = 0.

7.7 SLOT_MARKER

The SLOT_MARKER Command is used by a Poller during collision resolution to define the start of the Response time slot for a Listener.

7.7.1 SLOT_MARKER Command

Table 37 defines the format of the SLOT_MARKER Command.

Table 37: SLOT_MARKER Command Format

Byte 1
Apn

Requirements 80: SLOT_MARKER Command Format

Poll Mode		Listen Mode	
7.7.1.1	The Poller SHALL send the SLOT_MARKER Command formatted as specified in Table 37.	7.7.1.2	The Listener SHALL be ready to receive the SLOT_MARKER Command that has the format specified in Table 37.

Apn

The Anticollision Prefix byte (Apn) is formatted as follows:

- b4-b1 are set to 0101b.
- b8-b5 code the slot number, as specified in Table 38.

Table 38: Slot Number Coding

b8	b7	b6	b5	Meaning
0	0	0	0	Not Allowed
0	0	0	1	Slot 2
0	0	1	0	Slot 3
0	0	1	1	Slot 4
…				
1	1	1	0	Slot 15
1	1	1	1	Slot 16

Requirements 81: Slot Number Coding

Poll Mode		Listen Mode	
7.7.1.3	The Poller SHALL set the slot number as specified in Table 38.	7.7.1.4	The Listener SHALL accept the slot number coded as specified in Table 38.

7.7.2 SLOT_MARKER Response

The Response to the SLOT_MARKER Command is identical to the SENSB_RES Response, as specified in Section 7.6.2.

7.8 SLPB_REQ

The SLPB_REQ Command is used to set a Listener, identified by its NFCID0, to the **`SLEEP`** State. [ACTIVITY] specifies the Listen Mode state machine.

7.8.1 SLPB_REQ Command

Table 39 defines the format of the SLPB_REQ Command.

Table 39: SLPB_REQ Command Format

Byte 1	Byte 2 – 5
50h	NFCID0

Requirements 82: SLPB_REQ Command Format

Poll Mode		Listen Mode	
7.8.1.1	The Poller SHALL send the SLPB_REQ Command formatted as specified in Table 39.	7.8.1.2	The Listener SHALL be ready to receive the SLPB_REQ Command that has the format specified in Table 39.

NFCID0

Byte 2 through byte 5 include the NFCID0 sent by the Listener in the SENSB_RES Response.

7.8.2 SLPB_RES Response

Table 40 defines the format of the SLPB_RES Response.

Table 40: SLPB_RES Response Format

Byte 1
00h

Requirements 83: SLPB_RES Response Format

Poll Mode		Listen Mode	
7.8.2.1	The Poller SHALL be ready to receive a SLPB_RES Response that has the format specified in Table 40.	7.8.2.2	The Listener SHALL send the SLPB_RES Response formatted as specified in Table 40.

7.9 Timing Requirements

This section specifies the requirements for the guard time and Frame Delay Times for NFC-B Technology.

7.9.1 Frame Delay Time Poll→Listen

The Frame Delay Time Poll→Listen ($\mathbf{FDT}_{B,LISTEN}$) is the time between a Poll Frame and a Listen Frame, as illustrated in Figure 14. The time interval between the end of the minimum time period $\mathbf{FDT}_{B,LISTEN,MIN}$ and the end of the maximum time period $\mathbf{FDT}_{B,LISTEN,MAX}$ defines the time interval during which a Listen Frame is can be sent by a Listener in response to a Poll Frame of a Poller.

$\mathbf{FDT}_{B,LISTEN,MIN}$ is the minimum waiting time for a Listener before it sends the Listen Frame after the end of a Poll Frame. $\mathbf{FDT}_{B,LISTEN,MIN}$ is defined:

$$\mathbf{FDT}_{B,LISTEN,MIN} = \mathbf{TR0}_{B,MIN} + \mathbf{TR1}_{B,MIN} .$$

Requirements 84: $FDT_{B,LISTEN,MIN}$

Poll Mode		Listen Mode	
7.9.1.1	The Poller SHALL be ready to receive the SoS of a Listen Frame no later than $\mathbf{FDT}_{B,LISTEN,MIN}$ after the EoS of a Poll Frame.	7.9.1.2	Following the EoS of a Poll Frame, the Listener SHALL wait at least $\mathbf{FDT}_{B,LISTEN,MIN}$ before it sends the SoS of its Listen Frame.

$\mathbf{FDT}_{B,LISTEN,MAX}$ is the maximum time a Listener can wait before it sends the Listen Frame after the end of a Poll Frame (except for the SENSB_RES Response in response to SENSB_REQ, ALLB_REQ and SLOT_MARKER Commands). This parameter is also called Frame Waiting Time (FWT). The FWT is calculated using the formula:

$$FWT = (256 \times 16/\mathbf{f}_C) \times 2^{FWI},$$

where, by definition, the FWI integer value can have a maximum range from 0 to 14. In the SENSB_RES Response the Listener informs the Poller of the FWI value. Section 7.6.2 specifies the SENSB_RES Response format.

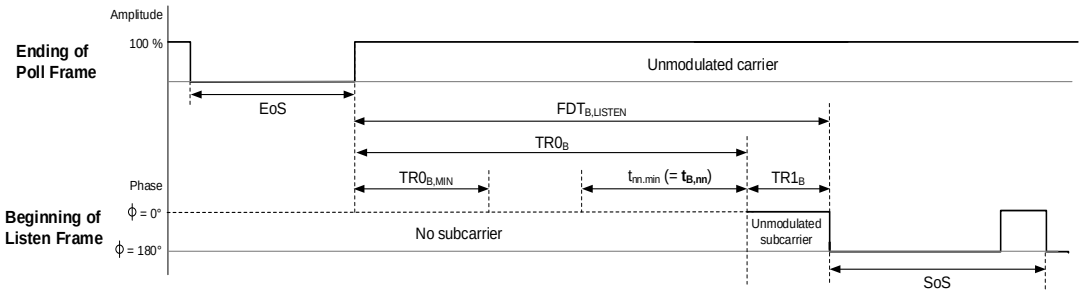
Requirements 85: Frame Waiting Time

Poll Mode		Listen Mode	
7.9.1.3	The Poller SHALL wait at least FWT + ΔFWT_B for the SoS of a Listen Frame after the EoS of a Poll Frame (except for ALLB_REQ, SENSB_REQ, and SLOT_MARKER Commands). If the Poller does not receive the SoS of a Listen Frame within the time period FWT + ΔFWT_B + $\Delta T_{B,POLL}$, then the Poller SHALL treat the lack of response as a Timeout Error (i.e., $FDT_{B,LISTEN,MAX}$ equals FWT + ΔFWT_B + $\Delta T_{B,POLL}$). During the interval between the end of the time period FWT + ΔFWT_B and the end of the time period FWT + ΔFWT_B + $\Delta T_{B,POLL}$, the Poller MAY accept the SoS of a Listen Frame or MAY generate a Timeout Error. Appendix B.3 lists the values of $\Delta T_{B,POLL}$ and ΔFWT_B.	7.9.1.4	Following the EoS of a Poll Frame, the Listener SHALL wait no longer than FWT before it sends the SoS of its Listen Frame (except for ALLB_REQ, SENSB_REQ, and SLOT_MARKER Commands; i.e., $FDT_{B,LISTEN,MAX}$ equals FWT).

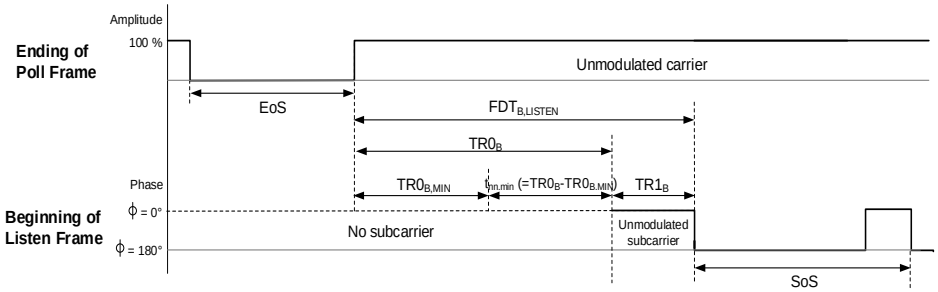
Poll Mode		Listen Mode	
7.9.1.5	In the case of the ALLB_REQ, SENSB_REQ and SLOT_MARKER Commands, the Poller SHALL wait $\mathbf{FWT}_{B,SENSB}$ for the SoS of the Listen Frame after the EoS of the Poll Frame. If the Poller does not receive the SoS of the Listen Frame within the time period $\mathbf{FWT}_{B,SENSB} + \mathbf{\Delta T}_{B,POLL}$, then the Poller SHALL treat the lack of response as a Timeout Error (i.e., $\mathbf{FDT}_{B,LISTEN,MAX}$ equals $\mathbf{FWT}_{B,SENSB} + \mathbf{\Delta T}_{B,POLL}$). During the interval between the end of the time period $\mathbf{FWT}_{B,SENSB}$ and the end of the time period $\mathbf{FWT}_{B,SENSB} + \mathbf{\Delta T}_{B,POLL}$, the Poller MAY accept the SoS of a Listen Frame or MAY generate a Timeout Error. Appendix B.3 lists the values of $\mathbf{FWT}_{B,SENSB}$ and $\mathbf{\Delta T}_{B,POLL}$.	7.9.1.6	In the case of the ALLB_REQ, SENSB_REQ and SLOT_MARKER Commands, following the EoS of the Poll Frame, the Listener SHALL generate a subcarrier no later than $\mathbf{TR0}_{B,MAX}$. Appendix B.3 lists the value of $\mathbf{TR0}_{B,MAX}$.

NOTE If the Poller does not detect the start of the subcarrier from the listen frame within $\mathbf{TR0}_{B,MAX}$, then the Poller might consider this as a Timeout Error. $\mathbf{FWT}_{B,SENSB}$ is used in 7.9.1.5 to guarantee backward compatibility, but might be replaced with $\mathbf{TR0}_{B,MAX}$ in future revisions of this specification.

The Listener cannot produce any detectable disturbance during a minimum time period $\mathbf{t}_{B,nn,min}$ before it sends a Response.



(a) $TR0_B >= TR0_{B,MIN} + t_{B,nn}$



(b) $TR0_B < TR0_{B,MIN} + t_{B,nn}$

Figure 18: NFC-B $t_{B,nn,min}$

Requirements 86: $t_{B,nn,min}$ for NFC-B

Listen Mode	
7.9.1.7	The Listener SHALL NOT produce any detectable disturbance during a time period of at least $\mathbf{t}_{B,nn,min}$, defined as the minimum of $\mathbf{TR0}_B - \mathbf{TR0}_{B,MIN}$ and $\mathbf{t}_{B,nn}$, before it sends a Response. The value of $\mathbf{t}_{B,nn,min}$ SHALL be measured before the start of $TR1_B$, as shown in Figure 18. For $\mathbf{TR0}_{B,MIN}$ the Poll Mode value SHALL be used. Appendix B.3 lists the value of $\mathbf{t}_{B,nn}$.

NOTE The exact quantity of what constitutes 'any detectable disturbance' is not defined by this version of the specification.

7.9.2 Frame Delay Time Poll→Poll

The Frame Delay Time Poll→Poll ($\mathbf{FDT}_{B,PP}$) is the time between the end of a Poll Frame and a start of the next Poll Frame without a Listen Frame in between. For all Commands with a defined Frame Waiting Time this time is equal to the Frame Waiting Time. For the Commands SENSB_REQ and ALLB_REQ this time ($\mathbf{FDT}_{B,PP}$) considers the maximum allowed response time of a Listen Frame. The minimum value $\mathbf{FDT}_{B,PP,MIN}$ defines the waiting time for a Poller before it sends a new Poll Frame after it has sent the last SENSB_REQ or ALLB_REQ Poll Frame. There is no maximum waiting time defined before it sends a new SENSB_REQ or ALLB_REQ Poll Frame.

Requirements 87: $\mathbf{FDT}_{B,PP,MIN}$

Poll Mode		Listen Mode	
7.9.2.1	Following the end of a SENSB_REQ or ALLB_REQ Poll Frame, the Poller SHALL wait at least for $\mathbf{FDT}_{B,PP,MIN}$ before transmitting the SoS of a new Command. Appendix B.3 lists the value of $\mathbf{FDT}_{B,PP,MIN.}$	7.9.2.2	The Listener SHALL be ready to receive the SoS of a new Command no later than $\mathbf{FDT}_{B,PP,MIN}$ after the end of a SENSB_REQ or ALLB_REQ Poll Frame. If the SoS of a new Command is received before $\mathbf{FDT}_{B,PP,MIN}$, then the Listener MAY treat this receipt as a Transmission Error. Appendix B.3 lists the value of $\mathbf{FDT}_{B,PP,MIN.}$
7.9.2.3	Following the end of the last SLOT_MARKER Poll Frame, the Poller SHALL wait at least $\mathbf{FDT}_{B,PP,MIN}$ before transmitting the SoS of a new Command – in case the Poller did not receive a Response to one of the SLOT_MARKER Commands. Appendix B.3 lists the value of $\mathbf{FDT}_{B,PP,MIN.}$	7.9.2.4	The Listener SHALL be ready to receive the SoS of a new Command no later than $\mathbf{FWT}_{B,SENSB}$ after the end of the last SLOT_MARKER Poll Frame.

7.9.3 Frame Delay Time Listen→Poll

The Frame Delay Time Listen→Poll ($\mathbf{FDT}_{B,POLL}$) is the time between a Listen Frame and a Poll Frame, as shown in Figure 13. The minimum value $\mathbf{FDT}_{B,POLL,MIN}$ defines the waiting time for a Poller before it sends a new Poll Frame after receipt of a Listen Frame. The maximum value $\mathbf{FDT}_{B,POLL,MAX}$ is not defined. The minimum value $\mathbf{FDT}_{B,POLL,MIN}$ is defined:

$$\mathbf{FDT}_{B,POLL,MIN} = \mathbf{TR2}_{B,MIN} .$$

$TR2_{B,MIN}$ is the configurable minimum value of $TR2_B$, as shown in Figure 13. The value of **$TR2_{B,MIN}$** that is supported by the Listener is indicated in the Protocol_Type of the SENSB_RES Response, as specified in Table 34 in Section 7.6.

Requirements 88: $FDT_{B,POLL,MIN}$

Poll Mode		Listen Mode	
7.9.3.1	Following the end of a Listen Frame, the Poller SHALL wait at least for **$FDT_{B,POLL,MIN}$** before transmitting the SoS of a new Command.	7.9.3.2	The Listener SHALL be ready to receive the SoS of a new Command no later than **$FDT_{B,POLL,MIN}$** after the end of a Listen Frame. If the SoS of a new Command is received before **$FDT_{B,POLL,MIN}$**, then the Listener MAY treat this receipt as a Transmission Error.

A specific value for **$FDT_{B,POLL,MIN}$** is defined for the reactivation of the Listener following an S(DESELECT) Response (see Section 16.2.7).

Requirements 89: $FDT_{B,REACTIVATION}$

Poll Mode		Listen Mode	
7.9.3.3	Following the end of an S(DESELECT) Response, the Poller SHALL wait at least **$FDT_{B,REACTIVATION}$** before transmitting the start of a new Poll Frame. Appendix B.3 lists the value of **$FDT_{B,REACTIVATION}$**.	7.9.3.4	The Listener SHALL be ready to receive the start of a new Poll Frame no later than **$FDT_{B,REACTIVATION}$** after the end of an S(DESELECT) Response. If the start of a new Poll Frame is received before **$FDT_{B,REACTIVATION}$**, then the Listener MAY treat this receipt as a Transmission Error.

7.9.4 Guard Time

This section specifies the guard time of an Unmodulated Carrier, after which the Listener is ready to receive an ALLB_REQ or SENSB_REQ Command.

Requirements 90: NFC-B Guard Time

Listen Mode	
7.9.4.1	When a Listener is exposed to an Unmodulated Carrier (see [ANALOG]), it SHALL be ready to receive an ALLB_REQ or SENSB_REQ Command after a guard time $\mathbf{GT}_{B}$. Appendix B.3 lists the value of $\mathbf{GT}_{B}$.

8 NFC-F Technology

This section specifies NFC-F Technology-related features of the NFC Forum Device.

This section is derived from [ISO/IEC_18092].

8.1 Sequence Format

This section describes the sequence format for NFC-F Technology.

8.1.1 Modulation

In both transmission directions, the analog signal is modulated using Manchester coding with ASK modulation. [ANALOG] contains details about the Modulation Index. Figure 19 illustrates Manchester coding.

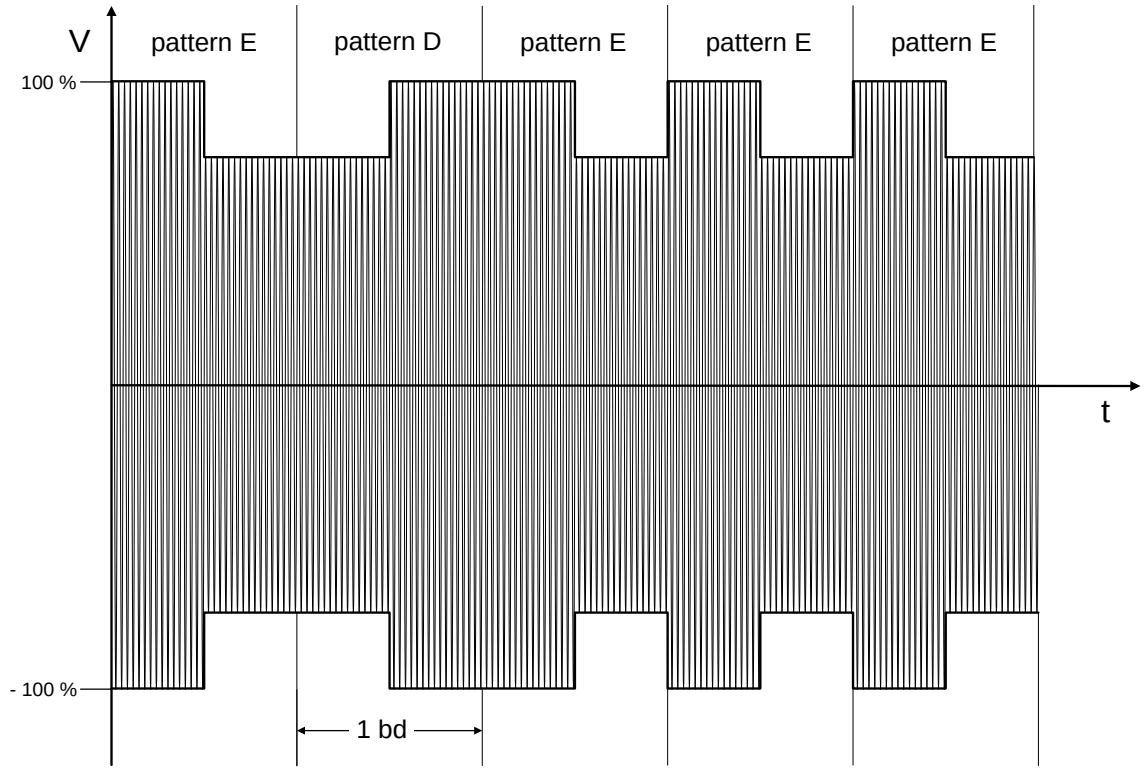


Figure 19: Manchester Coding with ASK Modulation

The modulation principle of ASK is used to modulate the carrier amplitude in a defined way between a $\mathbf{V}_a$ and a $\mathbf{V}_b$. The carrier signal not modulated is denoted as $\mathbf{V}_a$, the carrier signal when modulated is denoted as $\mathbf{V}_b$. [ANALOG] defines $\mathbf{V}_a$ and $\mathbf{V}_b$.

NOTE $\mathbf{V}_a$ (higher amplitude) and $\mathbf{V}_b$ (lower amplitude) are also applied to Listen Mode to Poll Mode in this specification, while those parameters are only defined for Poll Mode to Listen Mode in [ANALOG].

Manchester coding uses this principle to define two particular patterns: E and D.

Requirements 91: NFC-F Signal Pattern Creation

Poll and Listen Mode	
8.1.1.1	When it is transmitting, the NFC Forum Device SHALL build signal patterns E and D: • Pattern E: at the beginning of the bd, a $\mathbf{V}_a$ SHALL occur for a period of 1/2 bd. After this 1/2 bd, a transition to $\mathbf{V}_b$ SHALL occur and the $\mathbf{V}_b$ SHALL last for another period of 1/2 bd. • Pattern D: at the beginning of the bd, a $\mathbf{V}_b$ SHALL occur for a period of 1/2 bd. After this period of 1/2 bd, a transition to $\mathbf{V}_a$ SHALL occur and the $\mathbf{V}_a$ SHALL last for a period of another 1/2 bd.

Requirements 92: NFC-F Signal Pattern Reading

Poll and Listen Mode	
8.1.1.2	When it is receiving, the NFC Forum Device SHALL read signal patterns E and D as follows: • If at the beginning of a bd the NFC Forum Device detects a $\mathbf{V}_a$ that lasts for 1/2 bd, followed by a transition to $\mathbf{V}_b$ with the $\mathbf{V}_b$ lasting for 1/2 bd, the NFC Forum Device SHALL read this as pattern E. • If at the beginning of a bd the NFC Forum Device detects a $\mathbf{V}_b$ that lasts for 1/2 bd, followed by a transition to $\mathbf{V}_a$ with the $\mathbf{V}_a$ lasting for 1/2 bd, the NFC Forum Device SHALL read this as pattern D. • All other patterns SHALL be treated as invalid patterns.

NOTE The processing of invalid patterns is not defined by this version of the specification.

Requirements 93: NFC-F Transition Boundaries

Poll and Listen Mode	
8.1.1.3	In order to send signal patterns the transition boundary during 1 bd SHALL occur at nominal position bd/2.

NOTE It is appropriate to observe the SoS of an NFC-F Sequence for measurement of the transition, since the SoS contains either a series of 48 patterns D or a series of 48 patterns E.

8.1.2 Synchronization

Figure 20 illustrates the different parameters used for NFC-F Technology signal synchronization and related signal timing parameters. Manchester coding is allowed in two options according to [ISO/IEC_18092], termed “Positive Encoding” and “Negative Encoding” in this specification.

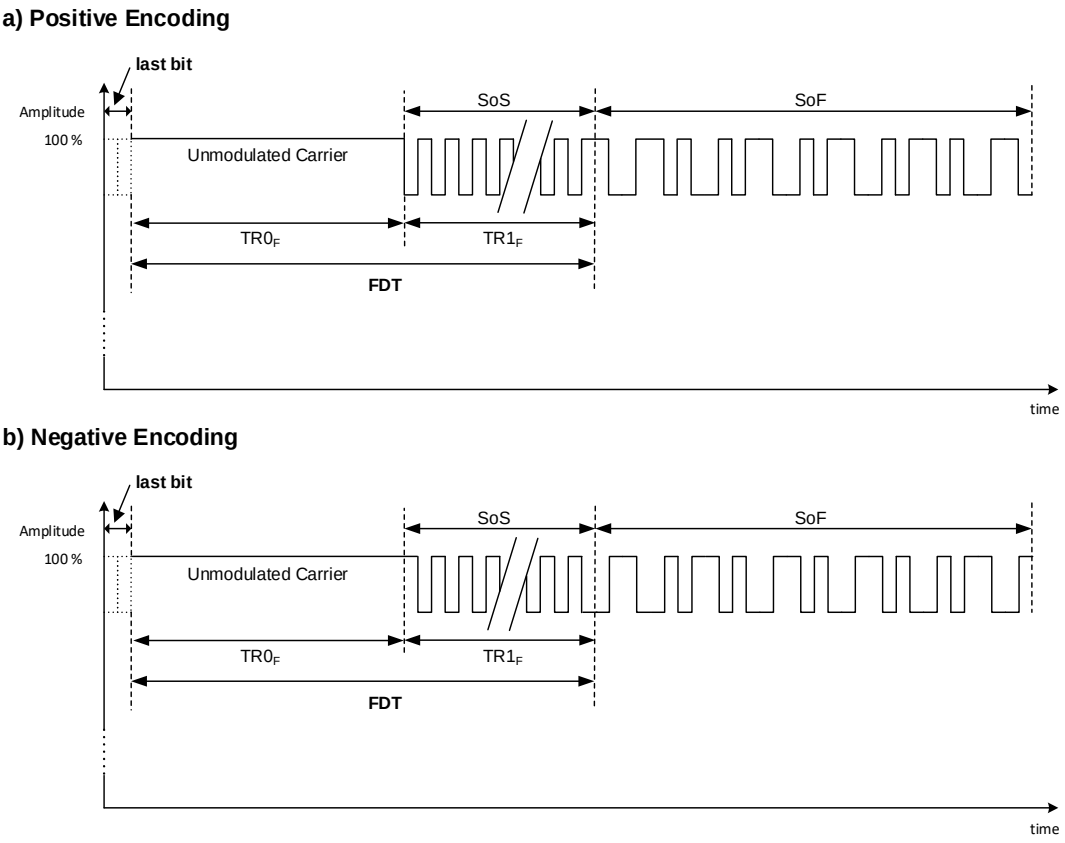


Figure 20: Signal Synchronization and Timing Parameters

For the demodulator the beginning of a signal is indicated by the SoS. The duration of the SoS is **$TR1_F$** and has a value of 48 bd.

Requirements 94: NFC-F Sequence Synchronization

Poll and Listen Mode	
8.1.2.1	When it is sending, the NFC Forum Device SHALL code SoS as a series of 48 patterns D or as a series of 48 patterns E.
8.1.2.2	When it is receiving, the NFC Forum Device SHALL be synchronized no later than a series of 48 patterns D or a series of 48 patterns E.

8.1.3 De-synchronization

The end of a signal is indicated by the content of the SoD.

Requirements 95: NFC-F De-synchronization

Poll and Listen Mode	
8.1.3.1	When it is receiving, the NFC Forum Device SHALL decode signal patterns until it has completely received the EoD and SHALL ignore any modulated signal or received data coming afterwards. The position of the EoD is indicated by the SoD (see 8.4).

NOTE NFC-F Technology de-synchronization is based on the length indicated in the SoD. Therefore NFC-F Technology does not define an EoS.

8.2 Bit Level Coding

The patterns E and D are used to code the digital Logic “0” and Logic “1”.

Requirements 96: NFC-F Bit Level Coding

Poll and Listen Mode	
8.2.1.1	If the NFC Forum Device has transmitted SoS as a series of patterns D, it SHALL code Logic “0” and Logic “1” as follows (Positive Encoding): • Logic “1”: pattern E • Logic “0”: pattern D If the NFC Forum Device has transmitted SoS as a series of patterns E, it SHALL code Logic “0” and Logic “1” as follows (Negative Encoding): • Logic “1”: pattern D • Logic “0”: pattern E

Requirements 97: NFC-F Bit Level Decoding

Poll and Listen Mode	
8.2.1.2	When it is receiving, the NFC Forum Device SHALL decode Logic “0” and Logic “1” as follows: • If the NFC Forum Device detects pattern E, then it SHALL decode this as Logic “1”. • If the NFC Forum Device detects pattern D, then it SHALL decode this as Logic “0”.

NOTE Decoding of the patterns following the SoF is defined in Section 8.3.

8.3 Frame Format

This section defines the frames and characters of the NFC Forum Device that is configured for NFC-F Technology.

A character consists of 8 data bits without start, stop, and parity bits, as shown in Figure 21. Characters are transmitted as a continuous string, with no separation in time between characters.

msb							**lsb**
b8	b7	b6	b5	b4	b3	b2	b1

Figure 21: NFC-F Character Format

A frame starts with the SoF followed by the data, as illustrated in Figure 22. The Data field consists of characters.

SoF	Data (consisting of Characters)

Figure 22: NFC-F Frame Format

Requirements 98: NFC-F Frame Format

Poll and Listen Mode	
8.3.1.1	When it is sending, an NFC Forum Device SHALL begin a frame with an SoF, followed by characters, as specified in Figure 21.
8.3.1.2	When it is sending, the NFC Forum Device SHALL code SoF as B24Dh.
8.3.1.3	When it is receiving, after the synchronization, the NFC Forum Device SHALL decode as constituting an SoF any series of Logic “0”s and Logic “1”s with a value B24Dh or 4DB2h. If the NFC Forum Device receives B24Dh, the NFC Forum Device SHALL decode all subsequent patterns as follows: • If the NFC Forum Device detects pattern E, then it SHALL decode this as Logic “1”. • If the NFC Forum Device detects pattern D, then it SHALL decode this as Logic “0” If the NFC Forum Device receives 4DB2h, the NFC Forum Device SHALL decode all subsequent patterns as follows: • If the NFC Forum Device detects pattern E, then it SHALL decode this as Logic “0”. • If the NFC Forum Device detects pattern D, then it SHALL decode this as Logic “1”.

8.4 Data and Payload Format

The Data field transmitted in an NFC-F frame has the following substructure. The included data consist of an SoD, the payload and an EoD.

The SoD contains a length byte LEN, indicating the length of the payload + 1.

The EoD contains a two-byte checksum, referred to as the CRC_F. The CRC_F is a function of k data bits. The number of bits k is a multiple of eight. The input for the CRC_F calculation is the SoD and the payload.

The NFC-F data and payload format is illustrated in Figure 23.

Data						
SoD	**Payload**				**EoD**	
LEN	Byte 1	Byte 2	…	Byte n	CRC_F1	CRC_F2

Figure 23: NFC-F Data and Payload Format

Requirements 99: NFC-F Data and Payload Format

Poll and Listen Mode	
8.4.1.1	Data SHALL be transmitted in frames that use the frame format defined in Section 8.3.
8.4.1.2	The SoD SHALL contain a length byte LEN (at the position shown in Figure 23) that has a value equal to n+1, where n indicates the number of bytes in the payload. The NFC Forum Device SHALL treat an incorrect LEN value as a Transmission Error.
8.4.1.3	The length byte LEN SHALL have a value in the range 1 to 255. When it receives a LEN with a different value, the NFC Forum Device SHALL treat that LEN as a Transmission Error.
8.4.1.4	The payload SHALL follow the SoD, as indicated in Figure 23.
8.4.1.5	The payload SHALL be followed by an EoD at the position indicated in Figure 23.
8.4.1.6	The EoD SHALL contain a CRC_F.
8.4.1.7	The CRC_F SHALL be calculated as defined in [ISO/IEC_18092]. CRC_F1 is the MSB and CRC_F2 is the LSB. The input for the CRC_F calculation SHALL be the payload.
8.4.1.8	The NFC Forum Device SHALL compare the received CRC_F with the CRC_F calculated according to Requirement 8.4.1.7. If different, then the NFC Forum Device SHALL treat the received data as a Transmission Error.

8.5 Command Set

The payload exchanged between NFC Forum Devices consists of Commands and Responses. Table 41 lists the Commands that are available to the NFC Forum Device that is configured for NFC-F Technology. For each Command the corresponding Response is indicated.

Table 41: NFC-F Command Set

Poll Mode (Command)	Listen Mode (Response)
SENSF_REQ	SENSF_RES

NOTE The Command and Response listed in Table 41 have different names in [ISO/IEC_18092]: SENSF_REQ refers to Polling Request and SENSF_RES refers to Polling Response.

Requirements 100: NFC-F Command Set

Poll and Listen Mode	
8.5.1.1	Commands and Responses SHALL be transmitted as the payload of the data and payload Format defined in Section 8.4. Commands SHALL be transmitted using a value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ equal either to 2 or to 4. Responses SHALL be transmitted using a value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ equal to $\mathbf{D}_{POLL \rightarrow LISTEN}$.

8.6 SENSF_REQ

The SENSF_REQ Command is used by a Poller to probe the Operating Field for Listeners.

NOTE Definitions in Section 8.6 deviate from definitions in [ISO/IEC_18092].

8.6.1 SENSF_REQ Command

Table 42 defines the format of the SENSF_REQ Command.

Table 42: SENSF_REQ Command Format

Byte 1	Byte 2 – 3	Byte 4	Byte 5
00h	SC	RC	TSN

Requirements 101: SENSF_REQ Command

Poll Mode		Listen Mode	
8.6.1.1	The Poller SHALL send the SENSF_REQ Command formatted as specified in Table 42.	8.6.1.2	The Listener SHALL be ready to receive a SENSF_REQ Command that has the format specified in Table 42.

Requirements 102: SENSF_REQ Bit Rates

Poll Mode		Listen Mode	
8.6.1.3	The Poller SHALL send the SENSF_REQ Command in a bit rate of either $\mathbf{f}_C/64$ or $\mathbf{f}_C/32$.	8.6.1.4	If configured as a Target, the Listener SHALL accept the SENSF_REQ Command in $\mathbf{f}_C/64$ and $\mathbf{f}_C/32$. Otherwise the Listener SHALL accept the SENSF_REQ Command in $\mathbf{f}_C/64$ and MAY accept it in $\mathbf{f}_C/32$.

The components of this Command are defined as follows:

SC

The System Code (SC) contains information regarding the NFC Forum Device to be polled (e.g., the Technology Subset).

Requirements 103: System Code (SC)

Poll Mode		Listen Mode	
8.6.1.5	If configured to poll for all devices in Listen Mode, the Poller SHALL set SC to FFFFh. If configured to poll for devices in a specific configuration, the Poller MAY set SC to a different value.	8.6.1.6	The Listener SHALL be ready to receive a SENSF_REQ Command with SC equal to FFFFh. If configured for the Type 3 Tag Platform, the Listener SHALL accept at least one different value for SC in accordance to the requirements given in [JIS_X_6319-4] for SC. In all other cases the Listener SHALL NOT send the SENSF_RES Response.

RC

The Request Code (RC) is used to retrieve additional information in the SENSF_RES Response. Table 43 specifies the RC code.

Table 43: RC Coding

Value	Meaning
00h	No System Code information requested
01h	System Code information requested
02h	Advanced protocol features supported
Other values	RFU

Requirements 104: Request Code (RC)

Poll Mode		Listen Mode	
8.6.1.7	If configured to poll for Listeners that are configured for the NFC-DEP Protocol, the Poller SHALL set RC to 00h. If configured to poll for Listeners that are configured for the Type 3 Tag Platform, the Poller SHALL set RC to 00h, 01h or 02h.	8.6.1.8	The Listener SHALL be ready to receive a SENSF_REQ Command with RC set to 00h. If configured for the Type 3 Tag Platform, the Listener SHALL be ready to receive a SENSF_REQ Command with RC set to 01h or 02h.

Requirements 105: Handling of RC Value of RFU

Listen Mode	
8.6.1.9	When sending a SENSF_RES Response indicating support for the Type 3 Tag Platform, a received RC value of RFU (as defined in Table 43) SHALL be treated as RC=00h.

TSN

The Time Slot Number (TSN) is used for collision resolution and reducing the probability of collisions. The anticollision scheme is based on the definition of time slots in which Listeners are invited to respond with minimum identification data.

The Poller sends a SENSF_REQ Command with a TSN value indicating the number of available time slots. Each Listener presents within the range of the Operating Field, and then randomly selects a time slot in which it responds. The TSN byte set to 00h forces all Listeners to respond in the first time slot, and, therefore, this TSN value is used if collision resolution is not employed.

Figure 24 shows an example of the operation of the collision resolution mechanism. The time slots are numbered sequentially, starting with 0. The Poller (Device 0) receives answers from four Listeners. Device 0 can distinguish Device 4 and Device 2 because they responded in different time slots. The Responses from Device 1 and Device 3 cannot be distinguished because a collision occurred in time slot 1.

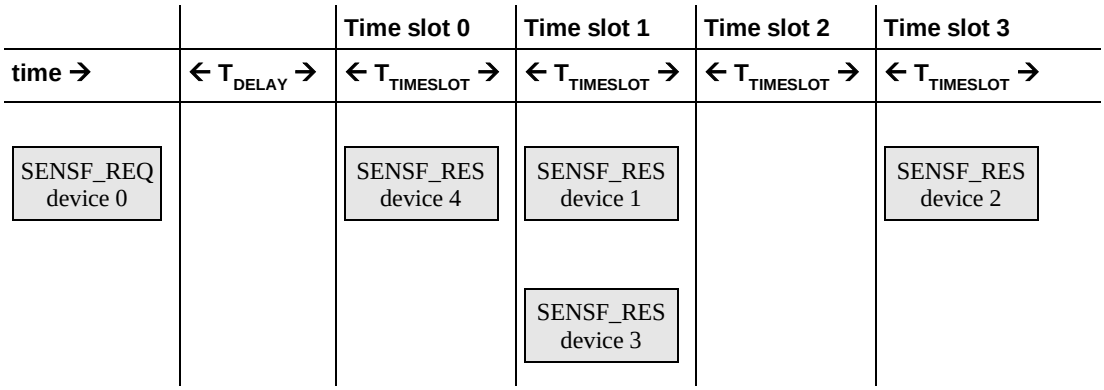


Figure 24: Collision Resolution (Example)

Table 44 specifies the coding of the TSN byte.

Table 44: TSN Coding

TSN	00h	01h	03h	07h	0Fh	All other values
Number of Time Slots	1	2	4	8	16	RFU
Time Slots allowed for SENSF_RES Response	Time slot 0	Time slot 0 Time slot 1	Time slot 0 ... Time slot 3	Time slot 0 ... Time slot 7	Time slot 0 ... Time slot 15	

Requirements 106: TSN

Poll Mode		Listen Mode	
8.6.1.10	The Poller SHALL set the TSN byte as specified in Table 44.	8.6.1.11	The Listener SHALL be ready to receive a SENSF_REQ Command with a TSN byte set to a value specified in Table 44.
8.6.1.12	The Poller SHALL be ready to receive SENSF_RES Responses in each allowed time slot, according to Table 44.	8.6.1.13	The Listener SHALL randomly select a time slot allowed according to Table 44, and send its SENSF_RES Response in the selected time slot.

Requirements 107: Handling of TSN Value of RFU

Listen Mode	
8.6.1.14	The Listener MAY treat a received TSN value of RFU (as defined in Table 44) in an implementation specific manner; e.g. it might mask the received value with 0Fh and select a timeslot between 0 and the resulting value.

8.6.2 SENSF_RES Response

Table 45 defines the SENSF_RES format.

Table 45: SENSF_RES Format

Byte 1	Byte 2-9	Byte 10-11	Byte 12-14	Byte 15	Byte 16	Byte 17	[Byte 18-19]
01h	NFCID2	PAD0	Undefined Values (PAD1)	**MRTI**$_{CHECK}$	**MRTI**$_{UPDATE}$	Undefined Values (PAD2)	[RD]

Requirements 108: SENSF_RES Response

Poll Mode		Listen Mode	
8.6.2.1	The Poller SHALL be ready to receive a SENSF_RES Response that has the format specified in Table 45.	8.6.2.2	The Listener SHALL send the SENSF_RES Response formatted as specified in Table 45.

NFCID2

NFCID2 is the NFC Forum Device identifier. Table 46 shows the NFCID2 format.

Table 46: NFCID2 Format

Byte 1	Byte 2	Byte 3 – Byte 8	Description
01h	FEh	Dynamically Generated	NFC-DEP Protocol supported
02h	FEh	Device Identification	Type 3 Tag Platform supported
All other values		Proprietary Identification	Type 3 Tag Platform supported

Requirements 109: NFCID2 in SENSF_RES

Poll Mode		Listen Mode	
8.6.2.3	The Poller SHALL accept any NFCID2.	8.6.2.4	The NFCID2 SHALL be coded by the Listener as specified in Table 46.
		8.6.2.5	When Byte 1 and Byte 2 of the NFCID2 are equal to 01FEh, the dynamically generated Byte 3 through Byte 8 of the NFCID2 SHALL be generated whenever the **IDLE** State of the Listen Mode state machine in [ACTIVITY] is entered from any of the following states: **NO_REMOTE_FIELD**, **ATR_READY_A**, **ATR_READY_F**, **TARGET_A**, **TARGET_F**.
		8.6.2.6	When Byte 1 and Byte 2 of the NFCID2 are equal to 02FEh, the Listener MAY set the Device Identification Byte 3 through Byte 8 of the NFCID2 to any value.
		8.6.2.7	An NFCID2 indicating support for NFC-DEP Protocol SHALL only be used when responding to a SENSF_REQ with SC set to FFFFh and RC set to 00h.

NOTE The Proprietary Identification, Byte 3 through Byte 8 of the NFCID2, is not defined by this version of the specification.

PAD0

NFC Forum Devices do not use PAD0 for information exchange.

Requirements 110: PAD0

Poll Mode		Listen Mode	
8.6.2.8	The Poller SHALL be ready to receive a PAD0 field coded as any value.	8.6.2.9	If configured for NFC-DEP Protocol, the Listener SHALL set PAD0 to FFh FFh. Otherwise the Listener MAY set PAD0 to any value.

$MRTI_{CHECK}$

The **$MRTI_{CHECK}$** coding depends on the NFC-F Technology Subset for which the Listener is configured. NFC Forum Devices configured for the NFC-DEP Protocol do not use **$MRTI_{CHECK}$**.

Requirements 111: $MRTI_{CHECK}$

Poll Mode		Listen Mode	
8.6.2.10	If configured to poll for Listeners that are configured for the NFC-DEP Protocol, the Poller SHALL disregard any value returned in the **$MRTI_{CHECK}$** field. If configured to poll for Listeners that are configured for the Type 3 Tag Platform, the Poller SHALL be ready to receive an **$MRTI_{CHECK}$** value, as specified in [T3T].	8.6.2.11	If configured for Type 3 Tag Platform, the Listener SHALL set **$MRTI_{CHECK}$** as specified in[T3T]. Otherwise the Listener MAY set **$MRTI_{CHECK}$** to any value.

$MRTI_{UPDATE}$

The **$MRTI_{UPDATE}$** format depends on the NFC-F Technology Subset for which the Listener is configured. NFC Forum Devices configured for the NFC-DEP Protocol do not use **$MRTI_{UPDATE}$**.

Requirements 112: $MRTI_{UPDATE}$

Poll Mode		Listen Mode	
8.6.2.12	If configured to poll for the NFC-DEP Protocol, the Poller SHALL disregard any value returned in the **$MRTI_{UPDATE}$** field. If configured to poll for the Type 3 Tag Platform, the Poller SHALL be ready to receive an **$MRTI_{UPDATE}$** value, as specified in[T3T].	8.6.2.13	If configured for the Type 3 Tag Platform, the Listener SHALL set **$MRTI_{UPDATE}$** as specified in[T3T]. Otherwise the Listener MAY set **$MRTI_{UPDATE}$** to any value.

RD

Request Data (RD) is included in the SENSF_RES Response if requested in the RC field of the SENSF_REQ Command. The Request Data (RD) format depends on the NFC-F Technology Subset for which the Listener is configured.

Requirements 113: Request Data (RD)

Poll Mode		Listen Mode	
8.6.2.14	If the preceding SENSF_REQ Command contained an RC byte set to 00h, the Poller SHALL treat the presence of the RD field in the SENSF_RES Response as Syntax Error.	8.6.2.15	If the preceding SENSF_REQ Command contained an RC byte set to 00h, the Listener SHALL NOT send RD bytes within the SENSF_RES Response.
8.6.2.16	If configured to poll for Type 3 Tag Platform and if the preceding SENSF_REQ Command contained an RC byte set to 01h, the Poller SHALL be ready to receive a 2 byte RD in the SENSF_RES Response. It also SHALL accept a SENSF_RES without RD bytes as a Valid Response.	8.6.2.17	If configured for the Type 3 Tag Platform and if the preceding SENSF_REQ Command contained an RC byte set to 01h, the Listener SHALL: • Set the 2 RD bytes equal to the System Code. • Send out the RD bytes as part of the SENSF_RES Response MSB first.
8.6.2.18	If the preceding SENSF_REQ Command contained an RC byte set to 02h, the Poller SHALL be ready to receive a 2 byte RD in the SENSF_RES Response. It also SHALL accept a SENSF_RES without RD bytes as a Valid Response.	8.6.2.19	If the preceding SENSF_REQ Command contained an RC byte set to 02h, the Listener MAY send RD bytes (as defined in Table 47 and Table 48) within the SENSF_RES Response.

Table 47: RD Format Advanced Protocol Features (Byte 18)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				0				RFU
					0			RFU
						0		RFU
							0	RFU

Table 48: RD Format Advanced Protocol Features (Byte 19)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x								Automatically selectable bit rates supported, if bit is set to 1b.
	0							RFU
		0						RFU
			0					RFU
				x				$\mathbf{D}_{POLL \rightarrow LISTEN}$ =16 and $\mathbf{D}_{LISTEN \rightarrow POLL}$ = 16 supported, if bit is set to 1b.
					x			$\mathbf{D}_{POLL \rightarrow LISTEN}$ =8 and $\mathbf{D}_{LISTEN \rightarrow POLL}$ = 8 supported, if bit is set to 1b.
						x		$\mathbf{D}_{POLL \rightarrow LISTEN}$ =4 and $\mathbf{D}_{LISTEN \rightarrow POLL}$ = 4 supported, if bit is set to 1b.
							x	$\mathbf{D}_{POLL \rightarrow LISTEN}$ =2 and $\mathbf{D}_{LISTEN \rightarrow POLL}$ = 2 supported, if bit is set to 1b.

NOTE Bit rates above 424 kbits/s for NFC-F are not defined by this version of the specification.

Automatically selectable bit rate support (b8) is announced if a Listener is able to automatically identify the bit rate of received NFC-F Commands in the following States of the Listen Mode state machine defined in [ACTIVITY]: **`IDLE`**, **`READY_F`**, **`SLEEP_AF`** and **`CARD_EMULATOR_3`**.

Requirements 114: RD Format Advance Protocol Features (Byte 19)

Listen Mode	
8.6.2.20	If the Listener wishes to indicate support for automatically selectable bit rates, and if two or more bits out of b4 through b1 are set to 1b, the Listener SHALL set b8 of Byte 19 to 1b.
8.6.2.21	The Listener that is configured for the Type 3 Tag Platform and that supports automatically selectable bit rates SHALL accept a Command sent in any of the bit rates it supports.

8.7 Timing Requirements

This section specifies the requirements for the guard time and Frame Delay Times for NFC-F Technology.

NOTE Definitions in Section 8.7 deviate from definitions in [ISO/IEC_18092].

8.7.1 Frame Delay Time Poll→Listen

The Frame Delay Time Poll→Listen ($\mathbf{FDT}_{F,LISTEN}$) is the time between the end of the last bit of a Poll Frame and the start of the SoF of a Listen Frame, as shown in Figure 20. The time between the minimum value $\mathbf{FDT}_{F,LISTEN,MIN}$ and the maximum value $\mathbf{FDT}_{F,LISTEN,MAX}$ defines the time interval during which a Listen Frame is allowed to be sent in response to a Poll Frame.

NOTE The term "Frame Response Time" used in [ISO/IEC_18092] is referred to as "Frame Delay Time" in this document.

$\mathbf{FDT}_{F,LISTEN}$ (except SENSF_REQ)

$\mathbf{FDT}_{F,LISTEN,MIN}$ is the minimum waiting time for a Listener before it sends the Listen Frame after the end of a Poll Frame.

Except for the SENSF_REQ Command, $\mathbf{FDT}_{F,LISTEN,MIN}$ is defined:

$$\mathbf{FDT}_{F,LISTEN,MIN} = \mathbf{TR0}_{F,LISTEN,MIN} + \mathbf{TR1}_{F},$$

where $\mathbf{TR1}_{F}$ is the synchronization time (the bd of SoS), as illustrated in Figure 20. Appendix B.4 lists the values of $\mathbf{TR0}_{F,LISTEN,MIN}$ and $\mathbf{TR1}_{F}$.

Requirements 115: $\mathbf{FDT}_{F,LISTEN,MIN}$ (except SENSF_REQ)

Poll Mode		Listen Mode	
8.7.1.1	Except for the SENSF_REQ Command, the Poller SHALL be ready to receive the SoF of a new Listen Frame no later than $\mathbf{FDT}_{F,LISTEN,MIN}$ after the end of a Poll Frame.	8.7.1.2	Except for the SENSF_REQ Command following the end of a Poll Frame, the Listener SHALL wait at least $\mathbf{FDT}_{F,LISTEN,MIN}$ before it sends the SoF of its Listen Frame.

FDT$_{F,LISTEN,SENSF\_REQ}$

For the SENSF_REQ Command, **FDT**$_{F,LISTEN,SENSF\_REQ}$ depends on the time slot the Listener is responding in, and is defined:

$$\mathbf{FDT}_{F,LISTEN,SENSF\_REQ}\mathbf{(n)} = \mathbf{TR0}_{F,LISTEN}\mathbf{(n)} + \mathbf{TR1}_{F},$$

where **TR1**$_F$ is the synchronization time (the bd of SoS), as illustrated in Figure 20, and *n* is the time slot number randomly selected by the Listener ($0 \leq n \leq$ TSN).

TR0$_{F,LISTEN}$ is given in the following formula:

$$\mathbf{TR0}_{F,LISTEN}\mathbf{(n)} = \mathbf{T}_{DELAY} + n \times \mathbf{T}_{TIMESLOT}$$

Appendix B.4 lists the values of **TR1**$_F$, **T**$_{DELAY}$ and **T**$_{TIMESLOT}$.

Requirements 116: $FDT_{F,LISTEN,SENSF\_REQ}$

Poll Mode		Listen Mode	
8.7.1.3	Following the end of a Poll Frame containing the SENSF_REQ Command, the Poller SHALL be ready to receive the first bit of the SoF of the Listen Frame that contains the SENSF_RES Response no later than the end of the time period $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(0)}$. Following the end of a Poll Frame that contains the SENSF_REQ Command, the Poller SHALL accept the EoD of the Listen Frame containing the SENSF_RES Response no later than the end of the time period $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(TSN+1)} - \mathbf{TR1_F}$. If the Poller does not receive the last bit of the EoD of the Listen Frame that contains the SENSF_RES Response before the end of the time period $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(TSN+1)} - \mathbf{TR1_F} + \Delta\mathbf{T_{F,POLL}}$, then the Poller SHALL treat the lack of response as a Timeout Error. If the last bit of the EoD of the SENSF_RES Response occurs between the end of the time period $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(TSN+1)} - \mathbf{TR1_F}$ and the end of the time period $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(TSN+1)} - \mathbf{TR1_F} + \Delta\mathbf{T_{F,POLL}}$, the Poller MAY accept a SENSF_RES Response or MAY generate a Timeout Error. Appendix B.4 lists the value of $\Delta\mathbf{T_{F,POLL}}$.	8.7.1.4	Following the end of a SENSF_REQ Command, the Listener SHALL wait at least $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(n)} - \mathbf{TR1_F}$ before it sends the SoS of its Listen Frame that contains the SENSF_RES Response.
		8.7.1.5	Following the end of a SENSF_REQ Command, the Listener SHALL finish sending the EoD of the Listen Frame that contains the SENSF_RES Response no later than $\mathbf{FDT_{F,LISTEN,SENSF\_REQ}(n+1)} - \mathbf{TR1_F}$.

8.7.2 Frame Delay Time Poll→Poll

The Frame Delay Time Poll→Poll ($\mathbf{FDT}_{F,PP}$) is the time between the end of the last time slot set as the TSN parameter in a SENSF_REQ Command and the start of the SoF of a next Poll Frame, if a SENSF_RES Response is not sent. The minimum value $\mathbf{FDT}_{F,PP,MIN}$ defines the waiting time for a Poller before it sends a new Poll Frame after the end of the last time slot specified in the previous SENSF_REQ Command. The maximum value $\mathbf{FDT}_{F,PP,MAX}$ is not defined.

$\mathbf{FDT}_{F,PP,MIN}$ is defined:

$$\mathbf{FDT}_{F,PP,MIN} = \mathbf{TR0}_{F,PP,MIN} + \mathbf{TR1}_{F} ,$$

where $\mathbf{TR1}_{F}$ is the synchronization time (the bd of SoS), as illustrated in Figure 20. Appendix B.4 lists the values of $\mathbf{TR0}_{F,PP,MIN}$ and $\mathbf{TR1}_{F}$.

Requirements 117: $\mathbf{FDT}_{F,PP,MIN}$

Poll Mode		Listen Mode	
8.7.2.1	Following the end of the last time slot specified in the SENSF_REQ Command, the Poller SHALL wait at least for a time $\mathbf{FDT}_{F,PP,MIN}$ before transmitting the SoF of a new Poll Frame.	8.7.2.2	The Listener SHALL be ready to receive the SoF of a new Poll Frame no later than $\mathbf{FDT}_{F,PP,MIN}$ after the end of the last time slot specified in the most recently received SENSF_REQ Command. If the SoF of a Poll Frame is received before $\mathbf{FDT}_{F,PP,MIN}$, then the Listener MAY treat this receipt as a Transmission Error.

8.7.3 Frame Delay Time Listen→Poll

The Frame Delay Time Listen→Poll ($\mathbf{FDT}_{F,POLL}$) is the time between the end of the last bit of a Listen Frame and the start of the SoF of a Poll Frame, as shown in Figure 20. If the Listen Frame is a SENSF_RES, the Frame Delay Time Listen→Poll ($\mathbf{FDT}_{F,POLL}$) is the time between the end of the time slot specified in the SENSF_REQ and the start of the SoF of a Poll Frame. The minimum value $\mathbf{FDT}_{F,POLL,MIN}$ defines the waiting time for a Poller before it sends a new Poll Frame after receipt of a Listen Frame. The maximum value $\mathbf{FDT}_{F,POLL,MAX}$ is not defined.

$\mathbf{FDT}_{F,POLL,MIN}$ is defined:

$$\mathbf{FDT}_{F,POLL,MIN} = \mathbf{TR0}_{F,POLL,MIN} + \mathbf{TR1}_{F} ,$$

where $\mathbf{TR1}_{F}$ is the synchronization time (the bd of SoS), as illustrated in Figure 20. Appendix B.4 lists the values of $\mathbf{TR0}_{F,POLL,MIN}$ and $\mathbf{TR1}_{F}$.

Requirements 118: $FDT_{F,POLL,MIN}$

Poll Mode		Listen Mode	
8.7.3.1	Except for the SENSF_RES following the end of a Listen Frame, the Poller SHALL wait at least **$FDT_{F,POLL,MIN}$** before transmitting the SoF of a new Poll Frame. For the SENSF_RES, following the end of the time slot specified in the SENSF_REQ, the Poller SHALL wait at least for a time **$FDT_{F,POLL,MIN}$** before transmitting the SoF of a new Poll Frame.	8.7.3.2	The Listener SHALL be ready to receive the SoF of a new Poll Frame no later than **$FDT_{F,POLL,MIN}$** after the Listen Frame. If the SoF of a Poll Frame is received before **$FDT_{F,POLL,MIN}$**, then the Listener MAY treat this receipt as a Transmission Error.

A specific value for **$FDT_{F,POLL,MIN}$** is defined for the re-activation of the Listener following a DSL_RES Response (see Section 17.9.2) or an RLS_RES Response (see Section 17.10.2).

Requirements 119: $FDT_{F,REACTIVATION}$

Poll Mode		Listen Mode	
8.7.3.3	Following the end of a DSL_RES Response or an RLS_RES Response, the Poller SHALL wait at least for **$FDT_{F,REACTIVATION}$** before transmitting the start of a new Poll Frame. Appendix B.4 lists the value of **$FDT_{F,REACTIVATION}$**.	8.7.3.4	The Listener SHALL be ready to receive the start of a new Poll Frame no later than **$FDT_{F,REACTIVATION}$** after the end of a DSL_RES Response or an RLS_RES Response. If the start of a new Poll Frame is received before **$FDT_{F,REACTIVATION}$**, then the Listener MAY treat this receipt as a Transmission Error.

NOTE Requirement 8.7.3.3 only applies for the Initiator and not for the Reader/Writer. When operating as Reader/Writer, reactivation of a Type 3 Tag Platform is done by means of a Reset, as defined in [ANALOG].

NOTE Requirement 8.7.3.4 applies for Target and not for Type 3 Tag (emulation).

8.7.4 Guard Time

This section specifies the guard time of an Unmodulated Carrier, after which the Listener is ready to receive the first Command.

Requirements 120: NFC-F Guard Time

Listen Mode	
8.7.4.1	When a Listener is exposed to an Unmodulated Carrier (see [ANALOG]), it SHALL be ready to receive the first Command. The first Command can be either a SENSF_REQ Command or a Type 3 Tag Platform Command, if it is configured for the Type 3 Tag Platform, after a guard time $\mathbf{GT}_F$. Appendix B.4 lists the value of $\mathbf{GT}_F$.

9 NFC-V Technology

This section specifies the NFC-V Technology-related features of the NFC Forum Device.

This section is derived from [ISO/IEC_15693].

9.1 Sequence Format

This section describes the sequence format for NFC-V Technology.

9.1.1 Poll→Listen Modulation

In Poll Mode the analog signal is modulated using pulse position coding with ASK 100% modulation.

The modulation principle of ASK 100% of the Operating Field is used to create “pause”. [ANALOG] defines “pause”.

For data coding, 1 out of 4 coding is used as illustrated in Figure 25, which simultaneously codes two data bits (named couple). This coding defines four particular patterns: A, B, C and D, which determine the couple value, two bits at a time. When grouped by four, they form a byte whose least significant bit is transmitted first.

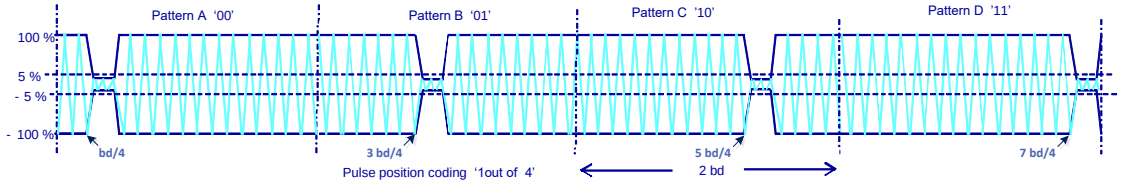


Figure 25: Pulse Position Coding Couple with ASK 100%

Requirements 121: NFC-V Data Coding Poll→Listen

Poll Mode		Listen Mode	
9.1.1.1	The Poller SHALL transmit patterns using the 1 out of 4 data coding modulation.	9.1.1.2	The Listener SHALL accept patterns using the 1 out of 4 data coding modulation.
9.1.1.3	The Poller SHALL transmit patterns using ASK 100% modulation.	9.1.1.4	The Listener SHALL accept patterns using ASK 100% modulation.

NOTE The requirements above limit the possible choices of Poll→Listen Modulation and data coding for the NFC Forum Device. Compare these to [ISO/IEC_15693].

Requirements 122: NFC-V Signal Patterns Poll→Listen

Poll Mode		Listen Mode	
9.1.1.5	The Poller SHALL build signal patterns A, B, C, and D with a pattern duration of 2 bd.	9.1.1.6	The Listener SHALL accept signal patterns A, B, C, and D with a pattern duration of 2 bd.
9.1.1.7	The Poller SHALL build signal patterns A, B, C, and D as follows: • Pattern A: at the beginning of the pattern duration, no modulation SHALL occur for a time of 1/4 bd, then a "pause" SHALL occur. For the rest of the pattern duration no modulation SHALL occur. • Pattern B: at the beginning of the pattern duration, no modulation SHALL occur for a time of 3/4 bd, then "pause" SHALL occur. For the rest of pattern duration no modulation SHALL occur. • Pattern C: at the beginning of the pattern duration, no modulation SHALL occur for a time of 5/4 bd, then "pause" SHALL occur. For the rest of pattern duration no modulation SHALL occur. • Pattern D: at the beginning of the pattern duration, no modulation SHALL occur for a time of 7/4 bd, then "pause" SHALL occur.	9.1.1.8	The Listener SHALL read signal patterns A, B, C, and D as follows: • If the Listener detects no modulation for a time 1/4 bd, then a "pause" and then no modulation for the rest of the pattern duration, it SHALL read this as pattern A. • If the Listener detects no modulation for a time of 3/4 bd, then "pause" and then no modulation for the rest of the pattern duration, it SHALL read this as pattern B. • If the Listener detects no modulation for a time of 5/4 bd, then "pause" and then no modulation for the rest of the pattern duration, it SHALL read this as pattern C. • If the Listener detects no modulation for a time of 7/4 bd, then "pause" and then no modulation for the rest of the pattern duration, it SHALL read this as pattern D.

NOTE The term "Pulse" as used in [ISO/IEC_15693] is referred to as "pause" in this document.

For the Start of Frame (SoF) a special Pulse Position coding pattern is used, as illustrated in Figure 26.

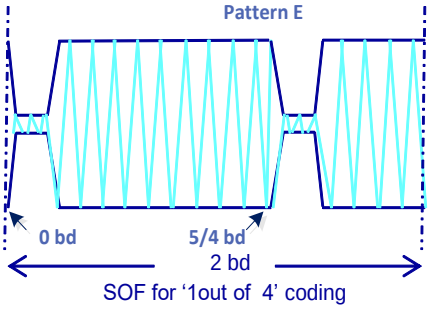


Figure 26: SoF using Pulse modulation coding

Requirements 123: NFC-V Start of Frame Poll→Listen

Poll Mode		Listen Mode	
9.1.1.9	The Poller SHALL build the SoF with a pattern duration of 2 bd as follows: "pause" SHALL occur at the beginning of the SoF, then no modulation SHALL occur during a time of 1 bd, then "pause" SHALL occur. No modulation SHALL occur for the rest of the SoF .	9.1.1.10	If the Listener detects a pattern of duration 2 bd starting with a "pause", then no modulation for a time of 1 bd, then a "pause" and then no modulation for rest of the pattern, it SHALL read this as an SoF.

For the End of Frame (EoF) a special Pulse Position coding pattern is used as respectively illustrated in Figure 27.

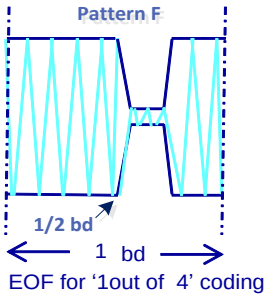


Figure 27: EoF using Pulse modulation coding

Requirements 124: NFC-V End of Frame Poll→Listen

Poll Mode		Listen Mode	
9.1.1.11	The Poller SHALL build the EoF with a pattern duration of 1 bd as follows: At the beginning of the EoF, no modulation SHALL occur for a period of 1/2 bd, then "pause" SHALL occur. No modulation SHALL occur for the rest of the pattern duration.	9.1.1.12	If the Listener detects a pattern of duration 1 bd starting with no modulation for a period of 1/2 bd, then a "pause" and then no modulation for the rest of the pattern, it SHALL read this as EoF.

9.1.2 Listen→Poll Modulation

The Poller receives data from a device that implements the Type 5 Tag Platform using the modulation defined in this section.

Requirements 125: NFC-V Modulation Listen→Poll

Poll Mode		Listen Mode	
9.1.2.1	The Poller SHALL request the load modulation with a Single Subcarrier (see Section 9.6.1 and [T5T]).	9.1.2.2	The Listener SHALL send Frames using load modulation with a Single Subcarrier (see Section 9.6.1 and [T5T]).
9.1.2.3	The Poller SHALL request the High Data Rate (see Section 9.6.1 and [T5T]).	9.1.2.4	The Listener SHALL send Frames using the High Data Rate (see Section 9.6.1 and [T5T]).

NOTE The requirements above limit the possible choices of Listen→Poll Modulation and data coding for the NFC Forum Device. Compare these to [ISO/IEC_15693].

For data coding the analog signal is modulated using Manchester coding with OOK subcarrier modulation (see Figure 28).

The frequency $\mathbf{f}_S$ of the subcarrier load modulation is $\mathbf{f}_C/32$. See [ANALOG].

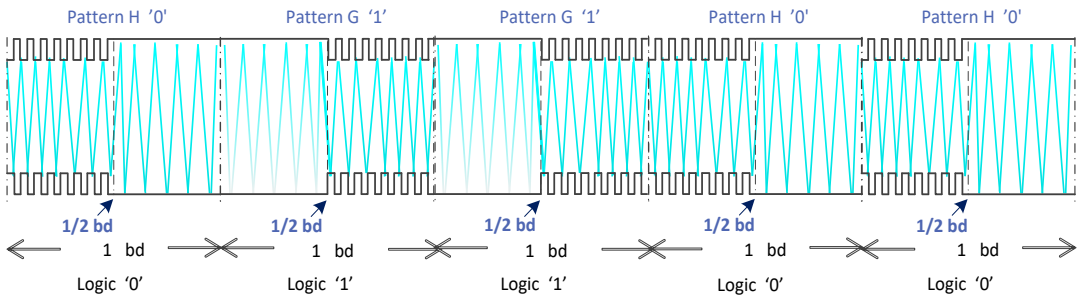


Figure 28: NFC-V Manchester coding with OOK

Manchester coding uses OOK subcarrier modulation to define two particular patterns: G representing a logical “1”, and H representing a logical “0”.

Requirements 126: NFC-V Signal Patterns Listen→Poll

Poll Mode			
9.1.2.5	The Poller SHALL read signal patterns G and H with a duration of 1 bd as follows: • If the carrier is not modulated with the subcarrier for the first 1/2 bd and is modulated for the second 1/2 bd, the Poller SHALL read that as pattern G representing a logical “1”. • If the carrier is modulated with the subcarrier for the first 1/2 bd and is not modulated for the second 1/2 bd, the Poller SHALL read that as pattern H representing a logical “0”.	9.1.2.6	The Listener SHALL encode a logical bit as follows: • Logical “1” is encoded by Pattern G, defined as no modulation for the first 1/2 bd, then subcarrier modulation for the second 1/2 bd. • Logical “0” is encoded by Pattern H, defined as subcarrier modulation for the first 1/2 bd then no modulation for the second 1/2 bd.

For the Start of Frame (SoF), a special load modulation pattern is used as illustrated in Figure 29.

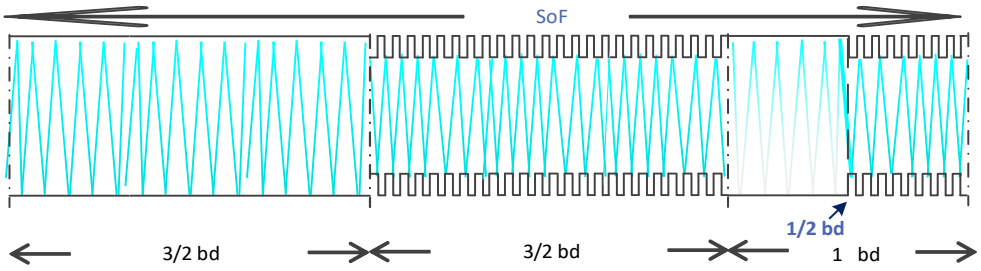


Figure 29: NFC-V SoF using Manchester coding with OOK

Requirements 127: NFC-V Start of Frame Listen→Poll

Poll Mode		Listen Mode	
9.1.2.7	If the Poller detects a pattern of duration 4 bd starting with no modulation for 3/2 bd, then followed by a subcarrier modulation for another 3/2 bd and followed by pattern G, it SHALL read that as an SoF.	9.1.2.8	The Listener SHALL build the SoF as follows: • No modulation for 3/2 bd, then subcarrier modulation for 3/2 bd, followed by pattern G.

For the End of Frame (EoF), a special load modulation pattern is used, as illustrated in Figure 30.

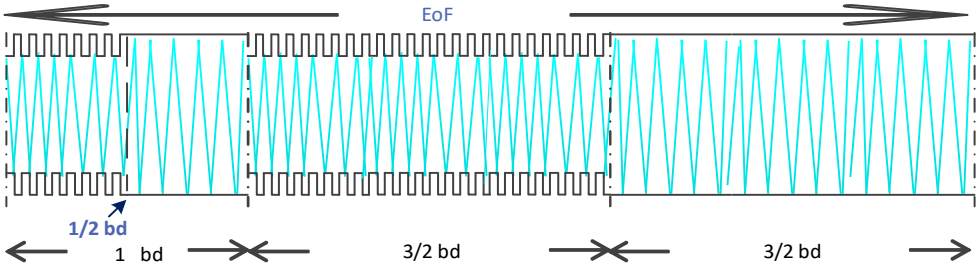


Figure 30: NFC-V EoF using Manchester coding with OOK

Requirements 128: NFC-V End of Frame Listen→Poll

Poll Mode		Listen Mode	
9.1.2.9	If the Poller detects a pattern of duration 4 bd starting with pattern H, followed by a subcarrier modulation for 3/2 bd, followed by no modulation for another 3/2 bd, it SHALL read that as an EoF.	9.1.2.10	The Listener SHALL build the SoF as follows: • Pattern H, followed by a subcarrier modulation for 3/2 bd, then followed by no modulation for another 3/2 bd.

9.1.3 Synchronization

NFC-V Technology does not require signal synchronization, and therefore SoS is not present.

9.1.4 De-synchronization

NFC-V Technology does not require signal de-synchronization, and therefore EoS is not present.

9.2 Bit Level Coding

9.2.1 Poll→Listen Coding Scheme

NFC-V Technology determines two bits at a time. Four successive pairs of bits determine a byte.

The patterns A, B, C and D are used to code the Logic 00b, 01b, 10b and 11b. The SoF and EoF (defined in Section 9.1.1) are used as delimiters.

Requirements 129: NFC-V Bit Level Coding Poll→Listen

Poll Mode		Listen Mode	
9.2.1.1	The Poller SHALL code Logic 00b, 01b, 10b, and 11b as follows: • Logic 00b: pattern A • Logic 01b: pattern B • Logic 10b: pattern C • Logic 11b: pattern D	9.2.1.2	The Listener SHALL decode Pattern A, B, C, and D as follows: • Pattern A SHALL be decoded as Logic 00b. • Pattern B SHALL be decoded as Logic 01b. • Pattern C SHALL be decoded as Logic 10b. • Pattern D SHALL be decoded as Logic 11b.

9.2.2 Listen→Poll Coding Scheme

The patterns G and H are used to code the digital Logic 0b and Logic 1b. Logic 0b and Logic 1b, referred to as data bits in the following, are the components of frames.

The SoF and EoF (defined in Section 9.1.2) are used as delimiters.

Requirements 130: NFC-V Bit Level Coding Listen→Poll

Poll Mode		Listen Mode	
9.2.2.1	The Poller SHALL decode pattern G and pattern H as follows: • Pattern G SHALL be decoded as Logic 1b. • Pattern H SHALL be decoded as Logic 0b.	9.2.2.2	The Listener SHALL code Logic 0b and 1b as follows: • Logic 1b: pattern G • Logic 0b: pattern H

9.3 Frame Format

This section defines the frames of the NFC Forum Devices that are configured for NFC-V Technology. NFC-V Technology uses three types of frames: Standard Frame, Isolated EoF Frame, and Special Frame.

Data bits, when transmitted between NFC Forum Devices, are grouped per byte within frames. NFC-V Technology groups the data bits per byte together in a frame by adding an SoF and an EoF.

9.3.1 Standard Frame

Standard Frames are used for data exchange and consist of the following (see Figure 31):

- SoF
- n * (8 data bits), with n ≥ 1
- EoF

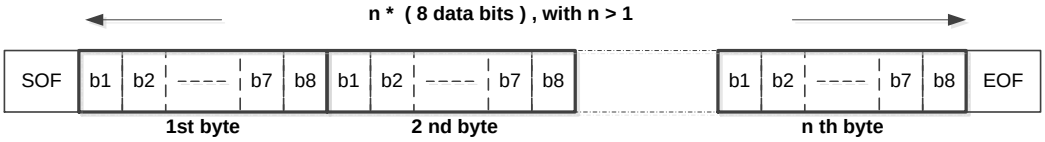


Figure 31: NFC-V Standard Frame Format

A frame contains fields.

A field can be flags, command code, parameters, data or CRC.

Any byte is transmitted least significant bit first.

A multiple byte field is transmitted LSB first.

Requirements 131: NFC-V Standard Frame Format for Transmission

Poll Mode	
9.3.1.1	A Standard Frame SHALL start with SoF.
9.3.1.2	A Standard Frame SHALL end with EoF.
9.3.1.3	A Standard Frame SHALL contain integer number of bytes.
9.3.1.4	For each byte in a Standard Frame the least significant bit SHALL be transmitted first.

Requirements 132: NFC-V Standard Frame Format for Reception

Poll Mode	
9.3.1.5	The Poller SHALL interpret a frame with a number of bits no multiple of eight as a Transmission error.
9.3.1.6	The Poller SHALL interpret the first received bit of a byte as the least significant bit of this byte.

9.3.2 Isolated EoF Frame

A frame consisting only of an EoF is used for INVENTORY_REQ to start a time slot and in NFC-V Special Frame to start the response window.

9.3.3 Special Frame

A frame consisting of a Standard Frame plus one Isolated EoF Frame transmitted after $\mathbf{FDT}_{V,EOF}$ (see Figure 32).

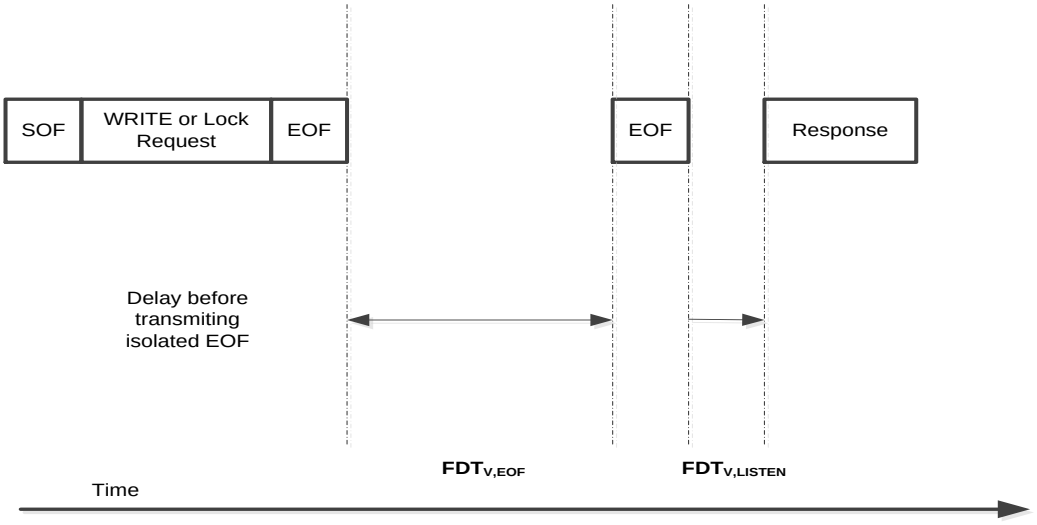


Figure 32: NFC-V Special Frame

Requirements 133: NFC-V Special Frame Format

Poll and Listen Mode	
9.3.3.1	A Special Frame SHALL start with a Standard Frame.
9.3.3.2	A Special Frame SHALL end with an Isolated EoF Frame transmitted or received after a time $\mathbf{FDT}_{V,EOF}$.

9.4 Data and Payload Format

Data transmitted in an NFC-V Technology Standard Frame (i.e., the bytes between SoF and EoF), have the following substructure. They consist of the payload and an EoD (SoD is not used).

The payload contains either a Command or a Response.

The EoD contains a two-byte checksum referred to as CRC_V. The CRC_V is defined as a function of payload data bits.

Figure 33 illustrates the NFC-V data and payload format included in a frame.

Data					
Payload (Command or Response)				**EoD**	
Byte 1	Byte 2	…	Byte n	CRC_V1	CRC_V2

Figure 33: NFC-V Standard Frame Data and Payload Format

Requirements 134: NFC-V Data and Payload Format

Poll and Listen Mode	
9.4.1.1	Data SHALL be transmitted in frames, as defined in Section 9.3.
9.4.1.2	The payload SHALL be followed by an EoD at the position indicated in Figure 33.
9.4.1.3	The CRC_V SHALL be calculated over all payload bytes (as defined in [ISO/IEC_13239]), and the initial register content SHALL be FFFFh. The input for the CRC_V calculation SHALL be the payload.
9.4.1.4	EoD SHALL be transmitted least significant byte first and least significant bit first. CRC_V1 is the LSB and CRC_V2 is the MSB.
9.4.1.5	The NFC Forum Device SHALL compare the received CRC_V with the CRC_V calculated according to 9.4.1.4. If different, then the NFC Forum Device SHALL treat the received data as a Transmission Error.

9.5 Command Set

9.5.1 General

The payload exchanged between NFC Forum Devices consists of Commands and Responses. Table 49 lists the Commands that are available to the NFC Forum Device configured for NFC-V Technology. For each Command the corresponding Response is indicated. Furthermore, Table 49 shows the associated flag types and frame types.

Table 49: NFC-V Command Set

Command	Response	Flag Type	Frame Type
INVENTORY_REQ		INV_FLAG	Standard Frame
	INVENTORY__RES	RES_FLAG	Standard Frame

Requirements 135: NFC-V Command Set

Poll and Listen Mode	
9.5.1.1	Commands and Responses SHALL be transmitted as the payload of the data and payload Format, defined in Section 9.4. Commands SHALL be transmitted using a bd of 512 / $\mathbf{f}_C$. Responses SHALL be transmitted using a bd of 512 / $\mathbf{f}_C$.

9.5.2 Unique Identifier

During communication a Type 5 Tag Platform is identified by the 8-byte UID formatted as defined in Figure 34.

LSB		UID		MSB
Byte 1	–	Byte 6	Byte 7	Byte 8
IC serial number			IC Manufacturer	Technology 0xE0

Figure 34: UID Format

The most significant byte (Byte 8) contains the value E0h to indicate an UID used for Type 5 Tag Platform.

The IC manufacturer byte (Byte 7) is coded according to [ISO/IEC_7816-6].

The IC serial number bytes (Byte 1 through Byte 6) are assigned by the manufacturer of the device that implements the Type 5 Tag Platform.

9.6 INVENTORY_REQ

The INVENTORY_REQ Command is sent by the Poller to probe the Operating Field for devices that implement the Type 5 Tag Platform.

The INVENTORY_REQ Command starts the communication.

Two operation modes of INVENTORY_REQ are defined as follows:

- “1 Slot Mode” provides 1 time slot for the Type 5 Tag to respond.
- “16 Slot Mode” provides up to 16 time slots for the Type 5 Tag to respond.

Time slots are started by the Poller by transmitting successive pattern F (Isolated EoF Frame). Timing reference is the rising edge of the pulse in the EoF.

The Poller can interrupt the collision resolution (as defined by [ACTIVITY]) by sending any Command.

9.6.1 INVENTORY_REQ Command

Table 50 defines the format of the INVENTORY_REQ Command.

Table 50: INVENTORY_REQ Command Format

Byte 1	Byte 2	Byte 3	Byte 4 – Byte (n+3)
INV_FLAG	01h	MASK_LEN	Mask value

Requirements 136: INVENTORY_REQ Command

Poll Mode		Listen Mode	
9.6.1.1	The Poller SHALL send the INVENTORY_REQ Command formatted as specified in Table 50.	9.6.1.2	The Listener SHALL be ready to receive an INVENTORY_REQ Command that has the format specified in Table 50.

INV_FLAG

Table 51 specifies the format of INV_FLAG.

Table 51: INV_FLAG Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0		0	0				RFU
		x						Number of Slots (NbS) 0b: 16 Slot Mode 1b: 1 Slot Mode
					1			1b: INVENTORY_REQ Command
						1		1b: High Data Rate
							0	0b: Single Subcarrier for Command Response

NOTE Table 51 limits the possible options for the INVENTORY_REQ Command compared to the options defined in [ISO/IEC_15693].

The Number of Slots (NbS) is used by the anticollision scheme defined in [ISO/IEC_15693]. The anticollision scheme is based on the definition of time slots in which the Type 5 Tag Platform is invited to respond. If b6 = 0b, NbS takes the value 16, corresponding to “16 Slot Mode”. If b6 = 1b, NbS takes the value 1, corresponding to “1 Slot Mode”.

MASK_LEN and Mask value

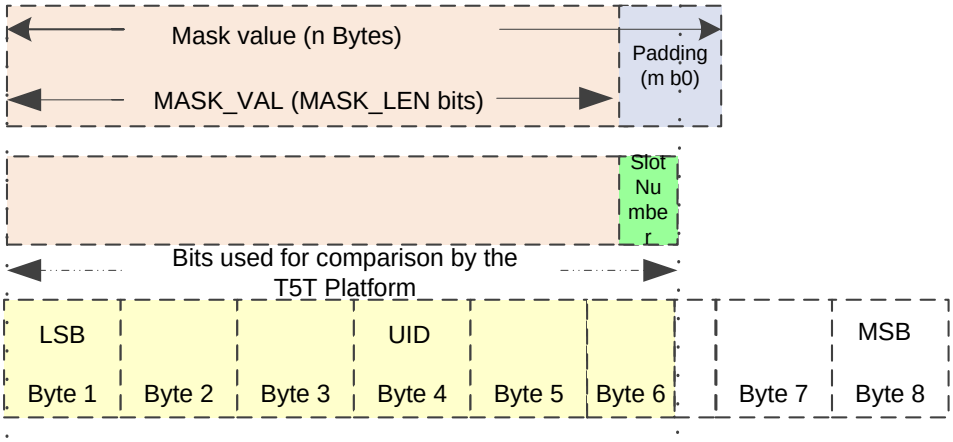


Figure 35: Example of comparison field in INVENTORY_REQ Command

Requirements 137: INVENTORY_REQ Command

Poll Mode		Listen Mode	
9.6.1.3	The Poller SHALL set the INV_FLAG according to Table 51.	9.6.1.4	The Listener SHALL be ready to receive the INV_FLAG according to Table 51.
9.6.1.5	The Poller SHALL set the NbS bit to 0b to operate in “16 Slot Mode”. The Poller SHALL set the NbS bit to 1b to operate in “1 Slot Mode”.		
9.6.1.6	The Poller SHALL set the MASK_LEN byte to a value in the range 0 to 60 for “16 Slot Mode”. The Poller SHALL set the MASK_LEN byte to a value in the range 0 to 64 for “1 Slot Mode”.	9.6.1.7	The Listener SHALL use the MASK_LEN value to define the partial UID that will be used for comparison.
9.6.1.8	The Poller SHALL compute the size in bytes of the Mask value field according to the following formula: n = (MASK_LEN +7) Div 8.	9.6.1.9	The Listener SHALL compare the partial UID defined by the MASK_LEN with the MASK_VAL. • If there is no comparison match, the Listener SHALL NOT respond. • If there is a comparison match and NbS is 1 Slot Mode, the Listener SHALL respond immediately. • If there is a comparison match and NbS is 16 Slot Mode, the Listener SHALL respond in the slot defined by the 4 bits of its UID following the partial UID.
9.6.1.10	The Poller SHALL fill the Mask value field as follows: • MASK_VAL which is composed of the MASK_LEN least significant bits used for partial UID comparison • Pad with 0b msb to obtain an integer number of bytes		

NOTE When it receives an INVENTORY_REQ Command in 1 Slot Mode, a device that is in Listen Mode and implements the Type 5 Tag Platform compares the MASK_VAL to the corresponding least significant bits of its UID. The device will respond if these bits match; otherwise it will not respond.

NOTE When it receives an INVENTORY_REQ Command in 16 Slot Mode, a device that is in Listen Mode and implements the Type 5 Tag Platform compares the (MASK_LEN + 4) bits formed by concatenation of the 4 bit slot number followed by MASK_VAL to the corresponding least significant bits of its UID. The device will respond if these bits match; otherwise it will not respond.

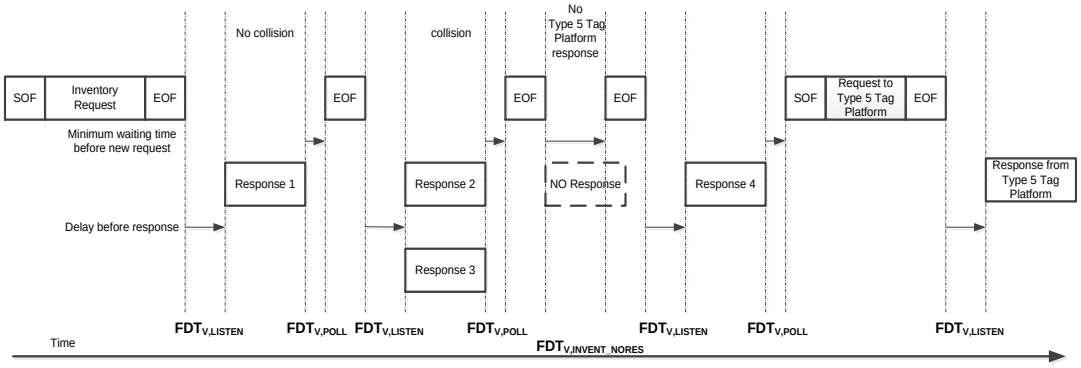


Figure 36: Example of INVENTORY_REQ Command in "16 Slot Mode"

9.6.2 INVENTORY_RES Response

Table 52 defines the format of the INVENTORY_RES Response.

Table 52: INVENTORY_RES Response Format

Byte 1	Byte 2	Byte 3 – 10
RES_FLAG	DSFID	UID

Requirements 138: INVENTORY_RES Response

Poll Mode		Listen Mode	
9.6.2.1	The Poller SHALL be ready to receive an INVENTORY_RES Response with a payload of 10 bytes, as defined in Table 52.	9.6.2.2	The Listener SHALL build the INVENTORY_RES Response to a valid INVENTORY_REQ Command according to Table 52.
9.6.2.3	The Poller SHALL treat the received frame as a Syntax Error if the RES_FLAG is not equal to 00h.	9.6.2.4	The Listener SHALL set the RES_FLAG to 00h.
9.6.2.5	The Poller MAY treat as a Syntax Error a received UID that is not formatted according to Figure 34.	9.6.2.6	The Listener SHALL set the UID according to Figure 34.
9.6.2.7	The Poller SHALL be ready to receive an INVENTORY_RES Response frame starting after a time $\mathbf{FTD}_{V,LISTEN}$. $\mathbf{FTD}_{V,LISTEN}$ starts after the rising edge of EoF defining a time slot. See Section 9.7.1 for $\mathbf{FTD}_{V,LISTEN}$.	9.6.2.8	The Listener SHALL respond to an INVENTORY_REQ after a time $\mathbf{FTD}_{V,LISTEN}$. $\mathbf{FTD}_{V,LISTEN}$ starts after the rising edge of EoF defining the time slot.
9.6.2.9	The Poller SHALL accept any value of DSFID.		

NOTE No DSFID value is reserved for NDEF format. It is recommended for the Type 5 Tag Platform to use a DSFID equal to 00h.

9.7 Timing Requirements

This section specifies the requirements for the guard time and Frame Delay Times for NFC-V Technology.

9.7.1 Frame Delay Time Poll→Listen

The Frame Delay Time Poll→Listen ($\mathbf{FDT}_{V,LISTEN}$) is the time between a Poll Frame and a Listen Frame.

Requirements 139: $FDT_{V,LISTEN}$

Poll Mode		Listen Mode	
9.7.1.1	If the Poller used a Standard Frame for the Command, then it SHALL be ready to receive a Response starting after a time $\mathbf{FDT}_{V,LISTEN,MIN}$ following the end of the Standard Frame. $\mathbf{FDT}_{V,LISTEN,MIN}$ starts from the rising edge of the Command EoF. Appendix B.5 lists the value of $\mathbf{FDT}_{V,LISTEN,MIN}$.		
9.7.1.2	If the Poller used a Special Frame for the Command, then it SHALL be ready to receive a Response starting after a time $\mathbf{FDT}_{V,LISTEN,MIN}$ following the end of the Isolated EoF Frame. $\mathbf{FDT}_{V,LISTEN,MIN}$ starts from the rising edge of the Isolated EoF Frame.		
9.7.1.3	For the Read-Alike Commands the Poller SHALL treat as a Timeout Error the receipt of a Response at a time after $\mathbf{FDT}_{V,LISTEN,MAX1} + \Delta\mathbf{T}_{V,POLL}$. Between $\mathbf{FDT}_{V,LISTEN,MAX1}$ and $\mathbf{FDT}_{V,LSITEN,MAX1} + \Delta\mathbf{T}_{V,POLL}$, the Poller MAY accept a Response or MAY generate a Timeout Error. Appendix B.5 lists the values of $\mathbf{FDT}_{V,LISTEN,MAX1}$ and $\Delta\mathbf{T}_{V,POLL}$. [T5T] defines the Read-Alike Commands.	9.7.1.4	The Listener SHALL respond to Read-Alike Commands received in a Standard Frame after a time $\mathbf{FDT}_{V,LISTEN,MIN}$ and before a time $\mathbf{FDT}_{V,LISTEN,MAX1}$.

Poll Mode		Listen Mode	
9.7.1.5	For the Write-Alike Commands the Poller SHALL treat as a Timeout Error the receipt of a Response at a time after $\mathbf{FDT}_{V,LISTEN,MAX2}$ + $\Delta\mathbf{T}_{V,POLL}$. Between $\mathbf{FDT}_{V,LISTEN,MAX2}$ and $\mathbf{FDT}_{V,LISTEN,MAX2}$ + $\Delta\mathbf{T}_{V,POLL}$, the Poller MAY accept a Response or MAY generate a Timeout Error. Appendix B.5 lists the values of $\mathbf{FDT}_{V,LISTEN,MAX2}$ and $\Delta\mathbf{T}_{V,POLL}$. [T5T] defines the Write-Alike Commands.	9.7.1.6	The Listener SHALL respond on the Write-Alike grid to Write-Alike Commands received in a Standard Frame after the time $\mathbf{FDT}_{V,LISTEN,MIN}$ and before the time $\mathbf{FDT}_{V,LISTEN,MAX2}$. Section 9.7.6 defines the Write-Alike grid.
		9.7.1.7	The Listener SHALL respond to Write-Alike Commands received in a Special Frame after the time $\mathbf{FDT}_{V,LISTEN,MIN}$ and before the time $\mathbf{FDT}_{V,LISTEN,MAX2}$.

9.7.2 Frame Delay Time Poll→Poll

The Frame Delay Time Poll→Poll ($\mathbf{FDT}_{V,PP}$) is the time between two consecutive Poll Frames during which no Listen Frame is received. If a Listen Frame is received in response to a Poll Frame, the requirements in Section 9.7.3 supersede the requirements for $\mathbf{FDT}_{V,PP}$. If no Listen Frame is received within the expected time window, the value $\mathbf{FDT}_{V,PP}$ defines the time a Poller waits before it sends a new Poll Frame after the end of a previous Poll Frame.

If the previous Poll Frame contains an INVENTORY_REQ Command or the isolated EOF part of anticollision, then $\mathbf{FDT}_{V,PP}$ is equal to $\mathbf{FDT}_{V,INVENT\_NORES}$. Otherwise, $\mathbf{FDT}_{V,PP}$ is equal to $\mathbf{FDT}_{V,POLL}$.

Appendix B.5 lists the values of $\mathbf{FDT}_{V,POLL}$ and $\mathbf{FDT}_{V,INVENT\_NORES}$.

Requirements 140: $FDT_{V,PP}$

Poll Mode		Listen Mode	
9.7.2.1	Following the end of an NFC-V Poll Frame, unless an NFC-V Listen Frame is received, the Poller SHALL wait at least **$FDT_{V,PP}$** before transmitting the SoF of a new Frame.	9.7.2.2	The Listener SHALL be ready to receive the SoF of a new Poll Frame no later than **$FDT_{V,PP}$** after the end of a Poll Frame, if it does not send a Listen Frame in response. If the Listener has not sent a Listen Frame and the SoF of a new Poll Frame is received before **$FDT_{V,PP}$**, then the Listener MAY treat this receipt as a Transmission Error.

9.7.3 Frame Delay Time Listen→Poll

The Frame Delay Time Listen→Poll (**$FDT_{V,POLL}$**) is the time between a Listen Frame and a new Poll Frame.

Requirements 141: $FDT_{V,POLL}$

Poll Mode		Listen Mode	
9.7.3.1	Following a Listen Frame of one or several remote targets, the Poller SHALL wait for at least **$FDT_{V,POLL}$** before transmitting a new Poll Frame. **$FDT_{V,POLL}$** starts from the rising edge of EoF of the Listen Frame. Appendix B.5 lists the value of **$FDT_{V,POLL}$**.	9.7.3.2	After transmitting any Listen Frame, the Listener SHALL be ready to receive the start of a new Poll Frame no later than **$FDT_{V,POLL}$**. After transmitting any Listen Frame, if the start of a new Poll Frame is received before **$FDT_{V,POLL}$**, then the Listener MAY treat this receipt as a Transmission Error.

9.7.4 Special Frame EoF Delay Time

In Poll Mode, during transmission of a Special Frame, the EoF Delay Time ($\mathbf{FDT}_{V,EOF}$) is the time between the EoF of a Standard Frame and the start of the following Isolated EoF Frame.

Requirements 142: $\mathbf{FDT}_{V,EOF}$

Poll Mode		Listen Mode	
9.7.4.1	The Poller that is sending a Special Frame SHALL wait $\mathbf{FDT}_{V,EOF}$ between the end of the Standard Frame and the start of the Isolated EoF Frame. Appendix B.5 lists the value of $\mathbf{FDT}_{V,EOF}$.	9.7.4.2	When it is receiving a Special Frame, the Listener SHALL be ready to receive the start of the Isolated EOF Frame no later than $\mathbf{FDT}_{V,EOF}$ after the end of the Standard Frame. When it is receiving a Special Frame, if the start of the Isolated EoF Frame is received before a time $\mathbf{FDT}_{V,EOF}$ after the end of the Standard Frame, the Listener MAY treat this receipt as a Transmission Error.

9.7.5 Guard Time

This section specifies the guard time of an Unmodulated Carrier, after which the Listener is ready to receive an INVENTORY_REQ Command.

Requirements 143: NFC-V Guard Time

Listen Mode	
9.7.5.1	When a Listener is exposed to an Unmodulated Carrier (see [ANALOG]), it SHALL be ready to receive an INVENTORY_REQ Command after a guard time $\mathbf{GT}_V$. Appendix B.5 lists the value of $\mathbf{GT}_V$.

9.7.6 Response Grid

This section specifies the requirements for the Response Timing for Write-Alike Commands.

The Write-Alike Response Grid is defined as a duration starting at the detection of the rising edge of the EoF of a Poll Frame and continuing with a delay of $4352/\mathbf{f}_C$ plus a multiple of $4096/\mathbf{f}_C$, with a tolerance of $32/\mathbf{f}_C$. The Write-Alike Response Grid is always lower than 20 ms. The Response Grid is defined:

Response Grid = $(4352/\mathbf{f}_C + n \times 4096/\mathbf{f}_C) \pm 32/\mathbf{f}_C$, where

- n is an integer.
- The Response Grid is always less than 20 ms.

Requirements 144: Response Grid

Poll Mode		Listen Mode	
9.7.6.1	For Write-Alike Commands the Poller SHALL be ready to receive a Listen Frame on the Response Grid and MAY process a Listen Frame received outside the Response Grid.	9.7.6.2	The Listener SHALL send the Listen Frame containing its Response to Write-Alike Commands on the Response Grid.

10 Half-duplex Protocols

For NFC Forum Devices, data are transmitted using half-duplex protocols. These protocols share common characteristics that are described in this section. Specific characteristics of each half-duplex protocol, such as timeouts and exception handling, are described in the following sections.

The term “half-duplex” means that only one device sends data at a time. Specifically, in NFC Forum Devices there is a master-slave relationship between each pair of devices. If the NFC Forum Device is in Poll Mode, it is the master; if the NFC Forum Device is in Listen Mode, it is the slave.

The Poller sends a single Poll Frame and then waits for a Listen Frame. The Listener waits for a Poll Frame before it responds with a single Listen Frame.

Requirements 145: General Rules for Half-duplex Transmission Protocol

Poll Mode		Listen Mode	
10.1.1.1	After it sends a Poll Frame, the Poller SHALL switch to receive mode and wait for a Listen Frame.	10.1.1.2	After it has been activated in Listen Mode, the Listener SHALL wait for a Poll Frame.
10.1.1.3	The Poller SHALL NOT send another Poll Frame until it has received a Listen Frame or until a timeout has occurred before any Listen Frame has been received.	10.1.1.4	When it has received a valid Poll Frame, the Listener SHALL send a Listen Frame, unless its actions are explicitly defined otherwise. After responding, the Listener SHALL return to receive mode.

11 VOID

12 Type 2 Tag Platform

12.1 Sequence Format

The Type 2 Tag Platform uses the NFC-A synchronization mechanism.

Requirements 146: Type 2 Tag Platform Sequence Format

Poll and Listen Mode	
12.1.1.1	Type 2 Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology sequence Format, as defined in Section 6.1.

12.2 Bit Level Coding

The Type 2 Tag Platform uses NFC-A bit level coding.

Requirements 147: Type 2 Tag Platform Bit Level Coding

Poll and Listen Mode	
12.2.1.1	Type 2 Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology bit level coding defined in Section 6.2.

12.3 Frame Format

The Type 2 Tag Platform transmits Commands and Responses in NFC-A Technology Standard Frames (defined in Section 6.3.3), except for the ACK and NACK Response.

A Listen Frame for the ACK and NACK Response consists of a Short Frame with 4 data bits (see Section 6.3.2).

Requirements 148: Type 2 Tag Platform Frame Format

Poll and Listen Mode	
12.3.1.1	Type 2 Tag Platform Commands and Responses SHALL be sent and received using either: • The NFC-A Technology Standard Frame (defined in Section 6.3.3) • The T2T Short Frame defined as an NFC-A Technology Short Frame (defined in Section 6.3.2) that contains 4 data bits.

12.4 Data and Payload Format

Type 2 Tag Platform data transmitted in an NFC-A Technology Standard Frame (i.e., the bytes following the SoF) consist of the payload and, depending on the payload, the EoD.

The payload consists of the Commands and Responses described in Section 12.5.

If present, the EoD contains a 2-byte checksum referred to as “CRC_A”. The input for the CRC_A calculation is the payload. If the payload consists of the ACK or NACK Response, then the EoD is not present.

Figure 37 shows the Type 2 Tag Platform data and payload format.

Data					
Payload				**EoD**	
Byte 1	Byte 2	…	Byte n	CRC_A1	CRC_A2

Figure 37: Type 2 Tag Platform (except for ACK and NACK Response) Data and Payload Format

Requirements 149: 4 Data Bit Response Frame Format

Poll Mode	
12.4.1.1	All 4 data bit Responses (ACK and NACK Responses) SHALL be sent and received in frames that use the T2T Short Frame format defined in Section 12.3.

Requirements 150: Type 2 Tag Platform Data and Payload Format

Poll and Listen Mode	
12.4.1.2	Type 2 Tag Platform Commands and Responses SHALL be sent and received in frames, as defined in Section 12.3.
12.4.1.3	If the payload consists of a Command or Response different from the ACK or NACK Response, the payload SHALL be followed by an EoD at the position, as indicated in Figure 37. The EoD SHALL contain a CRC_A. Otherwise the EoD SHALL NOT be present.
12.4.1.4	The CRC_A SHALL be calculated and verified as specified in Section 6.4.

12.5 Command Set

A Type 2 Tag Platform can be addressed for the applicable data exchange directly after its activation. The Type 2 Tag Platform can be activated using the Commands defined in Section 6.5. There are no additional Commands for the Type 2 Tag Platform activation. Type 2 Tag specific Commands are defined in [T2T].

Requirements 151: Type 2 Tag Platform Command Set

Poll and Listen Mode	
12.5.1.1	Type 2 Tag Platform Commands and Responses SHALL be compliant with the requirements defined for half-duplex protocols in Section 10.
12.5.1.2	Type 2 Tag Platform Commands and Responses SHALL be sent and received as the payload of the data and payload format defined in Section 12.4.

12.6 Error Handling

When errors are detected the following error handling is attempted.

Requirements 152: Type 2 Tag Platform Error Handling

Poll Mode	
12.6.1.1	If there is a Transmission Error in response to a READ or WRITE Command, as defined in [T2T], the Reader/Writer SHALL attempt error recovery by resending the Command until either a Valid Response frame is received or until a maximum of consecutive $\mathbf{n}_{T2T,RETRY,ERROR}$ retries of the same Command has been performed. If no Valid Response is received, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception.
12.6.1.2	If there is a Protocol Error or a Timeout Error in response to a READ, WRITE or SECTOR SELECT Command, as defined in [T2T], the Reader/Writer SHALL perform either of the following at least once: • Device Activation and Data Exchange Activity, as defined in [ACTIVITY]. Before executing the Device Activation Activity the Reader/Writer SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 1b for the Device which caused the error. • Operating Field Reset followed by the Technology Detection, Collision Resolution, Device Activation and Data Exchange Activity, as defined in [ACTIVITY].
NOTE	In case of such an error the T2T moves from **`T2T_ACTIVE*`** or **`SECTOR_SELECT*`** States to **`SLEEP_A`** State or from **`T2T_ACTIVE`** or **`SECTOR_SELECT`** States to **`IDLE`** State (see [T2T] for more details).

13 Type 3 Tag Platform

13.1 Sequence Format

The Type 3 Tag Platform uses the NFC-F synchronization mechanism.

Requirements 153: Type 3 Tag Platform Sequence Format

Poll and Listen Mode	
13.1.1.1	Type 3 Tag Platform Commands and Responses SHALL be sent and received using the NFC-F Technology sequence format defined in Section 8.1.

13.2 Bit Level Coding

The Type 3 Tag Platform uses NFC-F bit level coding.

Requirements 154: Type 3 Tag Platform Bit Level Coding

Poll and Listen Mode	
13.2.1.1	Type 3 Tag Platform Commands and Responses SHALL be sent and received using the NFC-F Technology bit level coding defined in Section 8.2.

13.3 Frame Format

The Type 3 Tag Platform transmits Commands and Responses in NFC-F frames.

Requirements 155: Type 3 Tag Platform Frame Format

Poll and Listen Mode	
13.3.1.1	Type 3 Tag Platform Commands and Responses SHALL be sent and received using the NFC-F Technology frame defined in Section8.3.

13.4 Data and Payload Format

Type 3 Tag Platform data follow the data and payload format specified in Section 8.4 for NFC-F Technology.

Requirements 156: Type 3 Tag Platform Data and Payload Format

Poll and Listen Mode	
13.4.1.1	Type 3 Tag Platform Commands and Responses SHALL be sent and received using the NFC-F Technology data and payload format defined in Section 8.4.

13.5 Command Set

A Type 3 Tag Platform can be addressed for the applicable data exchange directly after its activation. The Type 3 Tag Platform can be activated using the Commands defined in Section 8.5. There are no additional Commands for the Type 3 Tag Platform activation. Type 3 Tag specific Commands are defined in [T3T].

Requirements 157: Type 3 Tag Platform Command Set

Poll and Listen Mode	
13.5.1.1	Type 3 Tag Platform Commands and Responses SHALL be compliant with the requirements defined for half-duplex protocols in Section 10.
13.5.1.2	Type 3 Tag Platform Commands and Responses SHALL be sent and received as the payload of the data and payload format defined in Section 13.4.

14 Type 4A Tag Platform

This section is derived from [ISO/IEC_14443].

14.1 Sequence Format

The Type 4A Tag Platform uses the NFC-A synchronization mechanism.

Requirements 158: Type 4A Tag Platform Sequence Format

Poll and Listen Mode	
14.1.1.1	Type 4A Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology sequence format defined in Section 6.1.

14.2 Bit Level Coding

The Type 4A Tag Platform uses the NFC-A bit level coding.

Requirements 159: Type 4A Tag Platform Bit Level Coding

Poll and Listen Mode	
14.2.1.1	Type 4A Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology bit level coding defined in Section 6.2.

14.3 Frame Format

The Type 4A Tag Platform transmits Commands and Responses in NFC-A Technology Standard Frames.

Requirements 160: Type 4A Tag Platform Frame Format

Poll and Listen Mode	
14.3.1.1	Type 4A Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology Standard Frame defined in Section 6.3.

14.4 Data and Payload Format

The payload consists of the Commands and Responses described in Section 14.5.

Requirements 161: Type 4A Tag Platform Data and Payload Format

Poll and Listen Mode	
14.4.1.1	Type 4A Tag Platform Commands and Responses SHALL be sent and received using the NFC-A Technology data and payload format defined in Section 6.4.
14.4.1.2	The payload SHALL be followed by an EoD at the position indicated in Figure 6. The EoD SHALL contain a CRC_A.
14.4.1.3	The CRC_A SHALL be calculated and verified as specified in Section 6.4.

14.5 Command Set

A Type 4A Tag Platform can be addressed for data exchange directly after its activation.

This Section defines the Commands and Responses listed in Table 53, which are required to activate the Type 4A Tag Platform in addition to the Commands defined in Section 6.5.

Table 53: Type 4A Tag Platform Command Set

Poll Mode (Command)	Listen Mode (Response)
RATS Command	RATS Response (ATS)

14.5.1 General Rules

The Type 4A Tag Platform Commands and Responses are to be transmitted according to the rules defined for half-duplex protocols in Section 10.

Requirements 162: Type 4A Tag Platform General Rules

Poll and Listen Mode	
14.5.1.1	Type 4A Tag Platform Commands and Responses SHALL be compliant with the requirements defined for half-duplex protocols in Section 10.
14.5.1.2	Type 4A Tag Platform Commands and Responses SHALL be sent and received as the payload of the data and payload format defined in Section 14.4.

14.6 Request for Answer to Select (RATS)

The RATS Command is used by the Reader/Writer during the Device Activation Activity to negotiate the maximum frame size and the bit rate divisors (D) with the Listener.

14.6.1 RATS Command

Table 54 specifies the code for the RATS Command.

Table 54: RATS Command Format

Byte 1	Byte 2
E0h	PARAM

Requirements 163: RATS Command

Poll Mode		Listen Mode	
14.6.1.1	The Reader/Writer SHALL send the RATS Command that has the format specified in Table 54.	14.6.1.2	The Listener SHALL be ready to receive a RATS Command that has the format specified in Table 54.

PARAM, the parameter byte, consists of two parts, as shown in Table 55.

Table 55: RATS Parameter Byte (PARAM) Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x					FSDI
				x	x	x	x	DID

The most significant nibble b8 through b5 of PARAM is called “FSDI” (Frame Size for proximity coupling Device Integer) and provides a code for the FSD (Frame Size for proximity coupling Device). The FSD defines the maximum size of a frame that the Reader/Writer is able to receive. FSD is expressed in number of data bytes included in the frame.

Requirements 164: RATS Parameter Byte (PARAM)

Poll Mode	
14.6.1.3	The FSDI and DID fields SHALL be formatted and interpreted as defined in Table 55.

Requirements 165: Frame Size for Reader/Writer (FSD)

Poll Mode		Listen Mode	
14.6.1.4	The Reader/Writer SHALL accept frames that contain a number of data bytes less than or equal to FSD. If the Reader/Writer receives a frame with more than FSD data bytes, it SHALL treat that frame as a Syntax Error.	14.6.1.5	The Listener SHALL send frames with a number of data bytes less than or equal to FSD.
14.6.1.6	The FSD supported by the Reader/Writer SHALL be $\mathbf{FSD}_{T4AT,MIN}$. Appendix B.6 lists the value of $\mathbf{FSD}_{T4AT,MIN}$.	14.6.1.7	The Listener SHALL support an FSD equal to $\mathbf{FSD}_{T4AT,MIN}$.

Table 56 provides the coding of the FSDI to the FSD values.

Table 56: FSDI to FSD Conversion

FSDI	FSD (bytes)
0h	16
1h	24
2h	32
3h	40
4h	48
5h	64
6h	96
7h	128
8h	256
9h	512
Ah	1024
Bh	2048
Ch	4096
Dh – Fh	RFU

Requirements 166: FSDI

Poll Mode	
14.6.1.8	The Reader/Writer SHALL set the FSDI greater than or equal to the value corresponding to $\mathbf{FSD}_{T4AT,MIN}$. Appendix B.6 lists the value of $\mathbf{FSD}_{T4AT,MIN}$. Table 56 specifies the FSDI to FSD conversion.

Requirements 167: Listen Mode Handling of FSDI with Value RFU

Listen Mode	
14.6.1.9	A received FSDI value of RFU SHALL be treated by the Listener as FSDI=Ch.

The least significant nibble b4 to b1 of PARAM is named “DID” (Device Identification number), and defines the logical number of the addressed Listener (see Table 57).

Table 57: DID Coding

b4	b3	b2	b1	Meaning
0	0	0	0	Logical number=0
0	0	0	1	Logical number =1
0	0	1	0	Logical number =2
0	0	1	1	Logical number =3
0	1	0	0	Logical number =4
0	1	0	1	Logical number =5
0	1	1	0	Logical number =6
0	1	1	1	Logical number =7
1	0	0	0	Logical number =8
1	0	0	1	Logical number =9
1	0	1	0	Logical number =10
1	0	1	1	Logical number =11
1	1	0	0	Logical number =12
1	1	0	1	Logical number =13
1	1	1	0	Logical number =14
1	1	1	1	RFU

NOTE The term “CID” used in [ISO/IEC_14443] is referred to as “DID” in this document.

Requirements 168: DID Support

Poll Mode		Listen Mode	
14.6.1.10	The Reader/Writer SHALL set the DID field to a value in the range 0 to 14.	14.6.1.11	The Listener SHALL be ready to receive a DID field (as defined in Table 57) with a value in the range 0 to 14.
14.6.1.12	The Reader/Writer SHOULD assign a DID to 0 during communication if it intends to communicate with a single device in Listen Mode.	14.6.1.13	The Listener SHALL treat as a Syntax Error a DID field that has the RFU value of 15.

14.6.2 RATS Response (Answer To Select)

The RATS Response, also called "Answer To Select" (ATS), is transmitted by the Listener in response to the RATS Command. This section defines the RATS Response with all of its available fields, as shown in Table 58.

Table 58: RATS Response Format

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6 – 6+k-1
TL	T0	TA(1)	TB(1)	TC(1)	T1 … Tk

Requirements 169: RATS Response

Poll Mode		Listen Mode	
14.6.2.1	The Reader/Writer SHALL be ready to receive a RATS Response that has the format specified in Table 58.	14.6.2.2	The Listener SHALL send the RATS Response formatted as specified in Table 58.

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

- Format byte T0
- Interface bytes TA(1), TB(1), TC(1)
- Historical bytes T1 to Tk.

The parameter k denotes the number of historical bytes.

- Length byte

  The length byte TL is mandatory and specifies the length of the RATS Response, including the TL byte itself.

Requirements 170: RATS Response Length Byte TL

Poll Mode		Listen Mode	
14.6.2.3	The Reader/Writer SHALL treat an ATS with a TL value that is inconsistent with the length of the ATS itself as a Syntax Error.	14.6.2.4	The first byte (TL) of the RATS Response SHALL specify the length of the RATS Response, including the TL byte.
14.6.2.5	The Reader/Writer SHALL be ready to receive a RATS Response with a TL that specifies a length less than or equal to 20 bytes. The Reader/Writer MAY accept a RATS Response with a TL that indicates a length greater than 20 bytes.	14.6.2.6	The TL SHALL NOT indicate a length greater than 20 bytes.

- Format Byte T0

  The format byte T0 is coded as specified in Table 59.

Table 59: Format Byte T0 Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	x							TC(1) is transmitted, if bit is set to 1
		x						TB(1) is transmitted, if bit is set to 1
			x					TA(1) is transmitted, if bit is set to 1
				x	x	x	x	FSCI

Requirements 171: RATS Response Format Byte T0

Poll Mode		Listen Mode	
14.6.2.7	The Reader/Writer SHALL be ready to receive a Format Byte T0 that has the format specified in Table 59.	14.6.2.8	The Listener SHALL format the Format Byte T0 according to Table 59.

The least significant nibble, b4 through b1, is the Frame Size for proximity Card Integer (FSCI), which codes the Frame Size for proximity Card (FSC). The FSC defines the maximum size of a frame accepted by the Listener. FSC is expressed as the number of data bytes included in the frame.

Requirements 172: FSC

Poll Mode		Listen Mode	
14.6.2.9	The Reader/Writer SHALL send frames with a number of data bytes less than or equal to FSC.	14.6.2.10	The Listener SHALL accept frames with a number of data bytes less than or equal to FSC. If it receives a frame with more than FSC data bytes, the Listener MAY treat the frame as a Syntax Error.
14.6.2.11	The Reader/Writer SHALL be capable of sending frames in accordance with an FSC greater than or equal to $\mathbf{FSC}_{T4AT,MIN}$.	14.6.2.12	The FSC supported by the Listener SHALL be at least $\mathbf{FSC}_{T4AT,MIN}$. Appendix B.6 lists the value of $\mathbf{FSC}_{T4AT,MIN}$.

NOTE If blocks that contain more than FSC bytes are accepted according to requirement 14.6.2.10, they need to be handled correctly.

Table 60 specifies the FSCI coding for the FSC values. The default value of FSCI is 2, which of course indicates an FSC of 32 bytes.

Table 60: FSCI to FSC Conversion

FSCI	FSC (bytes)
0h	16
1h	24
2h	32
3h	40
4h	48
5h	64
6h	96
7h	128
8h	256
9h	512
Ah	1024
Bh	2048
Ch	4096
Dh – Fh	RFU

Requirements 173: FSCI

Listen Mode	
14.6.2.13	The Listener SHALL set FSCI greater than or equal to the value corresponding to $\mathbf{FSC}_{T4AT,MIN}$. Appendix B.6 lists the value of $\mathbf{FSC}_{T4AT,MIN}$. Table 60 specifies the FSCI to FSC conversion.

Requirements 174: Poll Mode Handling of FSCI with Value RFU

Poll Mode	
14.6.2.14	A received FSCI value of RFU SHALL be treated by the Reader/Writer as FSCI=Ch.

Requirements 175: RATS Response Format Byte T0

Poll Mode		Listen Mode	
14.6.2.15	The Reader/Writer SHALL be ready to receive a RATS Response including T0, TA(1), TB(1), and TC(1). If one or more of the fields T0, TA(1), TB(1), and TC(1) are missing, then the Reader/Writer SHALL use the default values specified in this section.	14.6.2.16	TA(1), TB(1) and TC(1) SHALL be present in the RATS Response and their presence SHALL be indicated in T0.

- Interface Byte TA(1)

  The interface byte TA(1) conveys information to define the bit rate capabilities of the Listener. The interface byte TA(1) is coded as specified in Table 61. The bits b7 through b5 code the bit rate capability of the Listener for the direction from Listener to Reader/Writer ($\mathbf{D}_{LISTEN \rightarrow POLL}$). The default value for bits b7 through b5 is 000b ($\mathbf{D}_{LISTEN \rightarrow POLL}$ = 1). The bits b3 through b1 code the bit rate capability of the Listener for the direction from Reader/Writer to Listener ($\mathbf{D}_{POLL \rightarrow LISTEN}$). The default value for bits b3 through b1 is 000b ($\mathbf{D}_{POLL \rightarrow LISTEN}$ = 1).

Table 61: Interface Byte TA(1) Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x								If b8 = 1b, then only the same bit rate divisor for both directions is supported ($\mathbf{D}_{LISTEN \rightarrow POLL} = \mathbf{D}_{POLL \rightarrow LISTEN}$). If b8 = 0b, then a different bit rate divisor for each direction is supported.
	x							$\mathbf{D}_{LISTEN \rightarrow POLL}$ = 8 supported, if bit is set to 1b.
		x						$\mathbf{D}_{LISTEN \rightarrow POLL}$ =4 supported, if bit is set to 1b.
			x					$\mathbf{D}_{LISTEN \rightarrow POLL}$ =2 supported, if bit is set to 1b.
				0				RFU
					x			$\mathbf{D}_{POLL \rightarrow LISTEN}$ =8 supported, if bit is set to 1b.
						x		$\mathbf{D}_{POLL \rightarrow LISTEN}$ =4 supported, if bit is set to 1b.
							x	$\mathbf{D}_{POLL \rightarrow LISTEN}$ =2 supported, if bit is set to 1b.

Requirements 176: RATS Response Format Byte TA(1)

Poll Mode		Listen Mode	
14.6.2.17	The Reader/Writer SHALL be ready to receive any TA(1) value (see Table 61).	14.6.2.19	The Listener MAY set the bits b7 through b5 and b3 through b1 of TA(1) to any value specified in Table 61.
14.6.2.18	The Reader/Writer SHALL support a bit rate of 106 kbits/s in both directions. The Reader/Writer MAY support higher bit rates indicated by TA(1).		

NOTE Bit rates above 106 kbits/s for NFC-A are not defined by this version of the specification.

Requirements 177: Poll Mode Handling of b4 in Interface Byte TA(1) with Value RFU

Poll Mode	
14.6.2.20	A received b4 value of RFU SHALL be interpreted by the Reader/Writer as if bits b8 through b1 = 00000000b (only 106 kbits/s in both directions).

- Interface Byte TB(1)

  The interface byte TB(1) conveys information to define the Frame Waiting Time (FWT) and the Start-up Frame Guard Time (SFGT). The interface byte TB(1) is coded as specified in Table 62.

Table 62: Interface Byte TB(1) Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x					FWI
				x	x	x	x	SFGI

The most significant nibble b8 through b5 is called FWI (Frame Waiting time Integer) and codes the Frame Waiting Time (FWT) for the subsequent Responses. Section 14.8 defines the FWT. FWI has a value in the range from 0 to 14. The value 15 is RFU. The default value of FWI is 4.

Requirements 178: RATS Response Interface Byte TB(1)

Poll Mode		Listen Mode	
14.6.2.21	The Reader/Writer SHALL be ready to accept the Interface Byte TB(1) that has the format specified in Table 62.	14.6.2.22	The Listener SHALL format the Interface Byte TB(1) according to Table 62.
14.6.2.23	The Reader/Writer SHALL support a RATS Response with TB(1) indicating an FWI less than or equal to $\mathbf{FWI}_{T4AT,MAX}$.	14.6.2.24	The Listener SHALL set FWI less than or equal to $\mathbf{FWI}_{T4AT,MAX}$. Appendix B.6 lists the value of $\mathbf{FWI}_{T4AT,MAX}$.

Requirements 179: Handling of FWI Value of RFU

Poll Mode	
14.6.2.25	A received FWI value of RFU SHALL be treated by the Reader/Writer as FWI=4.

The least significant nibble b4 through b1 codes SFGI (Start-up Frame Guard time Integer) and is used by the Listener to code a multiplier value used to define the SFGT. Section 14.8 defines SFGT. SFGI has a value in the range from 0 to 14. The value 15 is RFU. The default value of SFGI is 0.

Requirements 180: RATS Response Interface Byte TB(1)

Poll Mode		Listen Mode	
14.6.2.26	The Reader/Writer SHALL be ready to receive a RATS Response with TB(1) indicating an SFGI less than or equal to $\mathbf{SFGI}_{T4AT,MAX}$.	14.6.2.27	The Listener SHALL set SFGI less than or equal to $\mathbf{SFGI}_{T4AT,MAX}$. Appendix B.6 lists the value of $\mathbf{SFGI}_{T4AT,MAX}$.

Requirements 181: Handling of SFGI with Value RFU

Poll Mode	
14.6.2.28	A received SFGI value of RFU SHALL be treated by the Reader/Writer as SFGI=0.

- Interface Byte TC(1)

  The interface byte TC(1) indicates whether NAD and DID are supported by the Listener. The interface byte TC(1) is coded as specified in Table 63.

Table 63: Interface Byte TC(1) Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				0				RFU
					0			RFU
						x		DID supported, if bit is set to 1b
							x	NAD supported, if bit is set to 1b

Requirements 182: RATS Response Interface Byte TC(1)

Poll Mode		Listen Mode	
14.6.2.29	The Reader/Writer SHALL be ready to accept the Interface Byte TC(1) that has the format specified in Table 63.	14.6.2.30	The Listener SHALL format the Interface Byte TC(1) according to Table 63.

The bits b2 and b1 are used by the Listener to indicate which optional fields in the SoD of ISO-DEP Block it supports. Section 16.1 specifies the SoD. The default value for b2 is 1b (DID supported). The default value for b1 is 0b (NAD not supported).

- Historical Bytes

  The historical bytes T1 to Tk are optional and are used by the Listener to designate general information. The maximum length of the RATS Response gives the maximum possible number of historical bytes.

Requirements 183: RATS Response Historical Bytes

Poll Mode		Listen Mode	
14.6.2.31	The Reader/Writer SHALL be ready to receive a RATS Response with up to 15 historical bytes. The Reader/Writer MAY accept a RATS Response with more than 15 historical bytes.	14.6.2.32	The Listener SHALL send no more than 15 historical bytes.

14.7 Error Handling

When errors are detected the following error handling is attempted.

Requirements 184: Type 4A Tag Platform Error Handling

Poll Mode		Listen Mode	
14.7.1.1	If a Timeout Error occurs after the transmission of the RATS Command, the Reader/Writer SHALL attempt error recovery by resending the RATS Command until a Valid Response is received, or until a maximum of $\mathbf{n}_{T4AT,RETRY,RATS}$ error recovery RATS Commands have been sent. • If for every RATS Command a Timeout Error is detected, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. • If the received Response is not a Valid Response, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception. Appendix B.6 lists the values of $\mathbf{n}_{T4AT,RETRY,RATS}$.	14.7.1.2	The Listener SHALL detect Transmission Errors and Protocol Errors.
14.7.1.3	If a Transmission Error occurs after the transmission of the RATS Command, the Reader/Writer SHALL raise the Unrecoverable Transmission Exception.	14.7.1.4	The Listener SHALL NOT attempt error recovery. The Listener SHALL always stay in receive mode when a Transmission Error or a Protocol Error occurs.

NOTE Transmission Errors on the Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

14.8 Timing Requirements

14.8.1 FWT

The Frame Waiting Time (FWT) defines the time within which a Listener configured for the Type 4A Tag Platform has to start its Response after the end of a Poll Frame. The FWT is calculated using the following formula:

$$FWT = (256 \times 16/\mathbf{f}_C) \times 2^{FWI}$$

where the value of FWI has the range from 0 to 14 and is provided by the RATS Response specified in Section 14.6.2.

The RATS Command includes a definition for a specific FWT. For this Command the Listener starts sending its Response frame within $\mathbf{FWT}_{T4AT,ACTIVATION}$ (activation frame waiting time). Section 14.6.1 defines the RATS Command.

Requirements 185: Type 4A Tag Platform Frame Timing

Poll Mode		Listen Mode	
14.8.1.1	Following the RATS Command, the Reader/Writer SHALL wait at least $\mathbf{FWT}_{T4AT,ACTIVATION}$ for the RATS Response. If the Reader/Writer does not receive the RATS Response within the period $\mathbf{FWT}_{T4AT,ACTIVATION} + \Delta\mathbf{FWT}_{T4AT} + \Delta\mathbf{T}_{T4AT,POLL}$, then the Reader/Writer SHALL treat the lack of response as a Timeout Error (i.e., $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FWT}_{T4AT,ACTIVATION} + \Delta\mathbf{FWT}_{T4AT} + \Delta\mathbf{T}_{T4AT,POLL}$). Between the ends of the periods $\mathbf{FWT}_{T4AT,ACTIVATION}$ and $\mathbf{FWT}_{T4AT,ACTIVATION} + \Delta\mathbf{FWT}_{T4AT} + \Delta\mathbf{T}_{T4AT,POLL}$, the Reader/Writer MAY accept the RATS Response or MAY generate a Timeout Error. Appendix B.6 lists the values of $\Delta\mathbf{FWT}_{T4AT}$ and $\Delta\mathbf{T}_{T4AT,POLL}$.	14.8.1.2	The Listener SHALL start the RATS Response after the end of the RATS Command within the period $\mathbf{FWT}_{T4AT,ACTIVATION}$ (i.e., $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FWT}_{T4AT,ACTIVATION}$). Appendix B.6 lists the value of $\mathbf{FWT}_{T4AT,ACTIVATION}$.

Poll Mode		Listen Mode	
14.8.1.3	Except for the RATS Command, the Reader/Writer SHALL wait at least FWT + ΔFWT_{T4AT} for a Response. Except for the RATS Command, if the Reader/Writer does not receive a Response within FWT + ΔFWT_{T4AT} + $\Delta T_{T4AT,POLL}$, then the Reader/Writer SHALL treat the lack of response as a Timeout Error (i.e., $FDT_{A,LISTEN,MAX}$ equals FWT + ΔFWT_{T4AT} + $\Delta T_{T4AT,POLL}$). During the interval between the end of the time period FWT + ΔFWT_{T4AT} and the end of the time period FWT + ΔFWT_{T4AT} + $\Delta T_{T4AT,POLL}$, the Reader/Writer MAY accept the Response or MAY generate a Timeout Error. Appendix B.6 lists the values of ΔFWT_{T4AT} and $\Delta T_{T4AT,POLL}$.	14.8.1.4	Except for the RATS Command, the Listener SHALL start its Response after the end of a Command within the FWT (i.e., $FDT_{A,LISTEN,MAX}$ equals FWT).

14.8.2 SFGT

The SFGT is the minimum time the Reader/Writer is waiting before the Listener is ready to receive the next frame after it has sent the RATS Response. Section 14.6.2 defines the RATS Response.

The SFGT is calculated:

$$SFGT = (256 \times 16/\mathbf{f}_C) \times 2^{SFGI}$$

where SFGI has the range from 0 to 14 and is returned by the Listener in the interface byte TB(1) of the RATS Response. If the Listener returns SFGI equal to 0 or SFGI is not returned, then no SFGT is needed and $\mathbf{FDT}_{A,POLL,MIN}$ applies.

The requirements are defined with a margin $\mathbf{\Delta SFGT}_{T4AT}$ that allows the Reader/Writer to receive a frame within the defined waiting time plus the margin. The $\mathbf{\Delta SFGT}_{T4AT}$ is calculated by the following formula:

$$\mathbf{\Delta SFGT}_{T4AT} = 384 \times 2^{SFGI}$$

Requirements 186: Type 4A Tag Platform SFGT

Poll Mode		Listen Mode	
14.8.2.1	If the Listener returns an SFGI different from 0, then after the Listener has sent the RATS Response, the Reader/Writer SHALL wait at least **SFGT** + $\mathbf{\Delta SFGT}_{T4AT}$ before it sends the next frame. **SFGT** + $\mathbf{\Delta SFGT}_{T4AT}$ is computed from SFGI as defined above. Appendix B.7 lists the value of $\mathbf{SFGI}_{T4AT,MAX}$.	14.8.2.2	If the Listener returns an SFGI different from 0, then it SHALL be ready to receive the start of a new Poll Frame no later than **SFGT** after the end of the RATS Response frame. If the start of a new Poll Frame is received before **SFGT**, then the Listener MAY treat this receipt as a Transmission Error. **SFGT** is computed from SFGI as defined above. Appendix B.7 lists the value of $\mathbf{SFGI}_{T4AT,MAX}$.

Poll Mode		Listen Mode	
14.8.2.3	If the Listener returns an SFGI equal to 0 or SFGI is not returned, then, after the Listener has sent the RATS Response, the Reader/Writer SHALL wait at least $\mathbf{FDT}_{A,POLL,MIN}$ before it sends the next frame.	14.8.2.4	If the Listener returns an SFGI equal to 0 or does not return an SFGI, then it SHALL be ready to receive the start of a new Poll Frame no later than $\mathbf{FDT}_{A,POLL,MIN}$ after the end of the RATS Response frame. If the start of a new Poll Frame is received before $\mathbf{FDT}_{A,POLL,MIN}$, then the Listener MAY treat this receipt as a Transmission Error.

15 Type 4B Tag Platform

This section is derived from [ISO/IEC_14443].

15.1 Sequence Format

The Type 4B Tag Platform uses the NFC-B synchronization mechanism.

Requirements 187: Type 4B Tag Platform Sequence Format

Poll and Listen Mode	
15.1.1.1	Type 4B Tag Platform Commands and Responses SHALL be sent and received using the NFC-B Technology sequence format defined in Section 7.1.

15.2 Bit Level Coding

The Type 4B Tag Platform uses NFC-B bit level coding.

Requirements 188: Type 4B Tag Platform Bit Level Coding

Poll and Listen Mode	
15.2.1.1	Type 4B Tag Platform Commands and Responses SHALL be sent and received using the NFC-B Technology bit level coding defined in Section 7.2.

15.3 Frame Format

The Type 4B Tag Platform transmits Commands and Responses in NFC-B frames.

Requirements 189: Type 4B Tag Platform Frame Format

Poll and Listen Mode	
15.3.1.1	Type 4B Tag Platform Commands and Responses SHALL be sent and received using the NFC-B Technology frames defined in Section 7.3.

15.4 Data and Payload Format

The payload consists of the Commands and Responses described in Section 15.5.

Requirements 190: Type 4B Tag Platform Data and Payload Format

Poll and Listen Mode	
15.4.1.1	Type 4B Tag Platform Commands and Responses SHALL be sent and received using the NFC-B Technology data and payload format defined in Section 7.4.

15.5 Command Set

A Type 4B Tag Platform can be addressed for data exchange directly after its activation.

This Section defines Commands and Responses listed in Table 64, which are required to activate the Type 4B Tag Platform in addition to the Commands defined in Section 6.5.

Table 64: Type 4B Tag Platform Command Set

Poll Mode (Command)	Listen Mode (Response)
ATTRIB Command	ATTRIB Response

15.5.1 General Rules

The Type 4B Tag Platform Commands and Responses are to be transmitted according to the rules defined for half-duplex protocols in Section 10.

Requirements 191: Type 4B Tag Platform General Rules

Poll and Listen Mode	
15.5.1.1	Type 4B Tag Platform Commands and Responses SHALL be compliant with the requirements defined for half-duplex protocols in Section 10.
15.5.1.2	Type 4B Tag Platform Commands and Responses SHALL be sent and received as the payload of the data and payload format defined in 15.4.

15.6 ATTRIB

The ATTRIB Command is sent by the Reader/Writer during the Device Activation Activity in order to negotiate a set of communication parameters with the Listener.

15.6.1 ATTRIB Command

Table 65 defines the format of the ATTRIB Command.

Table 65: ATTRIB Command Format

Byte 1	Byte 2-5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10 – 10+$k_{T4BT,MAX,ATTRIB}$-1
1Dh	NFCID0	Param 1	Param 2	Param 3	Param 4	Higher layer – INF

Requirements 192: ATTRIB Command Format

Poll Mode		Listen Mode	
15.6.1.1	The Reader/Writer SHALL send the ATTRIB Command formatted as specified in Table 65.	15.6.1.2	The Listener SHALL be ready to receive an ATTRIB Command that has the format specified in Table 65.

NFCID0

Byte 2 through byte 5 include the NFCID0 sent by the Listener in the SENSB_RES.

Requirements 193: NFCID0 in the ATTRIB Command

Poll Mode		Listen Mode	
15.6.1.3	The Reader/Writer SHALL send the ATTRIB Command including the NFCID0 received in the valid SENSB_RES Response from the Listener.	15.6.1.4	The Listener SHALL recognize its own NFCID0 and respond only to a valid ATTRIB Command in which its NFCID0 is included.

Param 1

The Reader/Writer codes Param 1 with Minimum $TR0_B$ and Minimum $TR1_B$, and indicates whether SoS and EoS are allowed to be suppressed. The format of Param 1 is specified in Table 66.

Table 66: ATTRIB Command Param 1 Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x							Minimum $TR0_B$ 00b: $\mathbf{TR0}_{B,MIN,DEFAULT}$ 01b: $48 \times 16/\mathbf{f}_C$ 10b: $16 \times 16/\mathbf{f}_C$ 11b: RFU
		x	x					Minimum $TR1_B$ 00b: $\mathbf{TR1}_{B,MIN,DEFAULT}$ 01b: $64 \times 16/\mathbf{f}_C$ 10b: $16 \times 16/\mathbf{f}_C$ 11b: RFU
				x				Suppression of EoS 0b: EoS suppression not supported 1b: EoS suppression supported
					x			Suppression of SoS 0b: SoS suppression not supported 1b: SoS suppression supported
						0		RFU
							0	RFU

Requirements 194: ATTRIB Command Param 1 Format

Poll Mode		Listen Mode	
15.6.1.5	The Reader/Writer MAY set Minimum $TR0_B$ to any value specified for it in Table 66. For further communication the Reader/Writer SHALL be ready to receive a subcarrier no later than $TR0_{B,MIN}$, which is set to Minimum $TR0_B$.	15.6.1.6	The Listener SHALL accept Minimum $TR0_B$ set to any value specified for it in Table 66. The Listener SHALL NOT generate a subcarrier before $TR0_{B,MIN}$, which is set to the value of Minimum $TR0_B$.
15.6.1.7	The Reader/Writer MAY set Minimum $TR1_B$ to any value specified for it in Table 66. For further communication the Reader/Writer SHALL be capable of establishing a phase reference during $TR1_{B,MIN}$, which is set to Minimum $TR1_B$.	15.6.1.8	The Listener SHALL accept Minimum $TR1_B$ set to any value specified for it in Table 66. The Listener SHALL generate a subcarrier in order to establish a phase reference for $TR1_{B,MIN}$, which is set to Minimum $TR1_B$.
15.6.1.9	The Reader/Writer MAY set b4 and b3 of Param 1 according to its capabilities to receive frames from the Listener with or without SoS/EoS.	15.6.1.10	The Listener SHALL accept b4 and b3 of Param 1 set to any value. During subsequent ISO-DEP block exchange, the Listener MAY implement suppression of SoS/EoS.
		NOTE	Suppression of SoS/EoS only applies to communication at $\mathbf{f}_C/128$ in the Listener to Reader/Writer direction.

Requirements 195: Listen Mode Handling of RFU Values in Received Minimum $TR0_B$ and $TR1_B$

Listen Mode	
15.6.1.11	A received Minimum $TR0_B$ value of RFU SHALL be treated by the Listener as b8-b7=00b.
15.6.1.12	A received Minimum $TR1_B$ value of RFU SHALL be treated by the Listener as b6-b5=00b.

Param 2

The Reader/Writer codes Param 2, as specified in Table 67.

Table 67: ATTRIB Command Param 2 Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x							Bit rate Listen→Poll
		x	x					Bit rate Poll→Listen
				x	x	x	x	FSDI

Requirements 196: ATTRIB Command Param 2 Format

Poll Mode		Listen Mode	
15.6.1.13	The Reader/Writer SHALL format Param 2 as specified in Table 67.	15.6.1.14	The Listener SHALL be ready to receive Param 2 formatted as specified in Table 67.

The least significant nibble (b4 to b1) of Param 2 is used by the Reader/Writer to code the frame size (FSD) that it can receive. The FSD defines the maximum size of a frame the Reader/Writer is able to receive. FSD is expressed in number of data bytes included in the frame.

Requirements 197: Poll Frame Size (FSD)

Poll Mode		Listen Mode	
15.6.1.15	The Reader/Writer SHALL accept frames with a number of data bytes less than or equal to FSD. If the Reader/Writer receives a frame with more than FSD data bytes, it SHALL treat the frame as a Syntax Error .	15.6.1.16	The Listener SHALL send frames with a number of data bytes less than or equal to FSD.
15.6.1.17	The FSD supported by the Reader/Writer SHALL be $\mathbf{FSD}_{T4BT,MIN}$. Appendix B.8 lists the value of $\mathbf{FSD}_{T4BT,MIN}$.	15.6.1.18	The Listener SHALL support an FSD up to $\mathbf{FSD}_{T4BT,MIN}$.

Table 68 specifies the correlations of the FSDI integers to FSD values.

Table 68: FSDI to FSD Conversion

FSDI	FSD (bytes)
0h	16
1h	24
2h	32
3h	40
4h	48
5h	64
6h	96
7h	128
8h	256
9h	512
Ah	1024
Bh	2048
Ch	4096
Dh – Fh	RFU

Requirements 198: FSDI

Poll Mode	
15.6.1.19	The Reader/Writer SHALL set FSDI greater than or equal to the integer value corresponding to $\mathbf{FSD}_{T4BT,MIN}$. Appendix B.8 lists the value of $\mathbf{FSD}_{T4BT,MIN}$. Table 68 specifies the FSDI to FSD conversion.

Requirements 199: Listen Mode Handling of FSDI that has the Value RFU

Listen Mode	
15.6.1.20	A received FSDI value of RFU SHALL be treated by the Listener as FSDI=Ch.

The most significant nibble (b8 through b5) is used by the Reader/Writer for bit rate selection, using the Param 2 coding shown in Table 69 and Table 70. $\mathbf{D}_{LISTEN \rightarrow POLL}$ and $\mathbf{D}_{POLL \rightarrow LISTEN}$ are defined in Section 3.

Table 69: Param 2 b8 and b7 Coding

b8	b7	Meaning
0	0	$D_{LISTEN \rightarrow POLL} = 1$
0	1	$D_{LISTEN \rightarrow POLL} = 2$
1	0	$D_{LISTEN \rightarrow POLL} = 4$
1	1	$D_{LISTEN \rightarrow POLL} = 8$

Table 70: Param 2 b6 and b5 Coding

b6	b5	Meaning
0	0	$D_{POLL \rightarrow LISTEN} = 1$
0	1	$D_{POLL \rightarrow LISTEN} = 2$
1	0	$D_{POLL \rightarrow LISTEN} = 4$
1	1	$D_{POLL \rightarrow LISTEN} = 8$

Requirements 200: Setting the Bit Rate

Poll Mode		Listen Mode	
15.6.1.21	The Reader/Writer SHALL select bit rates compliant with the bit rates proposed by the Listener in its SENSB_RES Response.	15.6.1.22	The Listener SHALL accept the bit rates selected by the Reader/Writer in the ATTRIB Command, provided they comply with those proposed by the Listener in its SENSB_RES Response. The integer coding of the bit rates is defined in Table 69 and Table 70.

Param 3

Param 3 is used for confirmation of the protocol type and is coded as specified in Table 71.

Table 71: ATTRIB Command Param 3 Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0					Fixed Value
				0				RFU
					x	x		Minimum $TR2_B$
							x	Platform Type

Requirements 201: ATTRIB Command Param 3 Format

Poll Mode	
15.6.1.23	The Reader/Writer SHALL set bits b3 through b1 of Param 3 (as defined in Table 71) to the values b3 through b1 of the Protocol_Type parameter in the latest received SENSB_RES (defined in Table 33).
15.6.1.24	The Reader/Writer SHALL set bits b8 through b4 (defined in Table 71) to 00000b.

Requirements 202: ATTRIB Command Param 3 Listen Mode Handling

Listen Mode	
15.6.1.25	The Listener SHALL ignore bits b3 and b2 of Param 3, as defined in Table 71.

Param 4

Table 72 specifies the location of the DID code of the Param 4 byte.

Table 72: ATTRIB Command Param 4 Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x	x	x	x	DID

The least significant nibble b4 through b1 of Param 4 is named DID, and it defines the logical number of the addressed Listener (see Table 57).

Requirements 203: ATTRIB Command Param 4 Format

Poll Mode		Listen Mode	
15.6.1.26	If DID support is indicated by the Listener in its SENSB_RES Response, the Reader/Writer SHALL set the DID field (defined in Table 72) to a value in the range 0 to 14. Otherwise, the Poller SHALL set the DID field to 0.	15.6.1.27	The Listener SHALL be ready to receive a DID field (defined in Table 72) with a value in the range 0 to 14.
15.6.1.28	The Reader/Writer SHOULD assign a DID field (defined in Table 72) the value 0 during communication if it intends to communicate with a single device in Listen Mode.	15.6.1.29	The Listener SHALL treat as a Syntax Error a DID field (as defined in Table 72) that has the RFU value of 15.
		15.6.1.30	The Listener that does not indicate support for DID in the SENSB_RES Response (b1 of byte 3 of Protocol Info set to 0b) SHALL treat an ATTRIB Command whose DID field has a value in the range from 1 to 14 as a Protocol Error.

Higher layer – INF

The Higher layer – INF field can include any higher layer Command transferable as an INF field in the Half-Duplex Block Transmission protocol, as defined in Section 16.

Requirements 204: Higher Layer – INF Field

Poll Mode		Listen Mode	
15.6.1.31	A Reader/Writer MAY include a higher layer Command in the Higher layer – INF field of up to $\mathbf{k}_{T4BT,MAX,ATTRIB}$ bytes. Appendix B.8 lists the value of $\mathbf{k}_{T4BT,MAX,ATTRIB}$.	15.6.1.32	The Listener SHALL be ready to receive an ATTRIB Command with or without a Higher layer – INF field of up to $\mathbf{k}_{T4BT,MAX,ATTRIB}$ bytes. Appendix B.8 lists the value of $\mathbf{k}_{T4BT,MAX,ATTRIB}$.

15.6.2 ATTRIB Response

A Listener answers to an ATTRIB Command that has the format described in Table 73. A Valid Response to an ATTRIB Command is the means for a Reader/Writer to verify that activation of the Listener has been successful.

Table 73: ATTRIB Response Format

Byte 1		Byte 2 – $2+n_{T4BT,MAX,ATTRIB}-1$
MBLI	DID	Higher layer – Response

Requirements 205: ATTRIB Response Format

Poll Mode		Listen Mode	
15.6.2.1	The Reader/Writer SHALL be ready to receive an ATTRIB Response that has the format specified in Table 73.	15.6.2.2	The Listener SHALL send the ATTRIB Response formatted as specified in Table 73.

DID (Device Identification Number)

- The least significant nibble (b4 to b1) of Byte 1 contains the returned DID.

Requirements 206: ATTRIB Response DID

Poll Mode		Listen Mode	
15.6.2.3	The Reader/Writer SHALL be ready to receive a DID field that has the same value as was sent in the DID field of the preceding ATTRIB Command.	15.6.2.4	The Listener SHALL set the DID in the ATTRIB Response to the same value as it received in the DID field of the preceding ATTRIB Command.
15.6.2.5	The Reader/Writer SHALL treat a DID field that has a different value as a Syntax Error.		

NOTE A valid answer (same DID and valid CRC_B) to an ATTRIB Command is the way that a Reader/Writer verifies that the Listener has been selected successfully.

MBLI

- The most significant nibble (b8 through b5) of Byte 1 codes the Maximum Buffer Length Index (MBLI). This index is used by the Listener to inform the Reader/Writer about the Maximum Buffer Length (MBL) it can use to receive chained frames. The MBLI coding is:
  - MBLI = 0 means that the Listener provides no information about its internal input buffer size.
  - MBLI > 0 is used to calculate the actual internal maximum buffer length (MBL), which is calculated using the following formula:

$$MBL = FSC \times 2^{MBLI-1}$$

Requirements 207: MBLI

Poll Mode		Listen Mode	
15.6.2.6	The Reader/Writer SHALL accept an MBLI of any value.	15.6.2.7	The Listener SHALL set the MBLI to 0000b.

Higher layer – Response

- The Higher layer – Response field includes the answer to the higher layer Command included in the Higher layer – INF field of the ATTRIB Command.

Requirements 208: Higher Layer – INF Field Response

Poll Mode		Listen Mode	
15.6.2.8	After the Reader/Writer has transmitted an ATTRIB Command that contains a Higher layer – INF field of up to $\mathbf{n}_{T4BT,MAX,ATTRIB}$ bytes, it SHALL be ready to receive an ATTRIB Response, whether or not that response contains a Higher layer – INF field. Appendix B.8 lists the value of $\mathbf{n}_{T4BT,MAX,ATTRIB}$.	15.6.2.9	The Listener SHALL send an empty Higher layer – Response to an ATTRIB Command that does not contain a Higher layer – INF field. The Listener MAY respond with a Higher layer – Response to an ATTRIB Command that contains a Higher layer – INF field.

NOTE If the Listener responds with an empty Higher layer – Response to an ATTRIB Command with a Higher layer – INF field, it might indicate that the Higher layer Command is not supported.

15.7 Error Handling

When errors are detected the following error handling is attempted.

Requirements 209: Type 4B Tag Platform Error Handling

Poll Mode		Listen Mode	
15.7.1.1	If a Timeout Error occurs after the transmission of the ATTRIB Command, the Reader/Writer SHALL attempt error recovery by resending the ATTRIB Command until a Valid Response is received, or until a maximum of $\mathbf{n}_{T4BT,RETRY,ATTRIB}$ error recovery ATTRIB Commands have been sent. • If for every ATTRIB Command a Timeout Error is detected, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. • If the received Response is not a Valid Response, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception. Appendix B.8 lists the values of $\mathbf{n}_{T4BT,RETRY,ATTRIB}$.	15.7.1.2	The Listener SHALL detect Transmission Errors and Protocol Errors.
15.7.1.3	If a Transmission Error occurs after the transmission of the ATTRIB Command, the Reader/Writer SHALL raise the Unrecoverable Transmission Exception.	15.7.1.4	The Listener SHALL NOT attempt error recovery. The Listener SHALL always stay in receive mode when a Transmission Error or a Protocol Error occurs.

NOTE Transmission Errors on the Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

15.8 Timing Requirements

The Type 4B Tag Platform uses NFC-B frame timing rules, with an exception for the case when an extended SENSB_RES Response has been received. The maximum Activation Frame Waiting Time $\mathbf{FWT}_{T4BT,ACTIVATION}$ for the ATTRIB Response applies as well as the Special Frame Guard Time $\mathbf{SFGT}_{T4BT}$ for the first Command after the ATTRIB Response.

15.8.1 General Timing Requirements

Requirements 210: Type 4B Tag Platform Frame Timing

Poll and Listen Mode	
15.8.1.1	Type 4B Tag Platform Commands and Responses SHALL follow the NFC-B Technology frame timing definitions defined in Section 7.9.

15.8.2 Activation FWT

Requirements 211: Type 4B Tag Platform Activation Frame Timing

Poll Mode		Listen Mode	
15.8.2.1	If the Reader/Writer sends a SENSB_REQ/ ALLB_REQ Command with its bit b5 set to 1b and the Reader/Writer receives an extended SENSB_RES, including the optional 4th Protocol Info Byte, then the Reader/Writer SHALL send an ATTRIB Command and wait at least $\mathbf{FWT}_{T4BT,ACTIVATION} + \Delta\mathbf{FWT}_{B}$ for the ATTRIB Response. If the Reader/Writer does not receive the ATTRIB Response within $\mathbf{FWT}_{T4BT,ACTIVATION} + \Delta\mathbf{FWT}_{B} + \Delta\mathbf{T}_{B,POLL}$, then the Reader/Writer SHALL treat the lack of response as a Timeout Error (i.e., $\mathbf{FDT}_{B,LISTEN,MAX}$ equals $\mathbf{FWT}_{T4BT,ACTIVATION} + \Delta\mathbf{FWT}_{B} + \Delta\mathbf{T}_{B,POLL}$). During the interval between the end of the time period $\mathbf{FWT}_{T4BT,ACTIVATION} + \Delta\mathbf{FWT}_{B}$ and the end of the time period $\mathbf{FWT}_{B,ACTIVATION} + \Delta\mathbf{FWT}_{B} + \Delta\mathbf{T}_{B,POLL}$, the Reader/Writer MAY accept the ATTRIB Response or MAY generate a Timeout Error. Appendix B.3 lists the values of $\Delta\mathbf{T}_{B,POLL}$ and $\Delta\mathbf{FWT}_{B}$. Appendix B.8 lists the values of $\mathbf{FWT}_{T4BT,ACTIVATION}$.	15.8.2.2	If the Listener returns an extended SENSB_RES Response, including the optional 4th Protocol Info Byte, then the Listener SHALL start sending the ATTRIB Response within $\mathbf{FWT}_{T4BT,ACTIVATION}$ after the end of the ATTRIB Command.

15.8.3 SFGT

A specific value of $\mathbf{FDT}_{B,POLL,MIN}$ is defined after the ATTRIB Response. The $\mathbf{SFGT}_{T4BT}$ is the minimum time the Reader/Writer waits before the Listener is ready to receive the next frame after it has sent the ATTRIB Response.

The $\mathbf{SFGT}_{T4BT}$ is calculated by the following formula:

$$\mathbf{SFGT}_{T4BT} = (256 \times 16/\mathbf{f}_C) \times 2^{SFGI},$$

where the value of SFGI has a range from 0 to 14 and is returned by the Listener in the SENSB_RES Response (see 7.6.2). If the Listener returns SFGI equal to 0 or SFGI is not returned, then no **SFGT** is needed and $\mathbf{FDT}_{B,POLL,MIN}$ applies.

The requirements are defined with a margin $\Delta\mathbf{SFGT}_{T4BT}$ that includes enough time for the Reader/Writer to receive a frame within the defined waiting time plus the margin. The $\Delta\mathbf{SFGT}_{T4BT}$ is calculated using the following formula:

$$\Delta\mathbf{SFGT}_{T4BT} = 384 \times 2^{\mathbf{SFGI}}.$$

Requirements 212: NFC-B SFGT

Poll Mode		Listen Mode	
15.8.3.1	If the Reader/Writer has set b5 of its SENSB_REQ/ ALLB_REQ Command to 1b, and it receives an SFGI that is not equal to 0, then, after the Listener has sent the ATTRIB Response, the Reader/Writer SHALL wait at least $\mathbf{SFGT}_{T4BT} + \Delta\mathbf{SFGT}_{T4BT}$ before it sends the next frame. **SFGT** + $\Delta\mathbf{SFGT}_{T4BT}$ is computed from SFGI as defined above. Appendix B.8 lists the value of $\mathbf{SFGI}_{T4BT,MAX}$.	15.8.3.2	If the Listener returns an SFGI other than 0, then it SHALL be ready to receive the start of a new Poll Frame no later than $\mathbf{SFGT}_{T4BT}$ after the end of the ATTRIB Response frame. If the start of a new Poll Frame is received before $\mathbf{SFGT}_{T4BT}$, then the Listener MAY treat this as a Transmission Error. **SFGT** is computed from SFGI as defined above. Appendix B.8 lists the value of $\mathbf{SFGI}_{T4BT,MAX}$.
15.8.3.3	If the Reader/Writer receives an SFGI equal to 0 or SFGI is not returned, or if the Reader/Writer set bit b5 of its SENSB_REQ/ALL_REQ Command to 0b, then, after the Listener has sent the ATTRIB Response, the Reader/Writer SHALL wait at least $\mathbf{FDT}_{B,POLL,MIN}$ before it sends the next frame.	15.8.3.4	If the Listener returns an SFGI equal to 0 or does not return an SFGI, then it SHALL be ready to receive the start of a new Poll Frame no later than $\mathbf{FDT}_{B,POLL,MIN}$ after the end of the ATTRIB Response frame. If the start of a new Poll Frame is received before $\mathbf{FDT}_{B,POLL,MIN}$, then the Listener MAY treat this receipt as a Transmission Error.

15.8.4 Frame Delay Time Poll→Listen

The Frame Delay Time Poll→Listen for Type 4B Tag Platform is the time between a Poll Frame and a Listen Frame, as illustrated in Figure 14.

For a device that implements the Type 4B Tag Platform, the value of $\mathbf{FDT}_{T4BT,LISTEN,MIN}$ is the minimum waiting time (after the end of a Poll Frame) until the device will start to send the Listen Frame. $\mathbf{FDT}_{T4BT,LISTEN,MIN}$ is defined:

$$\mathbf{FDT}_{T4BT,LISTEN,MIN} = \mathbf{TR0}_{B,MIN} + \mathbf{TR1}_{B,MIN} .$$

$\mathbf{TR0}_{B,MIN}$ and $\mathbf{TR1}_{B,MIN}$ are indicated in Param 1 of the ATTRIB Command. Section 15.6.1 defines the ATTRIB Command.

Requirements 213: Type 4B Tag Platform $\mathbf{FDT}_{T4BT,LISTEN,MIN}$

Poll Mode		Listen Mode	
15.8.4.1	The Reader/Writer SHALL be ready to receive the SoS of a Listen Frame no later than $\mathbf{FDT}_{T4BT,LISTEN,MIN}$ after the EoS of a Poll Frame.	15.8.4.2	Following the EoS of a Poll Frame, the Listener SHALL wait at least $\mathbf{FDT}_{T4BT,LISTEN,MIN}$ before it sends the SoS of its Listen Frame.

16 ISO-DEP Protocol

When the NFC Forum Device has been configured for either the Type 4A or the Type 4B Tag Platform and has been activated, then during the Data Exchange Activity and until the device's deactivation, all Commands and Responses (i.e., the payload) are transmitted according to the Half-Duplex Block Transmission Protocol that is specified in this section.

This section is derived from [ISO/IEC_14443].

16.1 Block Format

Data bytes that are transmitted as part of the ISO-DEP Protocol are organized as blocks. ISO-DEP blocks are equivalent to the data and payload format (see Section 6.4 and 7.4) used by other protocols, and are sent in NFC-A Standard Frames (see Section 6.3.3) or NFC-B frames (see Section 7.3). A block complies with the data protocol layer used in this document.

16.1.1 Block

Blocks consist of three fields, the SoD, the payload and the EoD. The SoD contains the Protocol Control Byte (PCB), the optional DID and the optional NAD.

The EoD contains a two-byte checksum, referred to as the cyclic redundancy check (CRC). The SoD and the payload are the inputs for the CRC calculation.

NOTE For the Type 4A Tag Platform the CRC is referred to as "CRC_A"; for the Type 4B Tag Platform, the CRC is referred to as "CRC_B".

Figure 38 shows the block format.

Block (Data)					
SoD			**Payload**	**EoD**	
PCB	[DID]	[NAD]	[INF]	CRC_1	CRC_2

Figure 38: ISO-DEP Protocol Block Format

NOTE In this section the terms "Prologue Field", "Information Field", "Epilogue Field" and "EDC" used in [ISO/IEC_14443] are referred to as "SoD", "payload", "EoD" and "CRC", respectively.

Requirements 214: ISO-DEP Protocol Block

Poll and Listen Mode	
16.1.1.1	ISO-DEP blocks SHALL be coded and interpreted as defined in Figure 38.
16.1.1.2	ISO-DEP blocks SHALL be transmitted and received in frames as defined in Sections 14.3 and 15.3.

16.1.2 SoD

The SoD contains the mandatory PCB and, optionally, the DID and the NAD.

- PCB

  The Protocol Control Byte (PCB) is used to convey the information needed to control the data transmission. The transmission 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 Reader/Writer and the Listener. Two different types of S-blocks are defined:
    - Waiting Time eXtension (WTX) that contains a 1-byte INF field
    - DESELECT that contains no INF field.

  The format of the PCB depends on its type. The formats of I-blocks, R-blocks and S-blocks are shown in Table 74, Table 75, Table 76 and Table 77.

Table 74: Block Type Format

b8	b7	Meaning
0	0	I-block
0	1	Not Allowed
1	0	R-block
1	1	S-block

Table 75: I-block PCB Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							I-block indicator
		0						Fixed Value
			x					Chaining, if bit is set to 1b
				x				DID following, if bit is set to 1b
					x			NAD following, if bit is set to 1b
						1		Fixed Value
							x	Block number

Table 76: R-block PCB Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	0							R-block indicator
		1						Fixed Value
			x					ACK, if bit is set to 0b NAK, if bit is set to 1b
				x				DID following, if bit is set to 1b
					0			Fixed Value
						1		Fixed Value
							x	Block number

Table 77: S-block PCB Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	1							S-block indicator
		x	x					DESELECT, if set to 00b WTX, if set to 11b All other values are Not Allowed
				x				DID following, if bit is set to 1b
					0			Fixed Value
						1		Fixed Value
							0	Fixed Value

Requirements 215: ISO-DEP Protocol PCB

Poll and Listen Mode	
16.1.2.1	The I-block PCB SHALL be formatted and interpreted as defined in Table 75.
16.1.2.2	The R-block PCB SHALL be formatted and interpreted as defined in Table 76.
16.1.2.3	The S-block PCB SHALL be formatted and interpreted as defined in Table 77.
16.1.2.4	When it is sending, the NFC Forum Device SHALL set b4 of the PCB to 1b if the SoD contains a DID field.
16.1.2.5	When it is receiving, the NFC Forum Device SHALL be ready to receive a DID field if b4 of the PCB is equal to 1b.
16.1.2.6	When it is sending an I-block, the NFC Forum Device SHALL set b3 of the PCB to 1b if the SoD contains a NAD field.
16.1.2.7	When it is receiving an I-block, the NFC Forum Device SHALL be ready to receive a NAD field if b3 of the PCB is equal to 1b.

- DID Field

  The DID is used to identify a specific Listener. The two most significant bits – b8 and b7 – code the power level indicator. The format of the DID field is shown in Table 78. See also Section 14.6.1 and Section 15.6.1.

Table 78: Format of DID Field

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x							Power Level Indicator
		0	0					Fixed Value
				x	x	x	x	DID

The least significant nibble b4 through b1 of the DID field is named DID, and it defines the logical number of the addressed Listener (see Table 57).

Requirements 216: ISO-DEP Protocol DID Field

Poll and Listen Mode	
16.1.2.8	The DID field SHALL be formatted and interpreted as defined in Table 78.

Requirements 217: DID

Poll Mode		Listen Mode	
16.1.2.9	The Reader/Writer SHALL NOT include a DID field, if the Listener does not support DID.	16.1.2.10	A Listener that does not support DID SHALL ignore blocks including a DID field.
16.1.2.11	If the Listener supports DID, the Reader/Writer: • SHOULD omit the DID field, if the DID negotiated during the Device Activation Activity is 0 • SHALL include the DID field with this value in ISO-DEP blocks, if the DID negotiated during the Device Activation Activity is other than 0.	16.1.2.12	A Listener that supports DID: • SHALL respond to blocks that contain the DID negotiated during the Device Activation Activity by using this same DID value • SHALL ignore a block whose DID value is different from the DID value negotiated during the Device Activation Activity • SHALL, in case the DID value negotiated during the Device Activation Activity is other than 0, ignore a block that contains no DID field • SHALL, in case the DID value negotiated during the Device Activation Activity is 0, respond to a block that contains no DID field by using no DID field.

NOTE When a DID equal to 0 has been negotiated, devices in Poll Mode typically omit the DID field to reduce communication overhead rather than include a DID field whose DID is 0.

Requirements 218: Power Level Indicator

Poll Mode		Listen Mode	
16.1.2.13	The Reader/Writer SHALL set the power level indicator to 00b.	16.1.2.14	The Listener SHALL accept a power level indicator equal to 00b. The Listener SHALL treat a power level indicator different from 00b as a Syntax Error.
16.1.2.15	The Reader/Writer SHALL accept a power level indicator of any value.	16.1.2.16	The Listener SHALL set the power level indicator to 00b.

- NAD Field

  The NAD is used to identify a specific logical connection over the ISO-DEP Protocol link.

Requirements 219: NAD

Poll Mode		Listen Mode	
16.1.2.17	The Reader/Writer SHALL NOT include a NAD, regardless of whether the Listener indicated support for NAD.	16.1.2.18	A Listener that does not support NAD SHALL ignore blocks that include a NAD.

16.1.3 Payload

The payload consists of the optional INF field. 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 payload is calculated by counting the number of bytes of the whole block minus the length of the SoD and the EoD.

16.1.4 EoD

The EoD contains the CRC.

Requirements 220: CRC

Poll and Listen Mode	
16.1.4.1	A block SHALL contain an EoD at the position indicated in Figure 38. For the Type 4A Tag Platform the EoD SHALL contain a CRC_A as defined in Section 6.4. For the Type 4B Tag Platform the EoD SHALL contain a CRC_B as defined in Section 7.4.

16.2 Protocol Operation

16.2.1 General Rules

General rules for the Type 4A Tag and Type 4B Tag transmission protocol are described in Section 10.

16.2.2 Frame Waiting Time Extension

When the Listener needs more time than the defined FWT to process the received block, it uses an S(WTX) Request for a waiting time extension. An S(WTX) Request contains a 1-byte INF field, as specified in Table 79.

After it has sent an S(WTX) Request, the Listener waits for a response. If it receives an S(WTX) Response, it continues operation from its state before it sent the S(WTX) Request.

Table 79: S(WTX) Request INF Field Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x							Power level indication
		x	x	x	x	x	x	WTXM

Requirements 221: S(WTX) Request

Poll Mode		Listen Mode	
16.2.2.1	The Reader/Writer SHALL be ready to receive the INF field of an S(WTX) Request that has the format specified in Table 79.	16.2.2.2	The Listener SHALL format the INF field of an S(WTX) Request according to Table 79.

- Power Level Indicator

  The two most significant bits (b8 and b7) code the power level indication.

Requirements 222: Power Level Indication

Poll Mode		Listen Mode	
16.2.2.3	The Reader/Writer SHALL accept a power level indication of any value.	16.2.2.4	The Listener SHALL set the power level indication to 00b.

- WTXM

  The bits b6 through b1 code WTXM. The WTXM is coded in the range from 1 to 59.

  The Reader/Writer acknowledges the request by sending an S(WTX) Response that also contains a 1-byte INF field. The INF field consists of two parts that contain the same WTXM as received in the request (see Table 80).

Table 80: S(WTX) Response INF Field Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		x	x	x	x	x	x	WTXM

Requirements 223: S(WTX) Response

Poll Mode		Listen Mode	
16.2.2.5	The Reader/Writer SHALL format the INF field of an S(WTX) Response according to Table 80.	16.2.2.6	The Listener SHALL be ready to receive the INF field of an S(WTX) Response that has the format specified in Table 80.

Requirements 224: Handling of S(WTX) Response INF Field with Value RFU

Listen Mode	
16.2.2.7	A received value for b8 or b7 that is different from 0b SHALL be treated as a Syntax Error.

The temporary FWT value corresponding to the WTXM in the INF field is calculated using the following formula:

$$\mathbf{FWT}_{TEMP} = FWT \times WTXM$$

The time $\mathbf{FWT}_{TEMP}$ requested by the Listener starts after the Reader/Writer has sent the S(WTX) Response.

Requirements 225: Frame Waiting Time Extension

Poll Mode		Listen Mode	
16.2.2.8	The Reader/Writer SHALL be ready to receive an S(WTX) having a WTXM with a value in the range from 1 to 59. If it receives an S(WTX) with WTXM set to 0 or to a value in the range 60 to 63, the Reader/Writer SHALL treat the frame as a Syntax Error .	16.2.2.9	The Listener SHALL set the value of WTXM in the range from 1 to 59.
16.2.2.10	The Reader/Writer SHALL support a frame waiting time extension less than or equal to the maximum supported FWT**:** • $\mathbf{FWT}_{TEMP} \leq (256 \times 16/\mathbf{f}_C) \times 2^{\mathbf{FWI}_{T4AT,MAX}}$ for the Type 4A Tag Platform. • $\mathbf{FWT}_{TEMP} \leq (256 \times 16/\mathbf{f}_C) \times 2^{\mathbf{FWI}_{B,MAX}}$ for the Type 4B Tag Platform.	16.2.2.11	The Listener SHALL set the value of WTXM so that $\mathbf{FWT}_{TEMP}$ is less than or equal to the maximum value of FWT: • $\mathbf{FWT}_{TEMP} \leq (256 \times 16/\mathbf{f}_C) \times 2^{\mathbf{FWI}_{T4AT,MAX}}$ for the Type 4A Tag Platform. • $\mathbf{FWT}_{TEMP} \leq (256 \times 16/\mathbf{f}_C) \times 2^{\mathbf{FWI}_{B,MAX}}$ for the Type 4B Tag Platform.
16.2.2.12	The Reader/Writer SHALL set the value of WTXM in the S(WTX) Response to the same value as the value of WTXM in the S(WTX) request.	16.2.2.13	The Listener SHALL treat as a Syntax Error a WTXM in the S(WTX) Response that holds a value different from the value of WTXM in the S(WTX) request.

16.2.3 Chaining

The chaining feature supports the transmission of information that does not fit into a single block (as defined by the FSC or FSD), by dividing the information into several blocks.

Requirements 226: Chaining Rules

Poll and Listen Mode	
16.2.3.1	The chaining bit in the PCB of an I-block controls the chaining of blocks. When an I-block indicating chaining is received, the block SHALL be acknowledged by an R(ACK) block.

NOTE The requirement 16.2.3.1 applies even if the INF field of the received I-block has zero length.

Requirements 227: Block Sizes during Chaining

Poll Mode		Listen Mode	
16.2.3.2	When the Reader/Writer sends a chain of I-blocks, each block that indicates chaining (i.e., all blocks of the chain except the last one) SHALL have a: • Minimum length of 256 Bytes, if FSC is greater than 256 Bytes • Length of FSC, if FSC is equal to or less than 256 Bytes.	16.2.3.3	When the Listener sends a chain of I blocks, each block indicating chaining (i.e. all blocks of the chain except the last one) SHALL have an INF field with a length that is not equal to zero.

Requirements 228: Last Block

Poll Mode		Listen Mode	
16.2.3.4	The Reader/Writer SHOULD NOT send an empty I-block (i.e., without an INF field) not indicating chaining after an I-block indicating chaining (i.e., preferably, the last block in a chain of blocks is not empty).	16.2.3.5	The Listener SHALL accept an empty I-block that does not indicate chaining after receipt of an I-block indicating chaining.

16.2.4 Block Numbering Rules

This section specifies the rules for block numbering in a block of the Reader/Writer and Listener.

Requirements 229: Block Numbering Rules

Poll Mode		Listen Mode	
16.2.4.1	The block number of the Reader/Writer SHALL be initialized to 0 for the current activated Listener.	16.2.4.2	The block number of the Listener SHALL be initialized to 1 at activation.
16.2.4.3	When a valid I-block or a valid R(ACK) block whose block number is equal to the current block number is received, the Reader/Writer SHALL toggle the current block number for the current Listener before optionally sending a block.	16.2.4.4	When a received valid I-block causes the Listener to send a block as an answer, the Listener SHALL toggle its block number before it sends this block.
		16.2.4.5	When an R(ACK) block with a block number that is not equal to the current Listener's block number is received, the Listener SHALL toggle its block number before it sends a block.

16.2.5 Block Handling Rules

This section specifies the block handling rules for the Reader/Writer and the Listener.

Requirements 230: General Block Handling Rules

Poll and Listen Mode	
16.2.5.1	The first block SHALL be sent by the Reader/Writer.
16.2.5.2	An S-block Request SHALL always be followed by an S-block Response, except for the case in which the Reader/Writer does not want to accept any additional S(WTX) Requests. The Reader/Writer SHALL respond to the first S(WTX) Request that it receives after it sends an I-block or R-block to the Listener. The Reader/Writer MAY stop accepting subsequent S(WTX) Requests, preventing the Reader/Writer to be blocked by arbitrarily long Listener processing times. Nevertheless the Reader/Writer SHOULD accept a number of S(WTX) Requests to support the completion of the Listener's normal operation.

Requirements 231: Reader/Writer Block Handling Rules

Poll Mode	
16.2.5.3	When an R(ACK) block is received in response to an R(NAK) block sent by the Reader/Writer to notify a timeout, and that block is received with a block number that is not equal to the Reader/Writer's current block number, then the Reader/Writer SHALL re-transmit the last I-block. In all other cases, when an R(ACK) block with a block number that is not equal to the Reader/Writer's current block number is received, then the Reader/Writer MAY raise the Unrecoverable Protocol Exception or the Reader/Writer MAY retransmit the last I-block.
16.2.5.4	When the Reader/Writer has retransmitted an I-block two times (i.e., the same I-block was sent three times), it SHALL be treated as a Semantic Error if an R(ACK) block with a block number not equal to the Reader/Writer's current block number is received.
16.2.5.5	When an R(ACK) block is received, chaining SHALL be continued if its block number is equal to the Reader/Writer's current block number and the last I-block sent by the Reader/Writer indicates chaining. If the last I-block sent by the Reader/Writer did not indicate chaining, then the Reader/Writer SHALL treat the R(ACK) block as a Semantic Error.
16.2.5.6	When an R(NAK) block is received by the Reader/Writer, it SHALL be treated as a Semantic Error.

Requirements 232: Listener Block Handling Rules

Listen Mode	
16.2.5.7	The Listener MAY send an S(WTX) block instead of an I-block or an R(ACK) block, except in the case of a retransmitted I-block or a retransmitted R(ACK) block from the Listener.
16.2.5.8	When an I-block that does not indicating chaining is received, the block SHALL be acknowledged by an I-block. If the received I-block that does not indicating chaining is empty (i.e., without an INF field), then the mandatory I-block sent MAY either be empty or contain any applicable information (e.g., error code).
16.2.5.9	When an R(ACK) or an R(NAK) block is received with a block number equal to the Listener's current block number, then: • If the last block was sent by the Listener (i.e., the last block from the Listener has not been acknowledged by the Reader/Writer), then the last block SHALL be retransmitted. • If the last block was sent by the Reader/Writer (i.e., the last block from the Listener has been acknowledged by the Reader/Writer), then the next block SHALL be sent.
16.2.5.10	When an R(NAK) block is received, an R(ACK) block SHALL be sent if its block number is not equal to the Listener's current block number.
16.2.5.11	When an R(ACK) block is received, chaining SHALL be continued if its block number is not equal to the Listener's current block number and the last I-block sent by the Listener indicates chaining.

NOTE If the last I-block sent by the Listener did not indicate chaining, then the Listener might treat the R(ACK) block as a Semantic Error.

NOTE The situation described in requirement 16.2.5.9 can occur if, for example, the Reader/Writer has detected a disturbance in the field that it interprets as an invalid response, but which was not caused by the Listener.

16.2.6 Error Handling

When errors are detected the following error handling is attempted.

Requirements 233: Listener Error Handling

Listen Mode	
16.2.6.1	The Listener SHALL detect Transmission Errors and Protocol Errors.
16.2.6.2	The Listener SHALL NOT attempt error recovery. The Listener SHALL always stay in receive mode when a Transmission Error or a Protocol Error occurs.

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

Requirements 234: Reader/Writer Error Handling

Poll Mode	
16.2.6.3	If a block that contains a Transmission Error is received after receipt of a block that does not indicate chaining (except in the case of S(DESELECT)), then the Reader/Writer SHALL send an R(NAK) block. The Reader/Writer SHALL start to send the R(NAK) block within the time $\mathbf{t}_{ISODEP,RETRANSMISSION}$, measured from the end of the frame that has the Transmission Error. If a Transmission Error is detected after the Reader/Writer sends an R(NAK) block, the Reader/Writer SHALL attempt error recovery by resending the R(NAK) block until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,NAK}$ error recovery R(NAK) blocks have been sent. If no Valid Response is received for all of these R(NAK) blocks, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception. Appendix B.9 lists the values of $\mathbf{n}_{ISODEP,RETRY,NAK}$ and $\mathbf{t}_{ISODEP,RETRANSMISSION}$.
16.2.6.4	If a block that contains a Protocol Error is received after receipt of a block that does not indicate chaining (except in the case of S(DESELECT)), then the Reader/Writer SHALL raise the Unrecoverable Protocol Exception.
16.2.6.5	If a Timeout Error occurs after receipt of a block that does not indicate chaining (except in the case of S(DESELECT), then the Reader/Writer SHALL send an R(NAK) block. The Reader/Writer SHALL send the R(NAK) block after FWT + $\Delta\mathbf{FWT}$ and before FWT + $\Delta\mathbf{FWT}$ + $\mathbf{t}_{ISODEP,RETRANSMISSION}$. If a Timeout Error is detected after the Reader/Writer sends an R(NAK) block, the Reader/Writer SHALL attempt error recovery by resending the R(NAK) block either until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,NAK}$ error recovery R(NAK) blocks have been sent. If no Valid Response is received for all of these R(NAK) blocks, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. If the original Block that did not indicate chaining was an S(WTX) Request, the Reader/Writer SHALL maintain a count of the number of times that the above error recovery sequence resulted in a retransmission of that same S(WTX) Request. The Reader/Writer SHALL NOT resend an S(WTX) Response to the same S(WTX) Request more than $\mathbf{n}_{ISODEP,RETRY,WTX}$ times. If a Timeout Error occurs after the S(WTX) Response has been resent $\mathbf{n}_{ISODEP,RETRY,WTX}$ times, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. The term $\Delta\mathbf{FWT}$ stands for $\Delta\mathbf{FWT}_{T4AT}$ for NFC-A or $\Delta\mathbf{FWT}_{B}$ for NFC-B. Appendix B.3 lists the value of $\Delta\mathbf{FWT}_{B}$ and Appendix B.9 lists the value of $\Delta\mathbf{FWT}_{T4AT}$. Appendix B.9 lists the values of $\mathbf{n}_{ISODEP,RETRY,NAK}$ and $\mathbf{n}_{ISODEP,RETRY,WTX}$.

Poll Mode	
16.2.6.6	If a block that contains a Transmission Error is received after receipt of a block that indicates chaining, then the last R(ACK) block sent by the Reader/Writer SHALL be retransmitted. The Reader/Writer SHALL start to send the R(ACK) block within the time $\mathbf{t}_{ISODEP,RETRANSMISSION}$, measured from the end of the frame that contains the Transmission Error. If a Transmission Error is detected after the Reader/Writer sends an R(ACK) block, the Reader/Writer SHALL attempt error recovery by resending the R(ACK) block until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,ACK}$ error recovery R(ACK) blocks have been sent. If no Valid Response is received for all of these R(ACK) blocks, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception. Appendix B.9 lists the value of $\mathbf{n}_{ISODEP,RETRY,ACK}$.
16.2.6.7	If a block that contains a Protocol Error is received after receipt of a block that indicates chaining, then the Reader/Writer SHALL raise the Unrecoverable Protocol Exception.
16.2.6.8	If a Timeout Error occurs after receipt of a block that indicates chaining, the Reader/Writer SHALL attempt error recovery by resending the R(ACK) block until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,ACK}$ error recovery R(ACK) blocks have been sent. The Reader/Writer SHALL send the R(ACK) block after FWT + $\Delta\mathbf{FWT}$ and before FWT + $\Delta\mathbf{FWT}$ + $\mathbf{t}_{ISODEP,RETRANSMISSION}$. If no Valid Response is received for all of these R(ACK) blocks, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. The term $\Delta\mathbf{FWT}$ equals $\Delta\mathbf{FWT}_{T4AT}$ for NFC-A or $\Delta\mathbf{FWT}_{B}$ for NFC-B. Appendix B.3 lists the value of $\Delta\mathbf{FWT}_{B}$ and Appendix B.9 lists the value of $\Delta\mathbf{FWT}_{T4AT}$. Appendix B.9 lists the value of $\mathbf{n}_{ISODEP,RETRY,ACK}$.

NOTE Transmission Errors on the Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

16.2.7 De-activation Rules

This section lists the requirements related to the deactivation of the Listener. The Reader/Writer deactivates the Listener by sending an S(DESELECT) Request. The Listener acknowledges the deactivation by returning an S(DESELECT) Response.

Requirements 235: Deactivation Rules

Poll Mode		Listen Mode	
16.2.7.1	The Reader/Writer SHALL deactivate the Listener by sending an S(DESELECT) Request. The S(DESELECT) Request SHALL be coded as an S-block, as specified in Table 77.	16.2.7.2	The Listener SHALL acknowledge the deactivation request by sending an S(DESELECT) Response. The S(DESELECT) Response SHALL be coded as an S-block, as specified in Table 77.
16.2.7.3	Following the sending of the S(DESELECT) Request to the Listener, the Reader/Writer SHALL wait at least $\mathbf{FWT}_{ISODEP,DEACTIVATION}$ for the S(DESELECT) Response from the Listener.	16.2.7.4	The Listener SHALL send an S(DESELECT) Response to the Reader/Writer within $\mathbf{FWT}_{ISODEP,DEACTIVATION}$ (Appendix B.9 lists the value of $\mathbf{FWT}_{ISODEP,DEACTIVATION}$.

Requirements 236: Deactivation Error Handling

Poll Mode	
16.2.7.5	If a Protocol Error is detected in the S(DESELECT) Response, then the Reader/Writer SHALL raise the Unrecoverable Protocol Exception.
16.2.7.6	If a Transmission Error occurs in the S(DESELECT) Response, the Reader/Writer SHALL attempt error recovery by resending the S(DESELECT) Request until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,DESELECT}$ error recovery S(DESELECT) Requests have been sent. The Reader/Writer SHALL resend the S(DESELECT) Request within a delay of $\mathbf{t}_{ISODEP,RETRANSMISSION}$, measured from the end of the frame that contains the S(DESELECT) Response with the Transmission Error. If no valid S(DESELECT) Response is received for all of these S(DESELECT) Requests, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception. Appendix B.9 lists the value of $\mathbf{n}_{ISODEP,RETRY,DESELECT}$.
16.2.7.7	If no S(DESELECT) Response is received within the delay $\mathbf{FWT}_{ISODEP,DEACTIVATION}$, the Reader/Writer SHALL attempt error recovery by resending the S(DESELECT) Request until a Valid Response is received, or until a maximum of $\mathbf{n}_{ISODEP,RETRY,DESELECT}$ error recovery S(DESELECT) Requests have been sent. The Reader/Writer SHALL resend the S(DESELECT) Request after $\mathbf{FWT}_{ISODEP,DEACTIVATION}$ and before $\mathbf{FWT}_{ISODEP,DEACTIVATION} + \mathbf{t}_{ISODEP,RETRANSMISSION}$. If no valid S(DESELECT) Response is received for all of these S(DESELECT) Requests, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception.

NOTE Transmission Errors on Poll Mode side might be suppressed by the defined EMD handling (see Section 4).

16.3 Timing Requirements

16.3.1 Frame Waiting Time

The Type 4A and Type 4B Tag half-duplex Protocol uses either Type 4A Tag Platform Timing or Type 4B Tag Platform Timing, depending on whether NFC-A or NFC-B technology is used.

Requirements 237: Type 4A and Type 4B Tag Half-duplex Protocol Timing

Poll and Listen Mode	
16.3.1.1	If NFC-A Technology is used, the Commands and Responses specified in Section 16 SHALL follow the Timing Requirements of the Type 4A Tag Platform defined in Section 14.8.
16.3.1.2	If NFC-B Technology is used, the Commands and Responses specified in Section 16 SHALL follow the Timing Requirements of the Type 4B Tag Platform defined in Section 15.

16.3.2 Frame Waiting Time Extension

When the Listener sends an S(WTX) Request with a specific WTXM value, the Reader/Writer waits for $\mathbf{FWT}_{TEMP}$ for the next block. Section 16.2.2 defines $\mathbf{FWT}_{TEMP}$.

Requirements 238: Frame Waiting Time Extension

Poll Mode		Listen Mode	
16.3.2.1	After it sends the S(WTX) Response in response to an S(WTX) Request from the Listener, the Reader/Writer SHALL wait at least for $\mathbf{FWT}_{TEMP} + \mathbf{\Delta FWT}$ for a block of the Listener (that is, $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FDT}_{TEMP} + \mathbf{\Delta FWT}$ in case of NFC-A or $\mathbf{FDT}_{B,LISTEN,MAX}$ equals $\mathbf{FDT}_{TEMP} + \mathbf{\Delta FWT}$ in case of NFC-B). During the interval between the end of the time period $\mathbf{FWT}_{TEMP} + \mathbf{\Delta FWT}$ and the end of the time period $\mathbf{FWT}_{TEMP} + \mathbf{\Delta FWT} + \mathbf{\Delta T}_{POLL}$, the Reader/Writer MAY accept the Response or MAY generate a Timeout Error. If the Reader/Writer does not receive a block from the Listener before the end of the time period $\mathbf{FWT}_{TEMP} + \mathbf{\Delta FWT} + \mathbf{\Delta T}_{POLL}$, then the Reader/Writer SHALL treat the lack of response as a Timeout Error. $\mathbf{\Delta T}_{POLL}$ equals $\mathbf{\Delta T}_{T4AT,POLL}$ for NFC-A or $\mathbf{\Delta T}_{B,POLL}$ for NFC-B. $\mathbf{\Delta FWT}$ equals $\mathbf{\Delta FWT}_{T4AT}$ for NFC-A or $\mathbf{\Delta FWT}_{B}$ for NFC-B.	16.3.2.2	After it receives the S(WTX) Response of the Reader/Writer, the Listener SHALL start to send the next block within the time $\mathbf{FWT}_{TEMP}$ (.That is, the term $\mathbf{FDT}_{A,LISTEN,MAX}$ equals $\mathbf{FDT}_{TEMP}$ in case of NFC-A, or the term $\mathbf{FDT}_{B,LISTEN,MAX}$ equals $\mathbf{FDT}_{TEMP}$ in case of NFC-B).
16.3.2.3	The Reader/Writer SHALL apply $\mathbf{FWT}_{TEMP} + \mathbf{\Delta FWT}$ only until the next block has been received from the Listener.	16.3.2.4	The Listener SHALL apply $\mathbf{FWT}_{TEMP}$ only until the next block has been sent by the Listener.

17 NFC-DEP Protocol

This section specifies the NFC-DEP Protocol used by Peers.

This section is derived from [ISO/IEC_18092].

17.1 Sequence Format

Peers use NFC-A sequence format or NFC-F sequence format for the NFC-DEP Protocol, depending on the configuration.

Requirements 239: NFC-DEP Protocol Sequence Format

Initiator and Target	
17.1.1.1	Commands and Responses specified in Section 17 (NFC-DEP Protocol) SHALL be sent and received using either: • NFC-A Technology sequence format (defined in Section 6.1), if the Peer is using NFC-A Technology • NFC-F Technology sequence format (defined in Section 8.1), if the Peer is using NFC-F Technology.

17.2 Bit Level Coding

Peers use NFC-A bit level coding or NFC-F bit level coding for the NFC-DEP Protocol, depending on the configuration.

Requirements 240: NFC-DEP Protocol Bit Level Coding

Initiator and Target	
17.2.1.1	Commands and Responses specified in Section 17 (NFC-DEP Protocol) SHALL be sent and received using either: • NFC-A Technology bit level coding (defined in Section 6.2), if the Peer is using NFC-A Technology • NFC-F Technology bit level coding (defined in Section 8.2), if the Peer is using NFC-F Technology.

17.3 Frame Format

Peers use NFC-A frame format or NFC-F frame format for the NFC-DEP Protocol, depending on the configuration.

Requirements 241: NFC-DEP Protocol Frame Format

Initiator and Target	
17.3.1.1	Commands and Responses specified in Section 17 (NFC-DEP Protocol) SHALL be sent and received in either: • An NFC-A Technology Standard Frame (defined in Section 6.3), if the Peer is using NFC-A Technology • An NFC-F Technology frame (defined in Section 8.3), if the NFC Forum Device is using NFC-F Technology.

17.4 Data and Payload Format

Data transmitted in a frame has the following substructure and consists of an SoD, the payload, and an EoD.

The SoD contains a start byte SB (NFC-A only), followed by a length byte LEN, indicating the number of payload bytes.

The payload consists of the Commands and Responses described in Section 17.5.

The EoD contains a two-byte checksum, referred to as CRC. The input for the CRC calculation is the SoD and the payload.

Figure 39 shows the NFC-DEP Protocol data and payload format.

Data							
SoD		**Payload**				**EoD**	
SB (NFC-A only)	LEN	Byte 1	Byte 2	…	Byte n	CRC_1	CRC_2

Figure 39: NFC-DEP Protocol Data and Payload Format

Requirements 242: NFC-DEP Protocol Data and Payload Format

Initiator and Target	
17.4.1.1	Data SHALL be transmitted and received in frames, as defined in Section 17.3.
17.4.1.2	If configured for NFC-A Technology, the first byte in the SoD SHALL be a start byte (SB) set to the value F0h. If configured for NFC-A Technology, the Peer SHALL treat an incorrect SB value as a Transmission Error.
17.4.1.3	The SoD SHALL contain a length byte LEN at the position shown in Figure 39 with a value equal to n+1, where n indicates the number of bytes in the payload. The Peer SHALL treat an incorrect LEN value as a Transmission Error.
17.4.1.4	The length byte LEN SHALL have a value in the range 3 to 255. The Peer SHALL treat any other value as a Transmission Error.
17.4.1.5	The payload SHALL consist of one of the Commands or Responses specified in Section 17.5, and SHALL follow the SoD, as shown in Figure 39.
17.4.1.6	The payload, i.e., Commands and Responses specified in Section 17 (NFC-DEP Protocol), SHALL be followed by an EoD at the position shown in Figure 39. • The EoD SHALL contain a CRC_A (defined in Section 6.4), if the Peer is using NFC-A Technology. CRC_1 is the LSB and CRC_2 is the MSB. • The EoD SHALL contain a CRC_F (defined in Section 8.4), if the Peer is using NFC-F Technology. CRC_1 is the MSB and CRC_2 is the LSB.
17.4.1.7	The Peer SHALL verify a checksum contained in the EoD as follows: • If the Peer is using NFC-A Technology, then the Peer SHALL verify a CRC_A on receipt (as defined in Section 6.4), and SHALL treat failure of verification as a Transmission Error. • If the Peer is using NFC-F Technology, then the Peer SHALL verify a CRC_F on receipt (as defined in Section 8.4), and SHALL treat failure of verification as a Transmission Error.

NOTE Because all Commands and Responses in Section 17.5 contain two command bytes, the payload consists of at least 2 bytes.

17.5 Command Set

Payloads exchanged between Peers consist of Commands and Responses. Table 81 lists the Commands that are available to the Initiator. The corresponding Response from the Target is indicated for each Command.

Table 81: NFC-DEP Protocol Command Set

Initiator (Command)	Target (Response)
ATR_REQ	ATR_RES
PSL_REQ	PSL_RES
DEP_REQ	DEP_RES
DSL_REQ	DSL_RES
RLS_REQ	RLS_RES

NOTE The DSL_REQ Command and DSL_RES Response are not defined for Active Communication Mode. This definition deviates from [ISO/IEC_18092].

NOTE The WUP_REQ Command and WUP_RES Response are not defined in this specification. This definition deviates from [ISO/IEC_18092].

This section details the formats of these Commands and their Responses.

Requirements 243: NFC-DEP Bit Rates

Initiator		Target	
17.5.1.1	In the Passive communication Mode the Initiator SHALL send the ATR_REQ Command using the same Technology and bit rate as the SENS_REQ Command or the SENSF_REQ Command.	17.5.1.2	In the Passive Communication Mode the Target SHALL accept the ATR_REQ Command using the same Technology and bit rate as the SENS_REQ Command or the SENSF_REQ Command.
		17.5.1.3	In Active Communication Mode the Target SHALL accept the ATR_REQ Command in NFC-A and NFC-F Technology and all associated bit rates.
17.5.1.4	The Initiator SHALL accept the ATR_RES Response using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.	17.5.1.5	The Target SHALL send the ATR_RES Response using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.

17.5.1.6	Following an ATR_REQ Command, the Initiator SHALL send the next Command using the same Technology and bit rate as the ATR_REQ Command.	17.5.1.7	The Target SHALL accept the Command after the ATR_REQ Command using the same Technology and bit rate as the ATR_REQ Command.
17.5.1.8	The Initiator SHALL accept the Response to the Command after the ATR_REQ Command using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.	17.5.1.9	The Target SHALL send the Response to the Command after the ATR_REQ Command using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.
17.5.1.10	If PSL_REQ/RES has been exchanged, the Initiator SHALL send all subsequent Commands using $\mathbf{D}_{POLL \rightarrow LISTEN}$,according to the value of the Dsi parameter in the PSL_REQ Command.	17.5.1.11	If PSL_REQ/RES has been exchanged, the Target SHALL accept all subsequent Commands using $\mathbf{D}_{POLL \rightarrow LISTEN}$, according to the value of the Dsi parameter in the PSL_REQ Command.
17.5.1.12	If PSL_REQ/RES has been exchanged, the Initiator SHALL accept all subsequent Responses using $\mathbf{D}_{LISTEN \rightarrow POLL}$,according to the value of the Dri parameter in the PSL_REQ Command.	17.5.1.13	If PSL_REQ/RES has been exchanged, the Target SHALL send all subsequent Responses using $\mathbf{D}_{LISTEN \rightarrow POLL}$, according to the value of the Dri parameter in the PSL_REQ Command.
17.5.1.14	If PSL_REQ/RES has not been exchanged, the Initiator SHALL send all subsequent Commands using the same Technology and bit rate as the ATR_REQ Command.	17.5.1.15	If PSL_REQ/RES has not been exchanged, the Target SHALL accept all subsequent Commands using the same Technology and bit rate as the ATR_REQ Command.
17.5.1.16	If PSL_REQ/RES has not been exchanged, the Initiator SHALL accept all subsequent Responses using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.	17.5.1.17	If PSL_REQ/RES has not been exchanged, the Target SHALL send all subsequent Responses using the same value of $\mathbf{D}_{LISTEN \rightarrow POLL}$ as the value of $\mathbf{D}_{POLL \rightarrow LISTEN}$ used in the ATR_REQ Command.

NOTE The Initiator can only change the bit rate by using the PSL_REQ Command, as defined in [ACTIVITY].

NOTE The SENS_REQ Command uses $\mathbf{D}_{POLL \rightarrow LISTEN}$ equal to 1. The SENSF_REQ Command uses $\mathbf{D}_{POLL \rightarrow LISTEN}$ equal to 2 or 4.

17.6 Attribute Request (ATR_REQ)

The Attribute Request Command (ATR_REQ) is used by the Initiator to activate a Target. The Target responds to the ATR_REQ Command with the Attribute Request Response (ATR_RES).

17.6.1 ATR_REQ Length

Requirements 244: Attribute Request Length

Initiator		Target	
17.6.1.1	The number of bytes of the ATR_REQ Command SHALL be less than or equal to 64.	17.6.1.2	The Target SHALL treat as a Syntax Error an ATR_REQ Command that contains more than 64 bytes.
17.6.1.3	The Initiator SHALL treat as a Syntax Error an ATR_RES Response that contains more than 64 bytes.	17.6.1.4	The number of bytes of the ATR_RES Response SHALL be less than or equal to 64.

17.6.2 ATR_REQ Command

Table 82 defines the ATR_REQ format.

Table 82: ATR_REQ Command Format

Byte 1	Byte 2	Byte 3–12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17–17+n
D4h	00h	$\mathbf{NFCID3_I}$	$\mathbf{DID_I}$	$\mathbf{BS_I}$	$\mathbf{BR_I}$	$\mathbf{PP_I}$	[$\mathbf{G_I0}$ ... $\mathbf{G_In}$]

Requirements 245: ATR_REQ Command

Initiator		Target	
17.6.2.1	The Initiator SHALL send the ATR_REQ Command formatted as specified in Table 82.	17.6.2.2	The Target SHALL be ready to receive an ATR_REQ Command that has the format specified in Table 82.

$\mathbf{NFCID3_I}$

$NFCID3_I$ is the NFC Forum Device identifier of the Initiator for the NFC-DEP Protocol.

Requirements 246: NFCID3$_I$ Format

Initiator		Target	
17.6.2.3	If the Initiator is using NFC-F Technology in Passive Communication Mode, the Initiator SHALL fill Byte 3 through Byte 10 of ATR_REQ with NFCID2 of the Target it wants to address. Otherwise the Initiator MAY fill Byte 3 through Byte 10 of ATR_REQ with any value. Byte 11 through Byte 12 of ATR_REQ MAY be filled with any value.	17.6.2.4	If the Target is using NFC-F Technology in Passive Communication Mode, the Target SHALL ignore an ATR_REQ with Byte 3 through Byte 10 different from its NFCID2. Otherwise the Target SHALL accept any value of Byte 3 through Byte 10 of an ATR_REQ. The Target SHALL accept any value of Byte 11 through Byte 12 of an ATR_REQ.

NOTE Filling Byte 3 through Byte 10 of ATR_REQ deviates from the definitions in [ISO/IEC_18092].

DID$_I$

The Initiator Device Identification number (DID$_I$) is used to identify different Targets that are activated at one time.

Requirements 247: Initiator Device Identification Number (DID$_I$)

Initiator		Target	
17.6.2.5	In Passive Communication Mode the Initiator SHALL set the DID in the DID$_I$ field to a value in the range 0 to 14. In Active Communication Mode the Initiator SHALL set the DID$_I$ field to 0.	17.6.2.6	The Target SHALL be ready to receive a DID value in the range 0 to 14 in the DID$_I$ field transmitted by the Initiator. The Target MAY treat as a Syntax Error a DID$_I$ field whose DID value is not in the range 0 to 14.
17.6.2.7	The Initiator SHALL set the DID$_I$ equal to 0 if multiple target activation is not used.		

BS$_I$ and BR$_I$

BS_I and BR_I indicate the bit rates in Active Communication mode that are supported by the Initiator in both transmission directions. The coding of BS_I and BR_I is specified in Table 83 and Table 84.

Table 83: Bit Rates Supported by Initiator in the Sending Direction (BS_I)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x				$D_{POLL \rightarrow LISTEN}$ =64 supported, if bit is set to 1b.
					x			$D_{POLL \rightarrow LISTEN}$ =32 supported, if bit is set to 1b.
						x		$D_{POLL \rightarrow LISTEN}$ =16 supported, if bit is set to 1b.
							x	$D_{POLL \rightarrow LISTEN}$ =8 supported, if bit is set to 1b.

Table 84: Bit Rates Supported by Initiator in the Receiving Direction (BR_I)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x				$D_{LISTEN \rightarrow POLL}$ =64 supported, if bit is set to 1b.
					x			$D_{LISTEN \rightarrow POLL}$ =32 supported, if bit is set to 1b.
						x		$D_{LISTEN \rightarrow POLL}$ =16 supported, if bit is set to 1b.
							x	$D_{LISTEN \rightarrow POLL}$ =8 supported, if bit is set to 1b.

Requirements 248: BS_I and BR_I Coding

Initiator		Target	
17.6.2.8	The Initiator SHALL set bits b4 through b1 of the BS_I and BR_I fields according to Table 83 and Table 84.	17.6.2.9	The Target SHALL accept any values in the BS_I and BR_I fields.

Presence of Optional Parameters (PP_I)

The PP_I field indicates the Length Reduction field (LR_I) and the presence of optional parameters. The format of the PP_I byte is specified in Table 85.

Table 85: PP_I Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		x	x					LR_I
				0				RFU
					0			RFU
						x		General Bytes (G_I) 0b: General Bytes (G_I) not available 1b: General Bytes (G_I) available
							x	NAD 0b: NAD not used 1b: NAD used

The Length Reduction (LR_I) bits restrict the size of the Responses (payload) of the Target. Table 86 defines the Length Reduction coding for LR_I, with b6 as the msb and b5 as the lsb.

Table 86: Length Reduction Coding

msb	lsb	Meaning
0	**0**	Maximum payload size is 64 bytes.
0	**1**	Maximum payload size is 128 bytes.
1	**0**	Maximum payload size is 192 bytes.
1	**1**	Maximum payload size is 254 bytes.

Requirements 249: PP_I Format

Initiator		Target	
17.6.2.10	The Initiator SHALL format and code PP_I according to Table 85 and Table 86.	17.6.2.11	The Target SHALL be ready to receive PP_I that has the format and coding specified in Table 85 and Table 86.

Requirements 250: LR_I

Initiator	
17.6.2.12	The Initiator SHALL be ready to receive Responses that contain a number of payload bytes less than or equal to the value specified by **LR_I**.

The NAD bit indicates whether the Initiator uses NAD or not.

Requirements 251: NAD

Initiator		Target	
17.6.2.13	The Initiator SHALL NOT use NAD and set b1 of PP_I to 0b.	17.6.2.14	The Target SHALL disregard b1 of PP_I.

General Bytes (G_I)

The General Bytes (G_I) are optional and are used to provide general information. The maximum number of General Bytes is calculated by the maximum number of data bytes in the ATR_REQ Command, minus the number of mandatory data bytes in the ATR_REQ Command.

17.6.3 ATR_RES Response

Table 87 defines the ATR_RES format.

Table 87: ATR_RES Response Format

Byte 1	Byte 2	Byte 3–12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18–18+n
D5h	01h	**$NFCID3_T$**	**DID_T**	**BS_T**	**BR_T**	TO	**PP_T**	[**G_T0** ... **G_Tn**]

Requirements 252: ATR_RES Response

Initiator		Target	
17.6.3.1	The Initiator SHALL be ready to receive an ATR_RES Response that has the format specified in Table 87.	17.6.3.2	The Target SHALL send the ATR_RES Response formatted as specified in Table 87.

$NFCID3_T$

$NFCID3_T$ is the NFC Forum Device identifier of the Target for the NFC-DEP Protocol.

Requirements 253: $NFCID3_T$ Format

Initiator		Target	
17.6.3.3	The Initiator MAY treat as a Syntax Error an $NFCID3_T$ format that does not meet Requirement 17.6.3.4.	17.6.3.4	In Passive Communication Mode the following rules apply: If the Target is using NFC-F Technology, the Target SHALL fill Byte 3 through Byte 10 of ATR_RES with its NFCID2. Otherwise Byte 3 through Byte 10 of ATR_RES MAY be filled with any value. Byte 11 and Byte 12 of ATR_RES MAY be filled with any value.
		17.6.3.5	In Active Communication Mode the following rules apply: Byte 3 through Byte 10 of ATR_RES MAY be filled with any value.

NOTE Filling Byte 3 through Byte 10 of ATR_RES deviates from the definitions in [ISO/IEC_18092].

DID_T

In the DID_T field the Target returns the same value as was received in the DID_I field of the preceding ATR_REQ Command.

Requirements 254: Target Device Identification Number (DID_T)

Initiator		Target	
17.6.3.6	The Initiator SHALL be ready to receive an ATR_RES Response that contains a DID_T byte with the same value as sent in the DID_I field of the preceding ATR_REQ Command. The Initiator MAY treat as a Syntax Error an ATR_RES Response that contains a different DID_T byte.	17.6.3.7	In the DID_T field the Target SHALL return the same value as was received in the DID_I field of the preceding ATR_REQ Command.

BS_T and BR_T

BS_T and BR_T indicate the bit rates in Active Communication mode supported by the Target in both transmission directions. The formats of BS_T and BR_T are specified in Table 88 and Table 89.

Table 88: Bit Rates Supported by Target in Sending Direction (BS_T)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x				$D_{LISTEN \rightarrow POLL}$ =64 supported, if bit is set to 1b.
					x			$D_{LISTEN \rightarrow POLL}$ =32 supported, if bit is set to 1b.
						x		$D_{LISTEN \rightarrow POLL}$ =16 supported, if bit is set to 1b.
							x	$D_{LISTEN \rightarrow POLL}$ =8 supported, if bit is set to 1b.

Table 89: Bit Rates Supported by Target in Receiving Direction (BR_T)

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x				$D_{POLL \rightarrow LISTEN}$ =64 supported, if bit is set to 1b.
					x			$D_{POLL \rightarrow LISTEN}$ =32 supported, if bit is set to 1b.
						x		$D_{POLL \rightarrow LISTEN}$ =16 supported, if bit is set to 1b.
							x	$D_{POLL \rightarrow LISTEN}$ =8 supported, if bit is set to 1b.

Requirements 255: BS_T and BR_T Format

Initiator		Target	
17.6.3.8	The Initiator SHALL accept any values in the BS_T and BR_T fields.	17.6.3.9	The Target SHALL set bits b4 through b1 of the BS_T and BR_T fields according to Table 83 and Table 84.

TO

The least significant nibble b4 through b1 of the TO field is the Waiting Time (WT) and is used by the Target to code the Response Waiting Time (RWT). The format of the TO field is specified in Table 90. Section 17.11 defines the RWT. WT has a value in the range from 0 to 14. The value 15 is RFU.

Table 90: TO Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		0						RFU
			0					RFU
				x	x	x	x	WT

Requirements 256: TO Format

Initiator		Target	
17.6.3.10	The Initiator SHALL be ready to receive a TO that has the format specified in Table 90.	17.6.3.11	The Target SHALL format TO according to Table 85 and Table 90.

Requirements 257: WT

Initiator		Target	
17.6.3.12	The Initiator SHALL support a TO that indicates a WT value less than or equal to $\mathbf{WT}_{NFCDEP,MAX}$. Appendix B.10 lists the value of $\mathbf{WT}_{NFCDEP,MAX}$.	17.6.3.13	The Target SHALL set the value of WT less than or equal to $\mathbf{WT}_{NFCDEP,MAX}$. Appendix B.10 lists the value of $\mathbf{WT}_{NFCDEP,MAX}$.
17.6.3.14	A received WT value of RFU SHALL be treated by the Initiator as WT=14.		

NOTE Higher layer MAC mappings can override the value of $\mathbf{WT}_{NFCDEP,MAX}$ listed in Appendix B.10.

Presence of Optional Parameters (PP_T)

The PP_T field includes the Length Reduction field (LR_T) and the presence of optional parameters. The format of the PP_T byte is specified in Table 91.

Table 91: PP_T Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							RFU
		x	x					LR_T
				0				RFU
					0			RFU
						x		General Bytes (G_T) available, if set to 1b
							x	NAD used, if set to 1b.

Requirements 258: PP_T Format

Initiator		Target	
17.6.3.15	The Initiator SHALL be ready to receive the PP_T that has the format and coding specified in Table 91.	17.6.3.16	The Target SHALL format and code the PP_T according to Table 91.

The Length Reduction (LR_T) bits restrict the size of the Commands (payload) of the Initiator. Table 86 defines the Length Reduction coding for LR_T, with b6 as the msb and b5 as the lsb.

Requirements 259: LR_T

Target	
17.6.3.17	The Target SHALL be ready to receive Commands that contain a number of payload bytes less than or equal to the value specified by LR_T.

The NAD bit indicates whether the Target uses NAD or not.

Requirements 260: NAD

Initiator		Target	
17.6.3.18	The Initiator SHALL ignore b1 of PP_T.	17.6.3.19	The Target SHALL NOT use NAD and SHALL set bit b1 of PP_T to 0b.

General Bytes

General Bytes (G_T) are optional and are used to provide general information. The maximum number of General Bytes is calculated by the maximum number of data bytes in the ATR_RES Response, minus the number of mandatory data bytes in the ATR_RES Response.

17.7 Parameter Selection Request (PSL_REQ)

The PSL_REQ Command is used to switch communication parameters for the subsequent data exchange through the NFC-DEP Protocol.

17.7.1 PSL_REQ Command

The format of the PSL_REQ Command is specified in Table 92.

Table 92: PSL_REQ Command Format

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
D4h	04h	DID	BRS	FSL

Requirements 261: PSL_REQ Command

Initiator		Target	
17.7.1.1	The Initiator SHALL send the PSL_REQ Command formatted as specified in Table 92.	17.7.1.2	The Target SHALL be ready to receive an PSL_REQ Command that has the format specified in Table 92.

DID

Requirements 262: PSL_REQ Command DID

Initiator		Target	
17.7.1.3	The Initiator SHALL send the same DID field value as was received in the preceding ATR_REQ.	17.7.1.4	The Target SHALL accept a PSL_REQ Command with the same DID field value as was received in the preceding ATR_REQ Command. The Target SHALL ignore a PSL_REQ Command that contains a DID field value different from the one received in the preceding ATR_REQ.

BRS

The BRS byte specifies the selected bit rates for Initiator and Target, and is formatted as shown in Table 93.

Table 93: BRS Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							Fixed Value
		x	x	x				Dsi
					x	x	x	Dri

The Dsi codes the communication bit rate in the Initiator to Target direction. The Dri codes the communication bit rate in the Target to Initiator direction, as specified in Table 94.

Table 94: Dsi and Dri Coding

b6 b3	b5 b2	b4 (Dsi) b1 (Dri)	Divisor D
0	0	0	1
0	0	1	2
0	1	0	4
All other values			Not Allowed

Requirements 263: PSL_REQ Command BRS

Initiator		Target	
17.7.1.5	The Initiator SHALL set BRS to a value that is compliant with Table 93 and Table 94.	17.7.1.6	The Target SHALL be ready to receive a BRS that is compliant with Table 93 and Table 94.
17.7.1.7	The Initiator SHALL set Dsi and Dri to the same value.	17.7.1.8	A received Dri value that is different from the Dsi value SHALL be treated as a Syntax Error.
		17.7.1.9	A received Dsi and Dri value of Not Allowed MAY be accepted.

NOTE Requirements 17.7.1.7 and 17.7.1.8 deviate from [ISO/IEC_18092], where different values for Dsi and Dri are allowed.

NOTE Previous versions of this specification allowed different values for Dsi and Dri.

FSL

The FSL byte defines the maximum length of Commands and Responses (number of payload bytes), as specified in Table 95. If no PSL_REQ Command is sent, then the value exchanged in ATR_REQ/RES is used.

Table 95: FSL Format

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0			Fixed Value
						x	x	LR

The Length Reduction (LR) bits restrict the size of the Commands (payload) of both the Initiator and Target. Table 86 defines the Length Reduction coding for LR, with b2 as the msb and b1 as the lsb.

Requirements 264: FSL

Initiator		Target	
17.7.1.10	The Initiator SHALL set FSL to a value according to Table 95 which defines a maximum length of Commands and Responses that is less than or equal to the minimum of the lengths defined by LR_I and LR_T.	17.7.1.11	The Target SHALL be ready to receive a FSL value that is compliant with Table 95. If FSL is greater than the minimum of the lengths defined by LR_I and LR_T, then the Target MAY treat this as a Syntax Error.
17.7.1.12	The Initiator SHALL be ready to receive Responses that contain a number of payload bytes less than or equal to the value specified by FSL. The Initiator MAY treat as a Syntax Error a Response that contains a number of payload bytes greater than the value specified by FSL.	17.7.1.13	The Target SHALL be ready to receive Commands that contain a number of payload bytes less than or equal to the value specified by FSL. The Target MAY treat as a Syntax Error a Command that contains a number of payload bytes greater than the value specified by FSL.

17.7.2 PSL_RES Response

Table 96 defines the format of the PSL_RES Response.

Table 96: PSL_RES Response Format

Byte 1	Byte 2	Byte 3
D5h	05h	DID

Requirements 265: PSL_RES Response

Initiator		Target	
17.7.2.1	The Initiator SHALL be ready to receive a PSL_RES Response that has the format specified in Table 96.	17.7.2.2	The Target SHALL send the PSL_RES Response formatted as specified in Table 96.

DID

Requirements 266: PSL_RES Response DID

Initiator		Target	
17.7.2.3	The Initiator SHALL accept a PSL_RES Response that contains the same DID field value as was sent in the PSL_REQ Command. The Initiator SHALL treat a PSL_RES Response containing a different DID field value as a Syntax Error.	17.7.2.4	The Target SHALL return the same DID field value as it received in the PSL_REQ Command.

17.8 Data Exchange Protocol Request (DEP_REQ)

The Data Exchange Protocol Request Command (DEP_REQ) is used by the Initiator to exchange data with a Target that is configured for the NFC-DEP Protocol. The Target responds to the DEP_REQ with the Data Exchange Protocol Request Response (DEP_RES).

17.8.1 DEP_REQ Command

Table 97 defines the format of the DEP_REQ Command.

Table 97: DEP_REQ Command Format

Data Exchange Protocol Header					Transport Data Bytes
Byte 1	**Byte 2**	**Byte 3**	**Byte 4 (optional)**	**Byte 5 (optional)**	**Byte 6 – 6+k-1 (optional)**
D4h	06h	PFB	[DID]	[NAD]	[Data Byte 1] … [Data Byte k]

Parameter k denotes the variable number of transport data bytes. If optional bytes are not present in the DEP_REQ Command, then the position of the subsequent bytes is adapted accordingly (e.g., if the DID field and NAD field are not present in the DEP_REQ Command, then the first transport data byte is Byte 4 instead of Byte 6).

The format of the Protocol Format Byte (PFB) is specified in Section 17.8.3.

Requirements 267: DEP_REQ Command

Initiator		Target	
17.8.1.1	The Initiator SHALL send the DEP_REQ Command formatted as specified in Table 97.	17.8.1.2	The Target SHALL be ready to receive a DEP_REQ Command that has the format specified in Table 97.

DID

Requirements 268: DEP_REQ Command DID

Initiator		Target	
17.8.1.3	If the DID field value sent in the preceding ATR_REQ is not equal to 0, the Initiator SHALL include the same DID field value in the DEP_REQ Command. If the DID field value sent in the preceding ATR_REQ is equal to 0, the Initiator SHALL include no DID field in the DEP_REQ Command.	17.8.1.4	If the DID field value received in the preceding ATR_REQ is not equal to 0, then: • The Target SHALL accept a DEP_REQ Command with the same DID field value. • The Target SHALL ignore a DEP_REQ Command with a different DID field value. If the DID field value received in the preceding ATR_REQ is 0, then: • The Target SHALL accept a DEP_REQ Command that does not contain a DID field. • The Target SHALL ignore a DEP_REQ Command that contains a DID field.

17.8.2 DEP_RES Response

Table 98 defines the format of the DEP_RES Response.

Table 98: DEP_RES Response Format

Data Exchange Protocol Header					Transport Data Bytes
Byte 1	**Byte 2**	**Byte 3**	**Byte 4 (optional)**	**Byte 5 (optional)**	**Byte 6 – 6+k-1 (optional)**
D5h	07h	PFB	[DID]	[NAD]	[Data Byte 1] ... [Data Byte k]

Parameter k denotes the variable number of transport data bytes. If optional bytes are not present in the DEP_RES Response, then the position of the subsequent bytes is adapted accordingly (e.g., if the DID field and NAD field are not present in the DEP_RES Response, then the first transport data byte is Byte 4 instead of Byte 6).

The format of the PFB is specified in Section 17.8.3.

Requirements 269: DEP_RES Response

Initiator		Target	
17.8.2.1	The Initiator SHALL be ready to receive a DEP_RES Response that has the format specified in Table 98.	17.8.2.2	The Target SHALL send the DEP_RES Response formatted as specified in Table 98.

DID

Requirements 270: DEP_RES Response DID

Initiator		Target	
17.8.2.3	If the Initiator sent a DID field in the DEP_REQ, the Initiator SHALL accept a DEP_RES Response with the DID field, and that field has the same value as the DID field in the DEP_REQ. The Initiator SHALL treat as a Syntax Error a DEP_RES Response that contains a different value in its DID field. If the Initiator did not send a DID field in the DEP_REQ, a DEP_RES Response that contains a DID field SHALL be treated as a Syntax Error.	17.8.2.4	If the DEP_REQ contained a DID field, the Target SHALL return the same value of the DID field in the DEP_RES Response. Otherwise the Target SHALL NOT return a DID field in the DEP_RES Response.

17.8.3 Protocol Format Byte (PFB)

The mandatory PFB is used to convey the information needed to control data transmission. The protocol defines three fundamental types of blocks, called the Protocol Data Units (PDUs):

- The Information PDU is used to convey application layer data in the transport data bytes. Application layer data is information for use by the adjacent upper layer.
- The Response PDU is used to convey positive or negative acknowledgements. A Response PDU never contains transport data bytes. The acknowledgement relates to the last received PDU.
- The Supervisory PDU is used to exchange control information between the Initiator and the Target. Two different types of Supervisory PDUs are defined:
  - RTOX PDU containing one transport data byte
  - ATN PDU containing no transport data bytes.

Table 99: PDU PFB Coding

b8	b7	b6	Meaning
0	0	0	Information PDU
0	1	0	Response PDU
1	0	0	Supervisory PDU
All other values			RFU

The format of the PFB depends on its type. The formats of the Information PDU, Response PDU and Supervisory PDU are shown in Table 100, Table 101 and Table 102, respectively.

Table 100: Information PDU PFB Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0						Indicates Information PDU
			x					MI: Chaining, if bit is set to 1b
				x				NAD following, if bit is set to 1b
					x			DID following, if bit is set to 1b
						x	x	PNI

The More Information (MI) bit indicates chaining, i.e., the current PDU contains only a part of a larger block of application layer data that is split up into several PDUs. Section 17.12.2 provides more details about chaining.

The Packet Number Information (PNI) bits contains the number of packets sent by the Initiator to the Target, and vice versa, starting with 0. These bits are used for error detection during protocol handling.

Table 101: Response PDU PFB Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	1	0						Indicates Response PDU
			x					0b: ACK (Acknowledged) 1b: NACK (Not acknowledged)
				x				NAD following, if bit is set to 1b
					x			DID following, if bit is set to 1b
						x	x	PNI

An ACK PDU is a Response PDU with bit b5 set to 0b and a NACK PDU is a Response PDU with bit b5 set to 1b.

Table 102: Supervisory PDU PFB Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	0	0						Indicates Supervisory PDU
			x					0b: ATN (Attention) 1b: RTOX (Response Timeout Extension)
				x				NAD following, if bit is set to 1b
					x			DID following, if bit is set to 1b
						0		RFU
							0	RFU

An ATN PDU is a Supervisory PDU with bit b5 set to 0b, and a RTOX PDU is a Supervisory PDU with bit b5 set to 1b.

Requirements 271: Protocol Format Byte (PFB)

Initiator and Target	
17.8.3.1	The PFB of an Information PDU SHALL be coded and interpreted as defined in Table 100.
17.8.3.2	The PFB of a Response PDU SHALL be coded and interpreted as defined in Table 101.
17.8.3.3	The PFB of a Supervisory PDU SHALL be coded and interpreted as defined in Table 102.
17.8.3.4	When it is sending, the Peer SHALL set bit b3 to 1b if the DID field is present in the PDU.
17.8.3.5	When it is receiving, the Peer SHALL be ready to receive a DID field if bit b3 of the PFB is equal to 1b.
17.8.3.6	When it is sending, the Peer SHALL set bit b4 to 1b if the NAD field is present in the PDU.
17.8.3.7	When it is receiving, the Peer SHALL be ready to receive a NAD field if bit b4 of the PFB is equal to 1b.

17.8.4 Response Timeout Extension

When the Target needs more time than the defined RWT to process the received PDU, it sends an RTOX PDU to the Initiator. An RTOX PDU contains one transport data byte, as specified in Table 103.

After it has sent an RTOX Request, the Target will wait for a response. If it receives an RTOX Response, it continues operation from where it was before it sent the RTOX Request.

Table 103: RTOX PDU Transport Data Byte Coding

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0							RFU
		x	x	x	x	x	x	RTOX

The bits b6 through b1 identify the RTOX; these bits are coded in the range from 1 to 59. The value of 0 for RTOX is Not Allowed and the values of 60-63 are RFU.

The Initiator responds to an RTOX PDU by sending an RTOX PDU to the Target. The transport data byte contains the same RTOX value as was received in the RTOX PDU.

For clarity, an RTOX PDU sent by the Target is referred to as an RTOX Request and an RTOX PDU sent by the Initiator is referred to as an RTOX Response.

The corresponding RWT intermediate value is calculated using the following formula:

$$\mathbf{RWT}_{\mathbf{INT}} = \mathrm{RWT} \times \mathrm{RTOX}$$

The time period $\mathbf{RWT}_{\mathbf{INT}}$ requested by the Target starts after the Initiator has sent the RTOX Response.

Requirements 272: Response Timeout Extension

Initiator		Target	
17.8.4.1	The Initiator SHALL be ready to receive an RTOX Request with the Transport Data Byte coded as specified in Table 103.	17.8.4.2	The Target SHALL code the Transport Data Byte of an RTOX Request according to Table 103.
17.8.4.3	The Initiator SHALL be ready to receive an RTOX Request with an RTOX field that contains a value in the range from 1 to 59. If it receives an RTOX Request with the value of the RTOX field set to 0 or in the range 60 to 63, the Initiator SHALL treat that request as a Syntax Error.	17.8.4.4	The Target SHALL code the RTOX field in the range from 1 to 59.
17.8.4.5	The Initiator SHALL support a response waiting time extension less than or equal to the maximum supported RWT: $\mathbf{RWT}_{INT} \leq (256 \times 16/\mathbf{f}_C) \times 2^{\mathbf{WT}_{NFCDEP,MAX}}$. Appendix B.10 lists the value of $\mathbf{WT}_{NFCDEP,MAX}$.	17.8.4.6	The Target SHALL set the value of the RTOX field such that $\mathbf{RWT}_{INT}$ is less than or equal to the maximum value of RWT: $\mathbf{RWT}_{INT} \leq (256 \text{ x } 16/\mathbf{f}_C) \times 2^{\mathbf{WT}_{NFCDEP,MAX}}$. Appendix B.10 lists the value of $\mathbf{WT}_{NFCDEP,MAX}$.
17.8.4.7	The Initiator SHALL set the value of RTOX field in the RTOX Response to be the same value as the value of the RTOX field in the RTOX Request.	17.8.4.8	The Target SHALL treat as a Syntax Error an RTOX field in the RTOX Response that has a value different from the value of the RTOX field in the RTOX Request.

17.9 Deselect Request (DSL_REQ)

The Deselect Request Command (DSL_REQ) is used by the Initiator to deactivate a Target. The Target responds to the DSL_REQ with the Deselect Response (DSL_RES).

17.9.1 DSL_REQ Command

Table 104 defines the format of the DSL_REQ Command.

Table 104: DSL_REQ Command Format

Byte 1	Byte 2	Byte 3 (optional)
D4h	08h	[DID]

Requirements 273: DSL_REQ Command

Initiator		Target	
17.9.1.1	The Initiator SHALL send the DSL_REQ Command formatted as specified in Table 104.	17.9.1.2	The Target SHALL be ready to receive a DSL_REQ Command that has the format specified in Table 104.

DID

Requirements 274: DSL_REQ Command DID

Initiator		Target	
17.9.1.3	If the value of the DID field sent in the preceding ATR_REQ is different from 0, the Initiator SHALL include the same DID value in the DSL_REQ Command. If the value of the DID field sent in the preceding ATR_REQ is equal to 0, the Initiator SHALL NOT include a DID field in the DSL_REQ Command.	17.9.1.4	If the value of the DID field received in the preceding ATR_REQ is different from 0, then: • The Target SHALL accept a DSL_REQ Command with the same value in its DID field. • The Target SHALL ignore a DSL_REQ Command with a DID field that has a different value. If the value in the DID field received in the preceding ATR_REQ is 0, then: • The Target SHALL accept a DSL_REQ Command that does not contain a DID field. • The Target SHALL ignore a DSL_REQ Command that contains a DID field.

17.9.2 DSL_RES Response

Table 105 defines the format of the DSL_RES Response.

Table 105: DSL_RES Response Format

Byte 1	Byte 2	Byte 3 (optional)
D5h	09h	[DID]

Requirements 275: DSL_RES Response

Initiator		Target	
17.9.2.1	The Initiator SHALL be ready to receive a DSL_RES Response that has the format specified in Table 105.	17.9.2.2	The Target SHALL send the DSL_RES Response formatted as specified in Table 105.

DID

Requirements 276: DSL_RES Response DID

Initiator		Target	
17.9.2.3	If the Initiator sent a DID field in the DSL_REQ, the Initiator SHALL accept a DSL_RES Response that contains a DID field that has the same value. The Initiator SHALL treat as a Syntax Error a DSL_RES Response that contains a DID field containing a different value. If the Initiator did not send a DID field in the DSL_REQ, a DSL_RES Response that contains a DID field SHALL be treated as Syntax Error.	17.9.2.4	If the DSL_REQ contained a DID field, the Target SHALL return the same value in its DID field as in the DSL_RES Response. Otherwise the Target SHALL NOT return a DID field in the DSL_RES Response.

17.10 Release Request (RLS_REQ)

The Initiator uses the Release Request Command (RLS_REQ) to release a Target, which responds to the RLS_REQ with the Release Response (RLS_RES).

17.10.1 RLS_REQ Command

Table 106 defines the format of the RLS_REQ Command.

Table 106: RLS_REQ Command Format

Byte 1	Byte 2	Byte 3 (optional)
D4h	0Ah	[DID]

Requirements 277: RLS_REQ Command

Initiator		Target	
17.10.1.1	The Initiator SHALL send the RLS_REQ Command formatted as specified in Table 106.	17.10.1.2	The Target SHALL be ready to receive an RLS_REQ Command that has the format specified in Table 106.

DID

Requirements 278: RLS_REQ Command DID

Initiator		Target	
17.10.1.3	If the value of the DID field sent in the preceding ATR_REQ is different from 0, the Initiator SHALL include the same value in the DID field in the RLS_REQ Command. If the DID sent in the preceding ATR_REQ is equal to 0, the Initiator SHALL NOT include a DID field in the RLS_REQ Command.	17.10.1.4	If the value of the DID field received in the preceding ATR_REQ is not 0, then: • The Target SHALL accept an RLS_REQ Command with the same value in its DID field. • The Target SHALL ignore an RLS_REQ Command with a different value in its DID field. If the value of the DID field received in the preceding ATR_REQ is 0, then: • The Target SHALL accept RLS_REQ Command that does not contain a DID field. • The Target SHALL ignore an RLS_REQ Command that contains a DID field.

17.10.2 RLS_RES Response

Table 107 defines the format of the RLS_RES Response.

Table 107: RLS_RES Response Format

Byte 1	Byte 2	Byte 3 (optional)
D5h	0Bh	[DID]

Requirements 279: RLS_RES Response

Initiator		Target	
17.10.2.1	The Initiator SHALL be ready to receive an RLS_RES Response that has the format specified in Table 107.	17.10.2.2	The Target SHALL send the RLS_RES Response formatted as specified in Table 107.

DID

Requirements 280: RLS_RES Response DID

Initiator		Target	
17.10.2.3	If the Initiator sent a DID field in the RLS_REQ, the Initiator SHALL accept an RLS_RES Response that contains the same value in its DID field. The Initiator SHALL treat as a Syntax Error an RLS_RES Response that contains a different value in its DID field. If the Initiator did not send a DID in the RLS_REQ, an RLS_RES Response that contains a DID SHALL be treated as a Syntax Error.	17.10.2.4	If the RLS_REQ contained a DID field, the Target SHALL return the same value in its DID field as in the RLS_REQ. Otherwise the Target SHALL NOT return a DID field in the RLS_RES Response.

17.11 Timing Requirements

17.11.1 Frame Delay Time

Peers use NFC-A Frame Delay Time or NFC-F Frame Delay Time for the NFC-DEP Protocol, depending on the configuration.

Requirements 281: NFC-DEP Protocol Frame Timing

Initiator and Target	
17.11.1.1	If in Passive Communication Mode, the minimum Frame Delay Time Poll→Listen for Commands and Responses specified in Section 17 (NFC-DEP Protocol) SHALL be set to either: • The minimum Frame Delay Time Poll→Listen defined in Section 6.10.1, if the Peer is using NFC-A Technology • The minimum Frame Delay Time Poll→Listen defined in Section 8.7.1, if the Peer is using NFC-F Technology.
17.11.1.2	If in Passive Communication Mode, the minimum Frame Delay Time Listen→Poll for Commands and Responses specified in Section 17 (NFC-DEP Protocol) SHALL be set to either: • The minimum Frame Delay Time Listen→Poll defined in Section 6.10.3, if the Peer is using NFC-A Technology • The minimum Frame Delay Time Listen→Poll defined in Section 8.7.3, if the Peer is using NFC-F Technology.

NOTE Timings for Active Communication Mode are defined in [ACTIVITY].

17.11.2 Response Waiting Time

The Response Waiting Time (RWT) defines the maximum amount of time a Target has between the end of a Poll Frame and the SoD of its Response. It is calculated using the following formula:

$$RWT = (256 \times 16/\mathbf{f}_C) \times 2^{WT},$$

where the value of WT has the range from 0 to 14. WT is included in the ATR_RES Response, as specified in Section 17.6.3.

NOTE The RWT and the Frame Delay Time Poll→Listen are both started after the receipt of a Poll Frame. The Target sends the Response in the time interval between the minimum Frame Delay Time Poll→Listen and RWT. For the Frame Delay Time the SoF of the Response is relevant (as defined in Section 17.3), whereas for the RWT it is the SoD (defined in Section 17.4).

The RWT does not impose any limit on the timing of the Initiator; the delay before the Initiator sends the next Command is only constrained by the relevant Frame Delay Time Listen→Poll.

NOTE The actual response time window of a Target in Active Communication Mode to send its response is impacted by the timings of the Initiator. Therefore the value for WT has to be selected with care.

The following specific RWT is defined for the Technology Detection Activity in Active Communication Mode:

- $\mathbf{RWT}_{ACM,NFCDEP,ACTIVATION}$

  A specific RWT is defined for the ATR_REQ Command in Active Communication Mode. For this Command the Target starts to send its Response frame within $\mathbf{RWT}_{ACM,NFCDEP,ACTIVATION}$ (response waiting time). Section 17.6.2 defines the ATR_REQ Command.

The following specific RWTs are defined for the Device Activation Activity and the Data Exchange Activity:

- $\mathbf{RWT}_{NFCDEP,ACTIVATION}$

  A specific RWT is defined for the ATR_REQ Command in Passive Communication Mode. For this Command the Target starts to send its Response frame within $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ (response waiting time). Section 17.6.2 defines the ATR_REQ Command.

NOTE The response waiting time $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ is introduced and defined in this document.

- $\mathbf{RWT}_{INT}$

  When the Target sends an RTOX Request with a specific RTOX value, the Initiator waits for $\mathbf{RWT}_{INT}$ for the next PDU. Section 17.8.4 defines $\mathbf{RWT}_{INT}$.

Requirements 282: Response Waiting Time

Initiator		Target	
17.11.2.1	Following the ATR_REQ Command in Passive Communication Mode, the Initiator SHALL wait at least $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$ for the SoD of the ATR_RES Response. During the interval between the end of the time period $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$ and the end of the time period $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$ + $\mathbf{\Delta T}_{NFCDEP,INITIATOR}$, the Initiator MAY accept the ATR_RES Response or MAY generate a Timeout Error. If the Initiator does not receive the SoD of the ATR_RES Response before the end of the time period $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$ + $\mathbf{\Delta T}_{NFCDEP,INITIATOR}$, then the Initiator SHALL treat the lack of response as a Timeout Error. Appendix B.10 lists the values of $\mathbf{\Delta RWT}_{NFCDEP}$ and $\mathbf{\Delta T}_{NFCDEP,INITIATOR}$.	17.11.2.2	Following the EoD of the ATR_REQ Command in Passive Communication Mode, the Target SHALL wait no longer than $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ before it sends the SoD of the ATR_RES Response. Appendix B.10 lists the value of $\mathbf{RWT}_{NFCDEP,ACTIVATION}$.

Initiator		Target	
17.11.2.3	Following the ATR_REQ Command in Active Communication Mode, the Initiator SHALL wait at least $\mathbf{RWT}_{ACM,NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$ for the SoD of the ATR_RES Response. If the Initiator does not receive the SoD of the ATR_RES Response by the end of the time period $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ + $\mathbf{\Delta RWT}_{NFCDEP}$, then the Initiator SHALL treat the lack of response as a Timeout Error.	17.11.2.4	Following the EoD of the ATR_REQ Command in Active Communication Mode, the Target SHALL wait no longer than $\mathbf{RWT}_{ACM,NFCDEP,ACTIVATION}$ before it sends the SoD of the ATR_RES Response.
17.11.2.5	Except for the ATR_REQ Command, the Initiator SHALL wait at least RWT + $\mathbf{\Delta RWT}_{NFCDEP}$ for the SoD of the Response from the Target. During the interval between the end of the time period RWT + $\mathbf{\Delta RWT}_{NFCDEP}$ and the end of the time period RWT + $\mathbf{\Delta RWT}_{NFCDEP}$ + $\mathbf{\Delta T}_{NFCDEP,INITIATOR}$, the Initiator MAY accept the Response or MAY generate a Timeout Error. If the Initiator does not receive the SoD of the Response from the Target by the end of the time period RWT + $\mathbf{\Delta RWT}_{NFCDEP}$ + $\mathbf{\Delta T}_{NFCDEP,INITIATOR}$, then the Initiator SHALL treat the lack of response as a Timeout Error.	17.11.2.6	Except for the ATR_REQ Command following the EoD of a Command, the Target SHALL wait no longer than RWT before it sends the SoD of the Response.

Initiator		Target	
17.11.2.7	After the Initiator sends the RTOX Response PDU in response to an RTOX Request PDU from the Target, the Initiator SHALL wait for $\mathbf{RWT}_{INT} + \Delta\mathbf{RWT}_{NFCDEP}$ for the SoD of a PDU of the Target. During the interval between the end of the time period $\mathbf{RWT}_{INT} + \Delta\mathbf{RWT}_{NFCDEP}$ and the end of the time period $\mathbf{RWT}_{INT} + \Delta\mathbf{RWT}_{NFCDEP} + \Delta\mathbf{T}_{NFCDEP,INITIATOR}$, the Initiator MAY accept the PDU or MAY generate a Timeout Error. If the Initiator does not receive the SoD of a PDU from the Target before the end of the time period $\mathbf{RWT}_{INT} + \Delta\mathbf{RWT}_{NFCDEP} + \Delta\mathbf{T}_{NFCDEP,INITIATOR}$, then the Initiator SHALL treat the lack of response as a Timeout Error.	17.11.2.8	After the Target receives the RTOX Response PDU of the Initiator, the Target SHALL send the SoD of the next PDU within $\mathbf{RWT}_{INT}$.
17.11.2.9	The Initiator SHALL apply $\mathbf{RWT}_{INT} + \Delta\mathbf{RWT}_{NFCDEP}$ only until the next PDU has been received from the Target.	17.11.2.10	The Target SHALL apply $\mathbf{RWT}_{INT}$ only until the next PDU has been sent by the Target.
		17.11.2.11	A Target in Active Communication Mode SHALL set the value of WT larger than 1.

For NFC-F frame timing the $\mathbf{FDT}_{F,LISTEN,MAX}$ defined in Section 8.7.1 equals **RWT** - 16 bd (where 16 bd is the bit duration of the SoF). Similar for the ATR_REQ, the $\mathbf{FDT}_{F,LISTEN,MAX,ATR}$ equals $\mathbf{RWT}_{NFCDEP,ACTIVATION}$ - 16 bd.

17.12 NFC-DEP Protocol Operation

NFC-DEP Protocol operates by exchanging PDUs between an Initiator and a Target as the payload of DEP_REQ Commands and DEP_RES Responses.

17.12.1 General Rules

General rules for the NFC-DEP Protocol are described in Section 10.

In addition to these rules, there are general rules which apply to PDU exchange.

Requirements 283: General PDU Handling Rules

Initiator		Target	
17.12.1.1	The Initiator SHALL send the first PDU which SHALL be either an Information PDU to initiate a Data Exchange Initiator to Target (see 17.12.2) or an ATN PDU to initiate a Presence Check (see 17.12.6).	17.12.1.2	The Target SHALL wait for a Valid PDU from the Initiator after it has been activated.
17.12.1.3	When the Initiator sends an Information PDU, the number of payload bytes of that PDU SHALL be less than or equal to the maximum payload size defined by either the LR field in PSL_REQ or, if PSL_REQ has not been used, by the LR field in ATR_RES.	17.12.1.4	When the Target sends an Information PDU, the number of payload bytes of that PDU SHALL be less than or equal to the maximum payload size defined by either the LR field in PSL_REQ or, if PSL_REQ has not been used, by the LR field in ATR_REQ.

17.12.2 Data Exchange Initiator to Target

Data Exchange is the process by which a packet of application data is exchanged between an Initiator and a Target. The application data is carried as the payload of one or more Information PDUs. When the amount of data to be sent exceeds the capacity of a single Information PDU, chaining is used.

Data Exchange from Initiator to Target is achieved by the Initiator sending one or more Information PDUs to the Target. If more than one Information PDU is required to send the data, the Target responds to each chained Information PDU with an ACK PDU. If the Target needs more time to respond to an Information PDU, it can request a response timeout extension using an RTOX PDU. Once a complete data packet has been sent from Initiator to Target, the Target has to send data back to the Initiator

Requirements 284: Data Exchange Initiator to Target

Initiator		Target	
17.12.2.1	After the Initiator sends an unchained Information PDU, the Initiator: • SHALL accept a valid chained or unchained Information PDU and continue as defined for Data Exchange Target to Initiator (see 17.12.3) • SHALL accept a valid RTOX PDU, in accordance with the requirements for Response Timeout Extension (see 17.12.5) • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.2.2	When the Target receives a valid unchained Information PDU, the Target either (one of the following): • SHALL initiate a Response Timeout Extension by responding with an RTOX Request PDU if more time is required (see 17.12.5) • SHALL initiate a Data Exchange Target to Initiator by responding with a chained or unchained Information PDU (see 17.12.3).
17.12.2.3	After the Initiator sends a chained Information PDU, the Initiator: • SHALL accept a valid ACK PDU with a PNI equal to the Initiator's current PNI • MAY accept a valid ACK PDU with a PNI not equal to the Initiator's current PNI • SHALL accept a valid RTOX Request PDU, in accordance with the requirements for Response Time Extension (see 17.12.5) • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.2.4	When the Target receives a valid chained Information PDU, the Target either (one of the following): • SHALL respond with an ACK PDU if it is ready to accept more data • SHALL initiate a Response Timeout Extension by responding an RTOX Request PDU otherwise (see 17.12.5).

Initiator		Target	
17.12.2.5	When the Initiator receives a valid ACK PDU, the Initiator: • SHALL send the next chained or unchained Information PDU if the PNI matched • Otherwise MAY retransmit the last chained Information PDU or MAY treat this as a Semantic Error.	17.12.2.6	After the Target sends an ACK PDU, the Target: • SHALL accept a valid chained or unchained Information PDU • SHALL accept a valid NACK PDU, in accordance with the requirements for Retransmission (see 17.12.7) • SHALL accept a valid ATN PDU, in accordance with the requirements for Presence Check (see 17.12.6) • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.

17.12.3 Data Exchange Target to Initiator

Data Exchange from Target to Initiator is achieved by the Target sending one or more Information PDUs to the Initiator following the completion of Data Exchange from the Initiator to the Target. If more than one Information PDU is required to send the data, the Initiator responds to each chained Information PDU with an ACK PDU. Once a complete data packet has been sent from Target to Initiator, the Initiator has an opportunity to send more data to the Target.

Requirements 285: Data Exchange Target to Initiator

Initiator		Target	
17.12.3.1	When the Initiator receives a valid unchained Information PDU, the Initiator: • MAY initiate a Data Exchange Initiator to Target by sending a chained or unchained Information PDU (see 17.12.2) • MAY initiate a Presence Check by sending an ATN PDU (see 17.12.6).	17.12.3.2	After the Target sends an unchained Information PDU, the Target: • SHALL accept a valid chained or unchained PDU • SHALL accept a valid NACK PDU, in accordance with the requirements for Retransmission (see 17.12.7) • SHALL accept a valid ATN PDU, in accordance with the requirements for Presence Check (see 17.12.6) • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.
17.12.3.3	When the Initiator receives a valid chained Information PDU, the Initiator: • SHALL respond with an ACK PDU.	17.12.3.4	After the Target sends a chained Information PDU, the Target: • SHALL accept a valid ACK PDU • SHALL accept a valid NACK PDU, in accordance with the requirements for Retransmission (see 17.12.7) • SHALL accept a valid ATN PDU, in accordance with the requirements for Presence Check (see 17.12.6) • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.

Initiator		Target	
17.12.3.5	After the Initiator sends an ACK PDU, the Initiator: • SHALL accept a valid chained or unchained Information PDU • SHALL accept a valid RTOX Request PDU, in accordance with the requirements for Response Time Extension (see 17.12.5). • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.3.6	When the Target receives a valid ACK PDU, the Target either (one of the following): • SHALL respond with a chained or unchained Information PDU • SHALL initiate a Response Timeout Extension by responding with an RTOX Request PDU if more time is required (see 17.12.5).

17.12.4 PDU Numbering

Certain types of PDU contain a Packet Number Information (PNI) field which is used for error detection during PDU exchange.

Requirements 286: PDU Numbering

Initiator		Target	
17.12.4.1	The PNI of the Initiator SHALL be initialized to 00b for the current activated Target.	17.12.4.2	The PNI of the Target SHALL be initialized to 00b.
17.12.4.3	When the Initiator receives a valid Information PDU or a valid ACK PDU with a PNI that is equal to the current PNI, the Initiator SHALL increment the current PNI for the current Target before optionally sending a new PDU.	17.12.4.4	When the Target receives a valid Information PDU or a valid ACK PDU with a PNI that is equal to the current PNI of the Target, the Target SHALL send its Response with this PNI and increment the current PNI after.
17.12.4.5	When the Initiator receives a valid Information PDU or a valid ACK PDU with a PNI that is not equal to the Initiator's current PNI, the Initiator MAY treat this as a Semantic Error or MAY retransmit the last Information PDU.	17.12.4.6	When the Target receives a valid Information PDU or an ACK PDU with a PNI that is not equal to the current PNI of the Target, the Target SHALL treat this as a Semantic Error, except when having responded to a valid ATN PDU directly before.
		17.12.4.7	When, after the Target has just responded to a valid ATN PDU, it receives a valid Information PDU or a valid ACK PDU with a PNI that is equal to the current PNI of the Target minus one, the Target SHALL send its Response with the current PNI minus one and SHALL leave the current PNI unchanged afterwards.

17.12.5 Response Time Extension

If a Target requires more time to respond to certain received PDUs than is allowed by the Response Waiting Time, it can use the Response Timeout Extension procedure. This involves responding with an RTOX PDU. If the Initiator chooses to accept the request, it sends back an RTOX PDU with the same multiplier as was requested.

Requirements 287: Response Time Extension

Initiator		Target	
17.12.5.1	When the Initiator receives a valid RTOX Request, the Initiator: • SHALL accept the first RTOX Request PDU after sending an Information PDU or ACK PDU • MAY accept subsequent RTOX Request PDUs. If an RTOX Request PDU is accepted, the Initiator SHALL send an RTOX Response PDU to the Target with the same multiplier as was requested. The Initiator MAY stop accepting subsequent RTOX Request PDUs, thereby preventing blockage of the Initiator by arbitrary long processing time on the Target. Nevertheless, the Initiator SHOULD accept a number of RTOX Request PDUs to allow normal operation to finish on the Target side.	17.12.5.2	After the Target sends an RTOX Request PDU, the Target: • SHALL accept an RTOX Response PDU with the same multiplier as was requested • SHALL accept a valid NACK PDU, in accordance with the requirements for Retransmission (see 17.12.7). • SHALL accept a valid ATN PDU, in accordance with the requirements for Presence Check (see 17.12.6). • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.

Initiator		Target	
17.12.5.3	After the Initiator sends an RTOX Response PDU, the Initiator: • SHALL accept a valid chained or unchained Information PDU if the last Information PDU sent did not indicate chaining. • SHALL accept a valid ACK PDU with a PNI equal to the Initiator's current PNI if the last Information PDU sent indicated chaining. • MAY accept a valid ACK PDU with a PNI not equal to the Initiator's current PNI if the last Information PDU sent indicated chaining. • SHALL accept a valid RTOX Request PDU, in accordance with the requirements for Response Time Extension (see 17.12.5). • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.5.4	When the Target receives a valid RTOX Response, the Target either (one of the following): • SHALL treat it as a Protocol Error if it was not received in response to an RTOX Request PDU from the Target • SHALL initiate another Response Timeout Extension by responding with an RTOX Request PDU if further time is required • SHALL respond with an ACK PDU if the last received Information PDU indicated chaining • SHALL respond with a chained or unchained Information PDU otherwise.

17.12.6 Presence Check

If the Initiator wishes to check that the Target is still in range and available for communication, it can use the Presence Check procedure. This process involves sending an ATN PDU and checking to see whether an ATN PDU is received in response.

Requirements 288: Presence Check

Initiator		Target	
17.12.6.1	After the Initiator sends an ATN PDU, the Initiator: • SHALL accept a valid ATN PDU. • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.6.2	When the Target receives a valid ATN PDU, the Target: • SHALL respond with an identical ATN PDU.
17.12.6.3	When the Initiator receives a valid ATN PDU, the Initiator: • SHALL treat it as a Protocol Error if it is not received in response to an ATN PDU, or • SHALL continue in accordance with the requirements in 17.12.9, if the ATN PDU was sent as part of Error Recovery, or • MAY initiate a Data Exchange Initiator to Target by sending a chained or unchained Information PDU (see 17.12.2), or • MAY initiate another Presence Check by sending an ATN PDU (see 17.12.6).	17.12.6.4	After the Target sends an ATN PDU, the Target: • SHALL accept a valid chained or unchained Information PDU if there is currently no Response Timeout Extension or Data Exchange Target to Initiator ongoing. • SHALL accept a valid NACK PDU, in accordance with the requirements for Retransmission (see 17.12.7). • SHALL accept a valid ATN PDU, in accordance with the requirements for Presence Check (see 17.12.6). • SHALL accept a valid RTOX Response PDU if the last PDU sent before Presence Check is an RTOX Request PDU. • SHALL accept a valid ACK PDU if the last PDU sent before Presence Check is an Information PDU. • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.

17.12.7 Retransmission

If an Initiator needs the Target to retransmit the last PDU sent, for example because a transmission error was detected, it can use the Retransmission procedure. This involves sending a NACK PDU and waiting for the PDU to be retransmitted by the Target.

Requirements 289: Retransmission

Initiator		Target	
17.12.7.1	After the Initiator sends a NACK PDU, the Initiator: • SHALL accept a valid chained or unchained Information PDU if the last Information PDU sent did not indicate chaining. • SHALL accept a valid ACK PDU with a PNI equal to the Initiator's current PNI if the last Information PDU sent indicated chaining. • SHALL accept a valid ATN PDU if a Presence Check is in progress. • MAY accept a valid RTOX Request PDU, in accordance with the requirements for Response Time Extension (see 17.12.5). • SHALL treat all other PDUs (that do not contain Transmission Errors) as Protocol Errors.	17.12.7.2	When the Target receives a valid NACK PDU, • After sending an Information PDU or an ACK PDU, the Target SHALL resend its last PDU if the PNI is equal to the current PNI of the Target minus one. Otherwise, the Target SHALL treat this as a Semantic Error. • After sending an RTOX PDU or an ATN PDU, the Target SHALL respond to the RTOX PDU or ATN PDU. NOTE An RTOX PDU or an ATN PDU is only sent if it was the last PDU sent.
17.12.7.3	When the Initiator receives a valid NACK PDU, the Initiator SHALL treat this as a Protocol Error.	17.12.7.4	The Target SHALL NOT send a NACK PDU.

17.12.8 Error Detection

During the operation of NFC-DEP Protocol, certain errors can occur, and in some cases need to be detected.

Requirements 290: Error Detection

Initiator		Target	
17.12.8.1	The Initiator SHALL detect Transmission Errors, Timeout Errors, and Protocol Errors.	17.12.8.2	The Target SHALL detect Transmission Errors and Protocol Errors.
17.12.8.3	If the Initiator detects a Transmission Error, it SHALL begin Transmission Error Recovery.	17.12.8.4	If the Target detects a Transmission Error, it SHALL NOT send a response, and SHALL remain in receive mode.
17.12.8.5	If the Initiator detects a Timeout Error, it SHALL begin Timeout Error Recovery.		
17.12.8.6	If the Initiator receives a PDU with a Protocol Error, the Initiator SHALL raise the Unrecoverable Protocol Exception.	17.12.8.7	If the Target detects a Protocol Error, it SHALL NOT send a response, and SHALL remain in receive mode.

17.12.9 Error Recovery

In order to increase link robustness, procedures are provided by which an Initiator can attempt to recover from certain types of detected errors. The Target never attempts its own error recovery.

Requirements 291: General Error Recovery Rules

Initiator	
17.12.9.1	When it is entering the Initiator role, the Initiator SHALL allocate an error counter, reset to 0.

Transmission Error Recovery

If the Initiator detects a Transmission Error, the Transmission Error Recovery procedure is used. This involves invoking the Retransmission procedure one or more times.

Requirements 292: Transmission Error Recovery

Initiator	
17.12.9.2	If the Initiator receives a PDU with a Transmission Error, then the Initiator SHALL send a NACK PDU. The Initiator SHALL start to send the NACK PDU within a time $t_{NFCDEP,RETRANSMISSION}$ measured from the end of the frame with the Transmission Error. If the Initiator detects a Transmission Error after it sends a NACK PDU, the Initiator SHALL attempt error recovery by using the Retransmission procedure (see 17.12.7) until a Valid Response is received, or until the Retransmission procedure has been performed a maximum of $n_{NFCDEP,RETRY,NACK}$ times. If no Valid PDU is received for all of these NACK PDUs, then the Initiator SHALL raise the Unrecoverable Transmission Exception. Appendix B.10 lists the values of $t_{NFCDEP,RETRANSMISSION}$ and $n_{NFCDEP,RETRY,NACK}$.

Timeout Error Recovery

If the Initiator detects a Timeout Error, the Timeout Error Recovery procedure is used. This involves invoking the Retransmission or Presence Check procedure one or more times.

Requirements 293: Timeout Error Recovery

Initiator	

17.12.9.3	If the Initiator detects a Timeout Error, the Initiator SHALL continue as follows:
	If the error counter equals $\mathbf{n}_{NFCDEP,TO,MAX}$, the Initiator SHALL raise the Unrecoverable Timeout Exception and SHALL reset the error counter.
	Otherwise the Initiator SHALL increment the error counter and proceed as follows:
	• If the last PDU sent before the Timeout Error occurred was a NACK PDU, the Initiator SHALL attempt error recovery by using the Retransmission procedure (see 17.12.7) until a Valid Response is received, or until the Retransmission procedure has been performed a maximum of $\mathbf{n}_{NFCDEP,RETRY,NACK}$ times. The Initiator SHALL send a NACK PDU before $\mathbf{t}_{NFCDEP,RETRANSMISSION}$.
	  • If no Valid PDU is received after the Retransmission procedure is repeated $\mathbf{n}_{NFCDEP,RETRY,NACK}$ times, the Initiator SHALL raise the Unrecoverable Timeout Exception.
	  • Otherwise the Initiator SHALL reset the error counter and continue processing at the point before the Timeout Error occurred.
	• Otherwise the Initiator SHALL perform a Presence Check procedure (see 17.12.6). If no ATN PDU is received from the Target, the Initiator SHALL attempt error recovery by using the Presence Check procedure until a Valid PDU is received, or until the Presence Check Procedure has been performed a maximum of $\mathbf{n}_{NFCDEP,RETRY,ATN}$ times. The Initiator SHALL start the Presence Check procedure before $\mathbf{t}_{NFCDEP,RETRANSMISSION}$.
	  • If no Valid PDU is received by this point, the Initiator SHALL raise the Unrecoverable Timeout Exception.
	  • Otherwise the Initiator SHALL continue processing at the point before the Timeout Error occurred.
	Appendix B.10 lists the values of $\mathbf{n}_{NFCDEP,TO,MAX}$, $\mathbf{t}_{NFCDEP,RETRANSMISSION}$ and $\mathbf{n}_{NFCDEP,RETRY,ATN}$.

18 Type 5 Tag Platform

18.1 Sequence Format

The Type 5 Tag Platform does not use synchronization mechanism.

Requirements 294: Sequence Format – Type 5 Tag Platform

Poll and Listen Mode	
18.1.1.1	Type 5 Tag Platform Commands and Responses SHALL be sent and received using the NFC-V Technology sequences defined in Section 9.1.

18.2 Bit Level Coding

The Type 5 Tag Platform uses NFC-V bit level coding.

Requirements 295: Type 5 Tag Platform Bit Level Coding

Poll and Listen Mode	
18.2.1.1	Type 5 Tag Platform Commands and Responses SHALL be sent and received using the NFC-V Technology bit level coding defined in Section 9.2.

18.3 Frame Format

The Type 5 Tag Platform uses Commands and Responses in NFC-V Technology Standard Frames, as defined in Section 9.3.1.

When a Type 5 Tag on top of the Type 5 Tag Platform supports the NFC-V Special Frames (defined in Section 9.3.3), the Type 5 Tag Platform also supports the NFC-V Special Frames.

Requirements 296: Type 5 Tag Platform Frame Format

Poll and Listen Mode	
18.3.1.1	Type 5 Tag Platform Commands and Responses SHALL be sent and received using the NFC-V Technology frame format defined in Section 9.3.

18.4 Data and Payload Format

The payload consists of the Commands and Responses described in Section 18.5. The data and payload formats are defined in the following Requirements.

Requirements 297: Type 5 Tag Platform Data and Payload Format

Poll and Listen Mode	
18.4.1.1	Type 5 Tag Platform Commands and Responses SHALL be sent and received using the NFC-V Technology data and payload format defined in Section 9.4.

18.5 Command Set

A Type 5 Tag Platform can be addressed for data exchange directly after its activation. The Type 5 Tag Platform can be activated using the Commands defined in Section 9.5. There are no additional Commands for the Type 5 Tag Platform activation. Type 5 Tag specific Commands are defined in [T5T].

Requirements 298: Type 5 Tag Platform Command Set

Poll and Listen Mode	
18.5.1.1	Type 5 Tag Platform Commands and Responses SHALL be compliant with the requirements defined for half-duplex protocols in Section 10.
18.5.1.2	Type 5 Tag Platform Commands and Responses SHALL be transmitted as the payload of the data and payload format defined in Section 18.4.

18.6 Error Handling

When errors are detected, the following error handling is attempted.

Requirements 299: Type 5 Tag Platform Error Handling

Poll Mode		Listen Mode
18.6.1.1	If a Timeout Error occurs after the transmission of a Read-Alike Command (except the SLPV_REQ Command, as defined in [T5T]) or a Write-Alike Command, the Reader/Writer SHALL attempt error recovery by resending the last Command until a Valid Response is received, or until a maximum of $\mathbf{n}_{T5T,RETRY,ERROR}$ error recovery Commands have been resent without receiving a Valid Response. If for every Command no Valid Response is received, then the Reader/Writer SHALL raise the Unrecoverable Timeout Exception. Appendix B.11 lists the values of $\mathbf{n}_{T5T,RETRY,ERROR}$.	
18.6.1.2	If there is a Transmission Error in a response other than the INVENTORY_RES Response, the Reader/Writer SHALL attempt error recovery by resending the last Command until a maximum of $\mathbf{n}_{T5T,RETRY,ERROR}$ error recovery Commands have been resent without receiving a Valid Response. If for every Command no Valid Response is received, then the Reader/Writer SHALL raise the Unrecoverable Transmission Exception.	

A. Exhibit A

No items have been included in Exhibit A.

B. Values

Symbols are used throughout this specification to identify the values of parameters. The actual values of the parameters are listed in this appendix. For some of the parameters a minimum value and a maximum value are defined. Other parameters are defined by a single value.

Some parameters have separate values for the Poller and for the Listener. Unless otherwise specified, the value for Poll Mode is used when the parameter is referenced in a Poll Mode requirement, and the value for Listen Mode is used when referenced in a Listen Mode requirement.

Requirements 300: Values

Poll Mode		Listen Mode	
B.1.1.1	The actual Poll Mode value of a specific parameter SHALL be set according to the Poll Mode value of the same named parameter specified in this appendix.	B.1.1.1	The actual Listen Mode value of a specific parameter SHALL be set according to the Listen Mode value of the same named parameter specified in this appendix.

B.1 Technology Independent

Table 108: Technology Independent Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$t_{RECOVERY,MAX}$		1280			$1/f_C$

B.2 NFC-A Technology

Table 109: NFC-A Technology Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$FDT_{A,DELTA}$	400		400		ns
$FDT_{A,POLL,MIN}$	6780		1172		$1/f_C$
$FDT_{A,PP,SLP\_REQ,MIN}$	1.1				ms
$FDT_{A,REACTIVATION}$	5.1		5.0		ms
GT_A			5.0		ms
$t_{A,nn}$			1408		$1/f_C$
$n_{A,MIN}$	9				

B.3 NFC-B Technology

Table 110: NFC-B Technology Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$EGT_{B,POLL}$	0	752	0	768	$1/f_C$
$EGT_{B,LISTEN}$	0	272	0	256	$1/f_C$
$FWT_{B,SENSB}$	7680		7296		$1/f_C$
$FWI_{B,MAX}$	14		8		-
ΔFWT_B	49152				$1/f_C$
$\Delta T_{B,POLL}$	16.4				ms
$FDT_{B,REACTIVATION}$	5.1		5.0		ms
GT_B			5.0		ms
$FDT_{B,PP,MIN}$	5.1		5.0		ms
$t_{B,nn}$			1408		$1/f_C$
$TR0_{B,MIN,DEFAULT}$	1008		1024		$1/f_C$
$TR0_{B,MAX}$	6416		4096		$1/f_C$
$TR1_{B,MIN,DEFAULT}$	1264		1280		$1/f_C$
$TR1_{B,MAX}$	3216		3200		$1/f_C$
$TR2_{B,MIN,DEFAULT}$	6780				$1/f_C$
$t_{B,LISTEN,S,1}$	1264	1424	1272	1416	$1/f_C$
$t_{B,LISTEN,S,2}$	240	400	248	392	$1/f_C$
$t_{B,FSOFF}$	0	272	0	256	$1/f_C$
$t_{B,POLL,S,1}$	1280	1416	1272	1424	$1/f_C$
$t_{B,POLL,S,2}$	248	392	240	400	$1/f_C$
$t_{B,POLL,E}$	1280	1416	1272	1424	$1/f_C$
$t_{B,LISTEN,E}$	1264	1424	1272	1416	$1/f_C$
$FSC_{B,MIN}$	16		32		-

B.4 NFC-F Technology

Table 111: NFC-F Technology Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$TR0_{F,LISTEN,MIN}$	(42 × 64) - 16		42 × 64		$1/f_C$
$TR0_{F,POLL,MIN}$	(106 × 64) + 16		106 × 64		$1/f_C$
$TR0_{F,PP,MIN}$	(106 × 64) + 16				$1/f_C$
$TR1_F$	48		48		bd
$FDT_{F,REACTIVATION}$	20.4		20.0		ms
GT_F			20.0		ms
$\Delta T_{F,POLL}$	1				ms
$T_{F,TIMESLOT}$	256 × 64		256 × 64		$1/f_C$
$T_{F,DELAY}$	512 × 64		512 × 64		$1/f_C$

B.5 NFC-V Technology

Table 112: NFC-V Technology Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$FDT_{V,POLL}$	4202		4192		$1/f_C$
$FDT_{V,INVENT\_NORES}$	4394 + 2048		4384 + 2048		$1/f_C$
$FDT_{V,LISTEN,MIN}$	4310		4320		$1/f_C$
$FDT_{V,LISTEN,MAX1}$	4394		4384		$1/f_C$
$FDT_{V,LISTEN,MAX2}$	270644		270634		$1/f_C$
$FDT_{V,EOF}$	10	20	9.9	20.1	ms
GT_V			5.0		ms
$\Delta T_{V,POLL}$	50				ms

NOTE The definitions of $\mathbf{FDT}_{V,LISTEN,MIN}$, $\mathbf{FDT}_{V,LISTEN,MAX1}$ and $\mathbf{FDT}_{V,LISTEN,MAX2}$ contain a tolerance of $10/\mathbf{f}_C$ for Poll Mode Value, compared to the definition in [ISO/IEC_15693].

B.6 Type 2 Tag Platform

Table 113: Type 2 Tag Platform Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$\mathbf{n}_{T2T,RETRY,ERROR}$	1	2			

B.7 Type 4A Tag Platform

Table 114: Type 4A Tag Platform Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$\mathbf{FSD}_{T4AT,MIN}$	256		256		-
$\mathbf{FSC}_{T4AT,MIN}$	16		32		-
$\mathbf{FWI}_{T4AT,MAX}$	14		8		-
$\mathbf{FWT}_{T4AT,ACTIVATION}$	71680		65536		$1/\mathbf{f}_C$
$\Delta\mathbf{FWT}_{T4AT}$	49152				$1/\mathbf{f}_C$
$\Delta\mathbf{T}_{T4AT,POLL}$	16.4				ms
$\mathbf{SFGI}_{T4AT,MAX}$	14		8		-
$\mathbf{n}_{T4AT,RETRY,RATS}$	0	1			

B.8 Type 4B Tag Platform

Table 115: Type 4B Tag Platform Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$FSD_{T4BT,MIN}$	256		256		-
$FWT_{T4BT,ACTIVATION}$	71680		65536		$1/f_C$
$SFGI_{T4BT,MAX}$	14		8		
$n_{T4BT,RETRY,ATTRIB}$	0	1			
$k_{T4BT,MAX,ATTRIB}$	15		15		
$n_{T4BT,MAX,ATTRIB}$	15		15		

B.9 ISO-DEP Protocol

Table 116: ISO-DEP Protocol Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$t_{ISODEP,RETRANSMISSION}$	0	4096×2^{12}			$1/f_C$
$FWT_{ISODEP,DEACTIVATION}$	71680		65536		$1/f_C$
$n_{ISODEP,RETRY,ACK}$	2	5			
$n_{ISODEP,RETRY,NAK}$	2	5			
$n_{ISODEP,RETRY,WTX}$	2	5			
$n_{ISODEP,RETRY,DESELECT}$	0	5			

B.10 NFC-DEP Protocol

Table 117: NFC-DEP Protocol Poll Mode and Listen Mode Parameter Values

Parameter	Initiator Value		Target Value		Units
	Min	Max	Min	Max	
$WT_{NFCDEP,MAX}$	14		14		-
ΔRWT_{NFCDEP}	16				$1/f_C$
$\Delta T_{NFCDEP,INITIATOR}$	100				ms
$RWT_{NFCDEP,ACTIVATION}$	4096×2^{12}		4096×2^{12}		$1/f_C$
$RWT_{ACM,NFCDEP,ACTIVATION}$	4096×2^{9}		4096×2^{9}		$1/f_C$
$t_{NFCDEP,RETRANSMISSION}$	0	4096×2^{12}			$1/f_C$
$n_{NFCDEP,RETRY,NACK}$	1	4			
$n_{NFCDEP,RETRY,ATN}$	1	4			
$n_{NFCDEP,TO,MAX}$	2	5			

B.11 Type 5 Tag Platform

Table 118: Type 5 Tag Platform Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value		Listen Mode Value		Units
	Min	Max	Min	Max	
$n_{T5T,RETRY,ERROR}$	2	5			

C. Revision History

Table 119 outlines the revision history of the Digital Protocol Technical Specification.

Table 119: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Digital Protocol	Version 1 Nov 2010	Final		
Digital Protocol	Version 2.0 May 2017	Final	Numerous updates, bug fixes and clarifications, including terminology alignment and general edits.	Version 1.0 Nov 2010
Digital Protocol	Version 2.1 Apr 2019	Final	Support for RF frames at ISO-DEP up to 4KB. Some clarifications. Editorial updates.	Version 2.0 May 2017
Digital Technical Specification	Version 2.2 December 2019	Final	Adds error recovery for T2T and T5T platform plus some minor editorial changes.	Version 2.1 April 2019
Digital Protocol Technical Specification	Version 2.2 January 2020	Final	Editorial change to copyright notice.	Version 2.2 December 2019
Digital Protocol Technical Specification	Version 2.3 July 2021	Final	T1T Platform removal. Clarifications for: T2T Error handling, ACM timing, and NFC-F De-Synchronization. Editorial update.	Version 2.2 January 2020