



Activity

Technical Specification

Version 2.3

2023-02-03

[ACTIVITY]

NFC Forum™

Copyright © 2017-2023 NFC Forum

Contents

1	Introduction.....	1
1.1	Objectives	1
1.2	Applicable Documents or References	1
1.3	Administration	1
1.4	Name and Logo Usage	2
1.5	Intellectual Property	2
1.6	Special Word Usage.....	2
1.7	Requirement Numbering	2
1.8	Notational Conventions.....	5
1.8.1	Notations	5
1.8.2	Figures.....	6
1.9	Abbreviations	7
1.10	Glossary.....	10
2	Purpose.....	15
3	RF Collision Avoidance.....	17
3.1	Generic requirements for Collision Avoidance	17
3.2	Active Communication Mode Timing	17
3.3	Initial RF Collision Avoidance	20
3.4	Subsequent RF Collision Avoidance	21
4	Listen Mode – Generic Requirements	23
5	Listen Mode – Configuration	24
6	Listen Mode – State Machine.....	26
6.1	NO_REMOTE_FIELD State.....	26
6.2	IDLE State.....	27
6.3	READY_A State and READY_A* State	30
6.4	READY_A' State and READY_A'' State	31
6.5	READY_A'' State and READY_A''' State.....	32
6.6	ACTIVE_A State and ACTIVE_A* State	32
6.7	SLEEP_A State.....	33
6.8	ATR_READY_A State	33
6.9	TARGET_A State.....	34
6.10	CARD_EMULATOR_4A State	34
6.11	READY_B_REQU State.....	35
6.12	READY_B_DECL State	36
6.13	SLEEP_B State.....	37
6.14	CARD_EMULATOR_4B State	37
6.15	READY_F State	38
6.16	ATR_READY_F State.....	39
6.17	TARGET_F State	39
6.18	CARD_EMULATOR_3 State	40
6.19	SLEEP_AF State	41
7	Poll Mode – Generic Requirements	42
8	Poll Mode – Activity Model	44
9	Poll Mode – Activities.....	45

9.1	Activities – Requirements	46
9.2	Technology Detection Activity.....	46
9.2.1	Preconditions	47
9.2.2	Post-conditions	48
9.2.3	Flow Chart and Requirements	49
9.3	Collision Resolution Activity	57
9.3.1	Preconditions	57
9.3.2	Post-conditions	59
9.3.3	Flow Chart (Normative)	61
9.3.4	Flow Chart and Requirements for NFC-A	62
9.3.5	Flow Chart and Requirements for NFC-B.....	67
9.3.6	Flow Chart and Requirements for NFC-F.....	73
9.3.7	Flow Chart and Requirements for NFC-V	76
9.3.8	Flow Chart and Requirements for NFC-ACM	81
9.4	Device Activation Activity.....	82
9.4.1	Preconditions	82
9.4.2	Post-conditions	85
9.4.3	Flow Chart (Normative).....	86
9.4.4	Flow Chart and Requirements for NFC-A	87
9.4.5	Flow Chart and Requirements for NFC-B.....	92
9.4.6	Flow Chart and Requirements for NFC-F	93
9.4.7	Flow Chart and Requirements for NFC-V	97
9.4.8	Flow Chart and Requirements for NFC-ACM	97
9.5	Data Exchange Activity	99
9.5.1	Preconditions	99
9.5.2	Post-conditions	100
9.5.3	Flow Chart (Normative).....	100
9.5.4	Flow Chart and Requirements for NFC-DEP	101
9.5.5	Flow Chart and Requirements for ISO-DEP	102
9.5.6	Flow Chart and Requirements for Type 2 Tag Platform.....	103
9.5.7	Flow Chart and Requirements for Type 3 Tag Platform	104
9.5.8	Flow Chart and Requirements for Type 5 Tag Platform.....	105
9.6	Device Deactivation Activity	106
9.6.1	Preconditions	106
9.6.2	Post-conditions	107
9.6.3	Flow Chart (Normative).....	108
9.6.4	Flow Chart and Requirements for NFC-DEP	109
9.6.5	Flow Chart and Requirements for ISO-DEP	111
9.6.6	Flow Chart and Requirements for Type 2 Tag Platform.....	112
9.6.7	Flow Chart and Requirements for Type 3 Tag Platform	112
9.6.8	Flow Chart and Requirements for Type 5 Tag Platform	112
A.	Exhibit A.....	113
B.	Listen Mode – State Diagram (Informative).....	114
C.	Values.....	115
D.	Revision History	117

Figures

Figure 1: Example Flow Chart.....	3
Figure 2: Communication Flow illustration for Active Communication Mode.....	18
Figure 3: Initial RF Collision Avoidance – Flow Chart	20
Figure 4: Subsequent RF Collision Avoidance – Flow Chart.....	21
Figure 5: Activity	44
Figure 6: Technology Detection Activity – Flow Chart.....	50
Figure 7: Collision Resolution Activity (Sheet 1, Entry) – Normative Flow Chart.....	61
Figure 8: Collision Resolution Activity (Sheet 2, connector A, NFC-A) – Flow Chart	64
Figure 9: Collision Resolution Activity (Sheet 3, connector B, NFC-B) – Flow Chart.....	69
Figure 10: Collision Resolution Activity (Sheet 4, connector F, NFC-F) – Flow Chart.....	74
Figure 11: Collision Resolution Activity (Sheet 5, connector V, NFC-V) – Flow Chart.....	77
Figure 12: Collision Resolution Activity (Sheet 6, connector ACM, NFC-ACM) – Flow Chart..	81
Figure 13: Device Activation Activity (Sheet 1, Entry) – Normative Flow Chart.....	86
Figure 14: Device Activation Activity (Sheet 2, Connector DA_1, NFC-DEP (NFC-A), Type 2 & 4A Tag Platform) – Flow Chart	88
Figure 15: Device Activation Activity (Sheet 3, Connector DA_2, Type 4B Tag Platform) – Flow Chart.....	92
Figure 16: Device Activation Activity (Sheet 4, Connector DA_3, NFC-DEP (NFC-F), Type 3 Tag Platform) – Flow Chart	94
Figure 17: Device Activation Activity (Sheet 5, Connector DA_4, Type 5 Tag Platform – Flow Chart).....	97
Figure 18: Device Activation Activity (Sheet 6, Connector DA_5, NFC-DEP (NFC-ACM)) – Flow Chart	97
Figure 19: Data Exchange Activity (Sheet 1, entry) – Normative Flow Chart	100
Figure 20: Data Exchange Activity (Sheet 2, connector DE_1, NFC-DEP using Passive or Active Communication Mode) – Flow Chart	101
Figure 21: Data Exchange Activity (Sheet 3, Connector DE_2, ISO-DEP) – Flow Chart.....	102
Figure 22: Data Exchange Activity (Sheet 5, connector DE_4, Type 2 Tag Platform) – Flow Chart.....	103
Figure 23: Data Exchange Activity (Sheet 6, connector DE_5, Type 3 Tag Platform) – Flow Chart.....	104
Figure 24: Data Exchange Activity (Sheet 5, connector DE_6, Type 5 Tag Platform) – Flow Chart.....	105
Figure 25: Device Deactivation Activity (Sheet 1, Entry) – Normative Flow Chart	108
Figure 26: Device Deactivation Activity (Sheet 2, Connector DD_1, NFC-DEP) – Flow Chart	110
Figure 27: Device Deactivation Activity (Sheet 3, connector DD_2, ISO-DEP) – Flow Chart..	111
Activity	

Figure 28: Device Deactivation Activity (Sheet 4, connector DD_3, Type 2 Tag Platform) – Flow Chart.....	112
Figure 29: Listen Mode – State Diagram (Informative).....	114

Tables

Table 1: Sample Requirement.....	2
Table 2: Example Requirements	4
Table 3: Notational Conventions	5
Table 4: Figure Notation	6
Table 5: Abbreviations	7
Table 6: General Timing Definition for Field Handling – NFC-ACM	18
Table 7: Listen Mode – Configuration Parameters	24
Table 8: Technology Detection Activity – Configuration Parameters	47
Table 9: Technology Detection Activity – Output Parameters	48
Table 10: Technology Detection Activity – Output into Greedy Collection.....	49
Table 11: Collision Resolution Activity – Configuration Parameters.....	57
Table 12: Collision Resolution Activity – Input Parameters	58
Table 13: Collision Resolution Activity – Input from Greedy Collection.....	59
Table 14: Collision Resolution Activity – Output Parameters	59
Table 15: Collision Resolution Activity – Output into Greedy Collection	60
Table 16: Device Activation Activity – Configuration Parameters	82
Table 17: Device Activation Activity – Input Parameters.....	83
Table 18: Device Activation Activity – Input from Greedy Collection	84
Table 19: Device Activation Activity – Output Parameters	85
Table 20: Device Activation Activity – Output into Greedy Collection	85
Table 21: Data Exchange Activity – Input Parameters	99
Table 22: Device Activation Activity – Input from Greedy Collection	99
Table 23: Data Exchange Activity – Output Parameters.....	100
Table 24: Device Deactivation Activity – Input Parameters	106
Table 25: Device Deactivation Activity – Output Parameters.....	107
Table 26: Poll Mode and Listen Mode Parameter Values	116
Table 27: Revision History.....	117

b

Requirements

Requirements 1: RF Collision Avoidance	17
Requirements 2: Requirements for Operating and Remote Field Handling	19
Requirements 3: Initial RF Collision Avoidance	20
Requirements 4: Subsequent RF Collision Avoidance.....	22
Requirements 5: Listen Mode – Generic	23
Requirements 6: Listen Mode – NO_REMOTE_FIELD State.....	26
Requirements 7: Listen Mode – IDLE State.....	27
Requirements 8: Listen Mode – READY_A State and READY_A* State.....	30
Requirements 9: Listen Mode – READY_A' State and READY_A'* State	31
Requirements 10: Listen Mode – READY_A'' State and READY_A''* State.....	32
Requirements 11: Listen Mode – ACTIVE_A State and ACTIVE_A* State	32
Requirements 12: Listen Mode – SLEEP_A State.....	33
Requirements 13: Listen Mode – ATR_READY_A State	33
Requirements 14: Listen Mode – TARGET_A State.....	34
Requirements 15: Listen Mode – CARD_EMULATOR_4A State	34
Requirements 16: Listen Mode – READY_B_QUER State.....	35
Requirements 17: Listen Mode – READY_B_DECL State	36
Requirements 18: Listen Mode – SLEEP_B State.....	37
Requirements 19: Listen Mode – CARD_EMULATOR_4B State	37
Requirements 20: Listen Mode – READY_F State	38
Requirements 21: Listen Mode – ATR_READY_F State.....	39
Requirements 22: Listen Mode – TARGET_F State	39
Requirements 23: Listen Mode – CARD_EMULATOR_3 State.....	40
Requirements 24: Listen Mode – SLEEP_AF State	41
Requirements 25: Generic	42
Requirements 26: Activities – General.....	46
Requirements 27: Technology Detection Activity.....	51
Requirements 28: Collision Resolution Activity	62
Requirements 29: Collision Resolution Activity – NFC-A	65
Requirements 30: Collision Resolution Activity – NFC-B	70
Requirements 31: Collision Resolution Activity – NFC-F.....	75
Requirements 32: Collision Resolution Activity – NFC-V	78

Requirements 33: Collision Resolution Activity – NFC-ACM	81
Requirements 34: Device Activation Activity	87
Requirements 35: Device Activation Activity – NFC-DEP (NFC-A), Type 2, & 4A Tag Platform	89
Requirements 36: Device Activation Activity – Type 4B Tag Platform.....	93
Requirements 37: Device Activation Activity – NFC-DEP (NFC-F), Type 3 Tag Platform.....	95
Requirements 38: Device Activation Activity – Type 5 Tag Platform	97
Requirements 39: Device Activation Activity – NFC-ACM.....	98
Requirements 40: Data Exchange Activity.....	101
Requirements 41: Data Exchange Activity – NFC-DEP.....	102
Requirements 42: Data Exchange Activity – ISO-DEP	102
Requirements 43: Data Exchange Activity – Type 2 Tag Platform.....	103
Requirements 44: Data Exchange Activity – Type 3 Tag Platform.....	104
Requirements 45: Data Exchange Activity – Type 5 Tag Platform.....	105
Requirements 46: Device Deactivation Activity.....	109
Requirements 47: Device Deactivation Activity – NFC-DEP	110
Requirements 48: Device Deactivation Activity – ISO-DEP	111
Requirements 49: Device Deactivation Activity – Type 2 Tag Platform.....	112

1 Introduction

1.1 Objectives

This document describes how the NFC Digital Protocol Specification can be used to set up a communication protocol with another device.

For Poll Mode this document describes the building blocks, called Activities, for setting up the communication protocol.

These Activities can be used as defined in this specification or can be modified to define other ways of setting up the communication protocol, covering the same or different use cases. For Listen Mode this document defines a state machine that includes state transitions and Responses caused by Commands specified in [DIGITAL] and [T3T].

The state machine can be configured to enable or disable specific Listen Mode features such as P2P Target or a specific Card Emulator.

Activities can be combined in Poll Profiles. Each Poll Profile has specific Configuration Parameters and covers a particular use case.

In Listen Mode an NFC Forum device can use Listen Profiles, which contain all necessary configuration parameter values to enable a Listen Mode feature.

Specific Poll Profiles and Listen Profiles are defined in the [PROFILES] specification.

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.

[ANALOG]	Analog Technical Specification, NFC Forum
[DIGITAL]	Digital Protocol Technical Specification, NFC Forum
[JIS_X_6319-4]	JIS X 6319-4, Specification of implementation for integrated circuit(s) cards – Part 4: High speed proximity cards, JIS
[PROFILES]	Profiles Technical Specification, NFC Forum
[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 Activity 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 NFC Forum Activity Technical Specification conforms to the Intellectual Property guidelines specified in the NFC Forum *Intellectual Property Rights Policy*, as outlined in the NFC Forum *Rules of Procedure*. These documents are available on the [NFC Forum website](http://www.nfc-forum.org/).

1.6 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].

1.7 Requirement Numbering

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

Table 1: Sample Requirement

1.7.1.1	A car SHALL have four wheels.
----------------	-------------------------------

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

A figure that is labeled “flow chart” illustrates the behavior given by the corresponding requirements tables. Figures are informative if not otherwise stated. An example is shown in Figure 1.

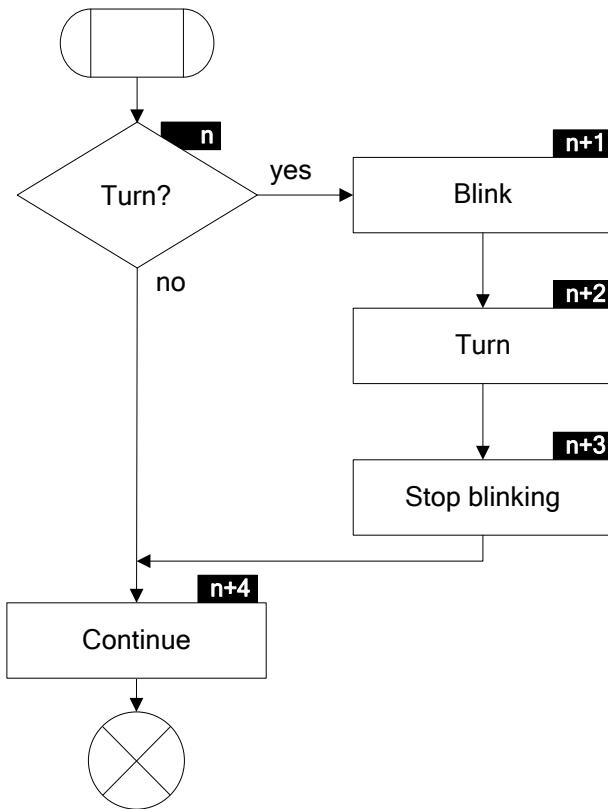


Figure 1: Example Flow Chart

A requirement can be labeled as a symbol, when referring to a flow chart, indicating a particular sequence. If the current requirement is labeled “Symbol n”, then the next requirement in the sequence is “Symbol n+1”, unless explicitly stated differently.

Table 2: Example Requirements

1.7.1.2	Symbol n Before turning to the left or right, a car SHALL proceed to Symbol $n+1$. Otherwise the car SHALL proceed to Symbol $n+4$.
1.7.1.3	Symbol $n+1$ The car SHALL start its left or right (appropriate to the direction) blinker.
1.7.1.4	Symbol $n+2$ The car SHALL turn.
1.7.1.5	Symbol $n+3$ The car SHALL stop its blinker.
1.7.1.6	Symbol $n+4$ The car SHALL continue to drive straight ahead or stop.

1.8 Notational Conventions

1.8.1 Notations

The notational conventions defined in Table 3 apply to this document.

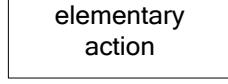
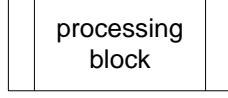
Table 3: Notational Conventions

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 trailing 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).

1.8.2 Figures

Table 4 defines the graphical notation used in the figures of this document.

Table 4: Figure Notation

Symbol	Meaning
	Activity
	Start of a flow chart
	Connection point with dedicated label as used when a flow chart is split into multiple figures
	End of a flow chart
	Test block with one input branch and several output branches
	Elementary action block
	Processing block that can be decomposed in elementary action blocks and/or other processing blocks
	Connecting element with processing flow indicated by the direction of the arrow

1.9 Abbreviations

Table 5 contains the definitions of the abbreviations and acronyms used in this specification.

Table 5: Abbreviations

Abbreviation	Description
AFI	Application Family Identifier
ALL_REQ	ALL NFC-A REQuest
ALLB_REQ (AFI, N1)	ALL NFC-B REQuest with matching AFI and N equal to 1
ALLB_REQ (AFI, N>)	ALL NFC-B REQuest with matching AFI and N greater than 1 and if R is greater than 1
ALLB_REQ (nAFI)	ALL NFC-B REQuest with not matching AFI
ANTICOLL	ANTICOLLision
BITR	BIT Rate
CLn	Cascade Level n ($1 \leq n \leq 3$)
CMD	CoMmanD
CUP	CHECK Command or UPDATE Command or Proprietary Command for the Type 3Tag Platform.
COLL	COLLision
DA	Device Activation
DD	Device Deactivation
DE	Data Exchange
DECL	DECLared
DEP_REQ	Data Exchange Protocol REQuest
DEP_RES	Data Exchange Protocol Response
DRI	Data rate Received by Initiator
DSI	Data rate Send by Initiator
DSL	DeSeLect
f _C	Carrier Frequency
FDT	Frame Delay Time
FWT	Frame Waiting Time
GB	General Bytes
GT	Guard Time
ID	IDentifier
ISO	International Organization for Standardization

Abbreviation	Description
LLCP	Logical Link Control Protocol
Max	Maximum
Min	Minimum
ms	millisecond
n.a.	not applicable
N	Number of slots
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NFC-A	Near Field Communication – Type A Technology
NFC-ACM	Near Field Communication – Active Communication Mode (based on either NFC-A or NFC-F)
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. NFCID0 is always 4 bytes long.
NFCID1	NFC-A identifier. 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 NFCID2 is always 8 bytes long.
NFCID3	NFC-DEP identifier NFCID3 is always 10 bytes long.
P2P	Peer-to-Peer
PEND	PENDing
PDU	Protocol Data Unit
PSL_REQ (A)	Parameter SeLECTION REQuest with DS1 indicating NFC-A
PSL_REQ (F)	Parameter SeLECTION REQuest with DS1 indicating NFC-F
PTGT	Proprietary Technology Guard Time
R	Randomly chosen slot number, NFC-B
RATS	Request for Answer to Select
RD	Request Data

Abbreviation	Description
REQU	REQuested
RF	Radio Frequency
RLS	ReLeaSe
SC	System Code, NFC-F
SDD	Single Device Detection
SEL	SElection
SENSB_REQ (AFI, N1)	SENS NFC-B REQuest with matching AFI and N equal to 1
SENSB_REQ (AFI, N>)	SENS NFC-B REQuest with matching AFI and N greater than 1 and if R is greater than 1
SENSB_REQ (nAFI)	SENS NFC-B REQuest with not matching AFI
TECH	TECHnology
T _{ID}	Initial Delay Time
TRFW	Radio Frequency (RF) Waiting Time
UID	NFC-V Unique Identifier

1.10 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 in an NFC Forum Device.

Bail-out Option

A configuration option that allows the NFC Forum Device to conclude the Technology Detection Activity, if the respective Bail-out parameter is set.

Byte Sequence

Concatenation of hexadecimal values.

Card Emulator

Role of an NFC Forum Device, reached when an NFC Forum Device in Listen Mode has gone through a number of States. In this mode the NFC Forum Device behaves as one of the Technology Subsets.

Collision

For NFC-A a collision is a superposition of a ‘0’ and a ‘1’ (as defined in [DIGITAL]).

For NFC-B, NFC-F, and NFC-V a collision is a superposition of multiple Responses, resulting in a Transmission Error.

Command

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

Configuration Parameters

Parameters that influence the execution of an Activity and which do not depend on the output of a previous Activity.

Correct Frame

A frame without Transmission Error.

Greedy Collection

Temporary storage for information collected as part of the Activity and used during processing.

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

The half-duplex block transmission protocol defined in [DIGITAL].

Listen Mode

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

Listen Profile

A set of Listen Mode configuration parameter values and device response rules that together enable the NFC Forum Device in Listen Mode to operate as either a P2P Target or as a Card Emulator for one of the defined Tag Platforms.

Listener

An NFC Forum Device in Listen Mode.

NFC-DEP Protocol

The half-duplex block transmission protocol defined in [DIGITAL].

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 Forum Tag

A contactless tag or (smart) card supporting NDEF.

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.

No Remote Field Sensed

A condition that indicates the absence of the Remote Field. See the definition of **V_{ov}**, **RESET** in [ANALOG].

Operating Field

The radio frequency field created by the NFC Forum Device.

Operating Field Off

A condition of the Operating Field when the field strength is below a well-defined threshold.

Operating Field On

A condition of the Operating Field when the field strength is above a well-defined threshold for a minimum period of time.

OTHER

A Protocol Error or Transmission Error.

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 for Listeners:

- 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 Mode

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

Poll Profile

The combination of a Resolution Process managing a set of Activities, an Initialization that chooses a set of values as Configuration Parameters, and Clean-up.

Poller

An NFC Forum Device in Poll Mode.

Profile

A set of definitions that determine the behavior of the NFC Forum Device in either Poll Mode (Poll Profile) or Listen Mode (Listen Profile).

Proprietary Command

Any Command from one of the NFC technologies of which the meaning is outside of the scope of this specification.

Proprietary Technology

Any technology of which the Command(s) used in the Technology Detection Activity do(es) NOT move the NFC Forum Device (in Listen Mode) out of the **IDLE** State.

Protocol Error

A Semantic Error or Syntax Error.

Reader/Writer

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

Remote Field

The radio frequency field generated by a remote device and sensed by the NFC Forum Device.

Remote Field Off

A condition in which the Remote Field is below a certain threshold as defined in [ANALOG].

Remote Field On

A condition of the Remote Field being stable and strong enough to put the NFC Forum Device in a state that it can operate. Defined in [ANALOG].

Resolution Process

The part of the adjacent upper layer managing the Activities. The Resolution Process decides the next Activity to perform and hands over the Parameters needed.

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 with 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 in the frame is not consistent with [DIGITAL].

Target

Role of a Listener when it has gone through a number of States. In this mode the NFC Forum Device 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 five 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. See [DIGITAL].

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 [DIGITAL].

Unmodulated Carrier

A condition of the Operating Field with no modulation present. Defined in [ANALOG].

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.

2 Purpose

The Activity Technical Specification describes a communications layer that is complementary to the Digital Technical Specification.

This specification lists the requirements of the behavior of an NFC Forum Device as it can be observed from monitoring the radio frequency field. The specification is intended to be read as such, focusing on the external behavior, even if the description can be interpreted as a software implementation specification. Any implementation that creates the same external behavior as specified—and that is therefore indistinguishable from a testing point of view—meets the requirements.

Section 3 describes the process to prevent two NFC Forum Devices that are in proximity from both generating an Operating Field.

The following sections separately describe Listen Mode and Poll Mode.

Listen Mode is described in sections 4 to 6:

- Generic requirements (Section 4)

These requirements need to be observed to ensure interoperability between Listeners and Pollers, and between Listeners and existing contactless infrastructure, independent of the implementation in the Listener.

- Configuration (Section 5)

This section defines the Configuration Parameters that are available to configure the Listen Mode state machine.

- State machine (Section 6)

This section contains the state machine with a detailed description of all the States.

Poll Mode is described in Sections 7 and 8:

- Generic requirements (Section 7)

These requirements need to be observed to ensure interoperability between Pollers and Listeners, and between Pollers and existing contactless infrastructure, independent of the implementation in the Poller.

- Activity Model (Section 8)

This section describes the model used to represent functional blocks, called Activities.

- Activities (Section 9)

This section describes process flows and Configuration Parameters for the following building blocks:

- Technology detection: detects whether there is another device to communicate with and, if so, what technologies it supports
- Collision resolution: detects the presence of multiple devices and enumerates the different identifiers
- Device activation: activates a particular device to establish communication
- Data exchange: exchange of application data
- Device deactivation: deactivates this device to end communication and to enable possible activation of another device.

Each flow or combination of flows is similar to a set of library functions that a developer can call. Even though it is not the intention of this specification to define or prescribe application programming interfaces (APIs), the facilities defined in this specification can be used in specific APIs. API developers have the choice of using the process flows and variables defined in this specification or developing their own.

3 RF Collision Avoidance

Before generating an Operating Field, an NFC Forum Device performs Radio Frequency (RF) Collision Avoidance. Two types of RF Collision Avoidance are defined:

- Initial RF Collision Avoidance is performed by a Poller prior to turning on the Operating Field, in the following cases:
 - After entering Poll Mode
 - Remaining in Poll Mode after a field reset.
- Subsequent RF Collision avoidance is performed by Pollers and Listeners when using Active Communication Mode prior to turning on an Operating Field. A Poller performs Subsequent RF Collision Avoidance starting with the second Command. A Listener always performs Subsequent RF Collision Avoidance before sending a Response.

3.1 Generic requirements for Collision Avoidance

The following requirements apply for RF Collision Avoidance.

Requirements 1: RF Collision Avoidance

Poll and Listen Mode

- 3.1.1.1** A Poller SHALL perform Initial RF Collision Avoidance prior to turning on an Operating Field in the following cases:
 - After entering Poll Mode
 - Remaining in Poll Mode after No Remote Field Sensed.
 - 3.1.1.2** In Active Communication Mode a Poller acting as Initiator SHALL perform Subsequent RF Collision Avoidance every time prior to sending a Command other than the ATR_REQ Command.
 - 3.1.1.3** In Active Communication Mode a Listener acting as a Target SHALL perform Subsequent RF Collision Avoidance every time prior to sending a Response.
 - 3.1.1.4** The Poller or Listener SHALL start Subsequent RF Collision Avoidance when the remote Operating Field changes from Operating Field On to Operating Field Off.
-

3.2 Active Communication Mode Timing

This section introduces general timing requirements to enable a robust communication in Active Communication Mode.

Figure 2 illustrates the communication flow between Initiator and Target, including all necessary timing relationships.

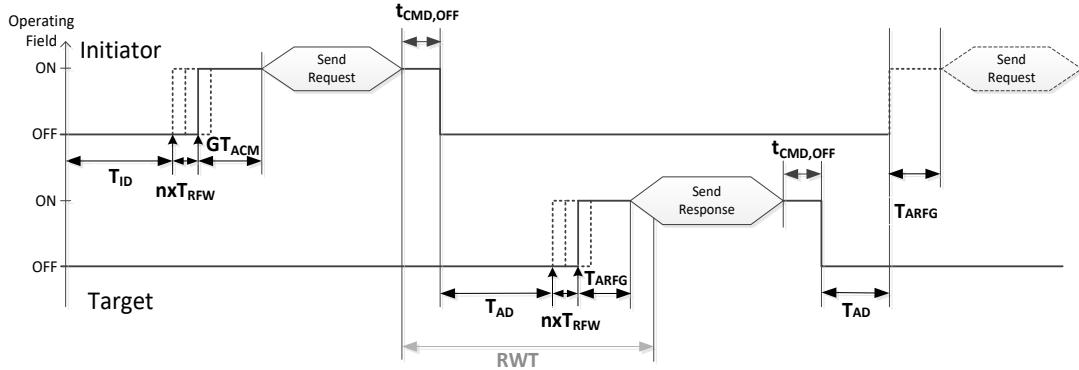


Figure 2: Communication Flow illustration for Active Communication Mode

The timing parameters used by the Initiator and the Target for Active Communication Mode are defined in Table 6.

Table 6: General Timing Definition for Field Handling – NFC-ACM

$t_{CMD,OFF}$	$t_{CMD,OFF}$ is defined as the time between the start of the rising edge of the last modulation and the start of the falling edge when the Peer turns off the Operating Field. Depending on the Technology, $t_{CMD,OFF}$ has different values: $t_{A,CMD,OFF}$, which is applied if NFC-A is the currently used Technology and $t_{F,CMD,OFF}$, which is used if NFC-F is the currently used Technology.
$T_{AD} + n \times T_{RFW}$	$(T_{AD} + n \times T_{RFW})$ is defined as the time between the end of $t_{FIELD,DEACT}$ and the start of the $t_{FIELD,ACT}$. See [ANALOG] for the definitions of $t_{FIELD,ACT}$ and $t_{FIELD,DEACT}$.
GT_{ACM}	GT_{ACM} is defined as the time between the end of the rising edge (V_4 defined by [ANALOG]) and the start of the first modulation.
T_{ARFG}	T_{ARFG} is defined as the time between the end of the rising edge (V_4 defined by [ANALOG]) and the start of the first modulation.

Requirements 2: Requirements for Operating and Remote Field Handling

Initiator	Target
<p>3.2.1.1 The Initiator SHALL treat each of the following as an error within an action block (see Section 8):</p> <ul style="list-style-type: none"> • If Remote Field Off is sensed for a time longer than $T_{NRF,MAX}$ • If a switch from Remote Field On to Remote Field Off is detected and no frame has been received during the time the field was on. <p>Appendix C lists the values of $T_{NRF,MAX}$.</p>	<p>3.2.1.2 The Target SHALL enter the NO_REMOTE_FIELD State (see Section 6) within a delay not greater than t_{FIELD_OFF}:</p> <ul style="list-style-type: none"> • If Remote Field Off is sensed for a time longer than $T_{NRF,MAX}$ • If a switch from Remote Field On to Remote Field Off is detected and no frame has been received during the time the field was on. <p>Appendix C lists the values of $T_{NRF,MAX}$ and t_{FIELD_OFF}.</p>
<p>3.2.1.3 After sending a frame, the Initiator SHALL turn off the Operating Field (as defined in [ANALOG]) within $t_{CMD,OFF}$ after the end of the frame.</p> <p>The following values SHALL be used for $t_{CMD,OFF}$:</p> <ul style="list-style-type: none"> • $t_{A,CMD,OFF}$: if NFC-A is the currently used Technology • $t_{F,CMD,OFF}$: if NFC-F is the currently used Technology. <p>Appendix C lists the values of $t_{A,CMD,OFF}$ and $t_{F,CMD,OFF}$.</p>	<p>3.2.1.4 After sending a frame, the Target SHALL turn off the Operating Field (as defined in [ANALOG]) within $t_{CMD,OFF}$ after the end of the frame.</p> <p>The following values SHALL be used for $t_{CMD,OFF}$:</p> <ul style="list-style-type: none"> • $t_{A,CMD,OFF}$: if NFC-A is the currently used Technology • $t_{F,CMD,OFF}$: if NFC-F is the currently used Technology. <p>Appendix C lists the values of $t_{A,CMD,OFF}$ and $t_{F,CMD,OFF}$.</p>
<p>3.2.1.5 If the Remote Field Off is not sensed within $t_{CMD,OFF,MAX} + T_{RFW}$ after the end of the frame, the Initiator SHALL treat this as an error within an action block (see Section 8).</p> <p>Appendix C lists the values of $t_{CMD,OFF,MAX}$ and T_{RFW}.</p>	<p>3.2.1.6 If the Remote Field Off is not sensed within $t_{CMD,OFF,MAX} + T_{RFW}$ after the end of the frame, the Target SHALL not send a Response.</p> <p>Appendix C lists the values of $t_{CMD,OFF,MAX}$ and T_{RFW}.</p>

3.3 Initial RF Collision Avoidance

Figure 3 shows the flow chart for Initial RF Collision Avoidance that is applied by the Poller (using either Passive Communication Mode or Active Communication Mode) before it generates an Operating Field to send the first Command of the Technology Detection Activity.

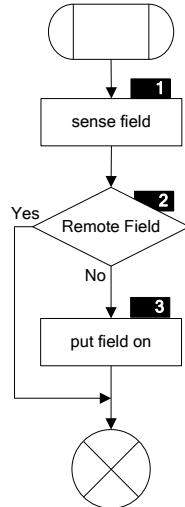


Figure 3: Initial RF Collision Avoidance – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 3.

Requirements 3: Initial RF Collision Avoidance

Poll Mode

3.3.1.1 Symbol 1:

The Poller SHALL check that Remote Field Off applies during a time $T_{ID} + n \times T_{RFW}$:

- T_{ID} SHALL be greater than or equal to $T_{ID,MIN}$.
- The integer value of n SHALL be randomly generated in the range from n_{MIN} to n_{MAX} .

Appendix C lists the values of $T_{ID,MIN}$, T_{RFW} , n_{MIN} and n_{MAX} .

3.3.1.2 Symbol 2:

If the check in Symbol 1 is successful, the Poller SHALL proceed to Symbol 3.

Otherwise the Poller SHALL conclude Initial RF Collision Avoidance.

3.3.1.3 Symbol 3:

The Poller SHALL turn the Operating Field to the Operating Field On condition (as defined in [ANALOG]) and SHALL conclude Initial RF Collision Avoidance.

3.4 Subsequent RF Collision Avoidance

Figure 4 shows the flow chart that is applied by a Peer using Active Communication Mode before it generates an Operating Field. This flow chart covers the complete communication except for the first Command of the Technology Detection Activity.

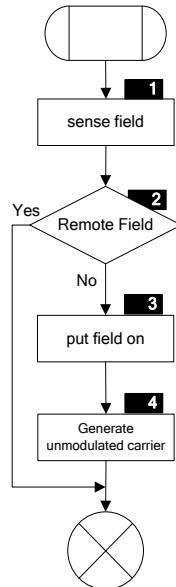


Figure 4: Subsequent RF Collision Avoidance – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 4.

Requirements 4: Subsequent RF Collision Avoidance**Poll and Listen Mode (Active Communication Mode)****3.4.1.1** Symbol 1:

The Peer SHALL check that Remote Field Off applies during a time $T_{AD} + n \times T_{RFW}$.

- If Subsequent RF Collision Avoidance is performed prior to sending an ATR_RES Response, the integer value of n SHALL be randomly generated in the range from n_{MIN} to n_{MAX} .
- Otherwise the integer value n SHALL be 0.

Appendix C lists the values of T_{AD} , T_{RFW} , n_{MIN} and n_{MAX} .

The Peer SHALL proceed to Symbol 2.

3.4.1.2 Symbol 2:

If the check in Symbol 1 is successful, the Peer SHALL proceed to Symbol 3.

Otherwise the Peer SHALL conclude Subsequent RF Collision Avoidance.

3.4.1.3 Symbol 3:

The Peer SHALL turn the Operating Field to the Operating Field On state (as defined in [ANALOG]).

The Peer SHALL proceed to Symbol 4.

3.4.1.4 Symbol 4:

Before proceeding, the Peer SHALL maintain an Unmodulated Carrier for at least T_{ARFG} :

- T_{ARFG} SHALL be greater than or equal to $T_{ARFG,MIN}$.

Appendix C lists the value of $T_{ARFG,MIN}$.

The Peer SHALL conclude Subsequent RF Collision Avoidance.

4 Listen Mode – Generic Requirements

The following generic requirements apply to Listen Mode.

Requirements 5: Listen Mode – Generic

Listen Mode

- | |
|--|
| <p>4.1.1.1 In order to enter the Listen Mode state machine, the Operating Field of the NFC Forum Device SHALL be in the Operating Field Off condition.</p> <p>4.1.1.2 If, during a single period of Remote Field On, the Listener responds only to a single Technology and answers corresponding Poll Commands with a single Response, then the Listener SHALL maintain a single state machine.</p> <p>4.1.1.3 If, during a single period of Remote Field On, the Listener responds to multiple Poll Commands in different Technologies and/or to a single Poll Command with multiple Responses, then the Listener SHALL maintain the equivalent number of independent state machines (i.e. a state machine for each Response).</p> <p>4.1.1.4 The start State of the Listener is the NO_REMOTE_FIELD State.</p> <p>4.1.1.5 When the Listener uses Passive Communication Mode and if there is No Remote Field Sensed and the Listener is not in NO_REMOTE_FIELD State, the Listener SHALL enter the NO_REMOTE_FIELD State within a delay not greater than t_{FIELD_OFF}.
Appendix C lists the Min and Max values of t_{FIELD_OFF}.</p> <p>4.1.1.6 For each state machine a Listener SHALL keep a record of its current Technology. When the Listener is in the IDLE State, it SHALL update this record in accordance with 6.2.1.2. Following its first Response to a Command in the IDLE State, the Listener SHALL set the current Technology to the Technology used for this first Response. If the Technology is subsequently changed, following the exchange of PSL_REQ Command/PSL_RES Response or in the SLEEP_AF State, the Listener SHALL update the current Technology accordingly.
If a Listener detects RF field modulation that is not consistent with the current Technology, then it SHALL not send a Response and either:</p> <ul style="list-style-type: none"> • Remain in its current State (recommended) • Go to IDLE State • Go to SLEEP_A State. This option is only applicable when the Listener is in READY_A*, READY_A'*, READY_A"* or ACTIVE_A* State and SHALL NOT be used when it is in any other State. <p>NOTE If the Listener is unable to handle the RF field modulation, it might reset (enter IDLE State).</p> <p>NOTE It is recommended that design implementations not reset but remain in the Listener's current State, which allows any ongoing transaction to continue.</p> |
|--|

5 Listen Mode – Configuration

Configuration Parameters need to be set before the Listen Mode state machine can be started. The Configuration Parameters defined in this section are used to configure a single Listen Mode state machine. Table 7 lists the Configuration Parameters.

Table 7: Listen Mode – Configuration Parameters

Name	Format	Size	Description
CON_LISTEN_ACM	binary	1 bit	<p>Controls whether to listen for Active Communication Mode or not:</p> <ul style="list-style-type: none"> • 1b: Listen for ATR_REQ Commands from an NFC-DEP Initiator indicating Active Communication Mode. • 0b: Ignore ATR_REQ Commands from an NFC-DEP Initiator indicating Active Communication Mode
CON_LISTEN_DEP_A	binary	1 bit	<p>Controls whether to listen for NFC-A with NFC-DEP support or not:</p> <ul style="list-style-type: none"> • 1b: Listen for NFC-A with NFC-DEP support • 0b: Do not listen for NFC-A with NFC-DEP support
CON_LISTEN_DEP_F	binary	1 bit	<p>Controls whether to listen for NFC-F with NFC-DEP support or not:</p> <ul style="list-style-type: none"> • 1b: Listen for NFC-F with NFC-DEP support • 0b: Do not listen for NFC-F with NFC-DEP support
CON_LISTEN_T3TP	binary	1 bit	<p>Controls whether to listen for NFC-F with Type 3 Tag Platform support or not:</p> <ul style="list-style-type: none"> • 1b: Listen for NFC-F with Type 3 Tag Platform support • 0b: Do not listen for NFC-F with Type 3 Tag Platform support
CON_LISTEN_T4ATP	binary	1 bit	<p>Controls whether to listen for NFC-A with Type 4 Tag Platform support or not:</p> <ul style="list-style-type: none"> • 1b: Listen for NFC-A with Type 4 Tag Platform support • 0b: Do not listen for NFC-A with Type 4 Tag Platform support
CON_LISTEN_T4BTP	binary	1 bit	<p>Controls whether to listen for NFC-B with Type 4 Tag Platform support or not:</p> <ul style="list-style-type: none"> • 1b: Listen for NFC-B Technology with Type 4 Tag Platform support • 0b: Do not listen for NFC-B with Type 4 Tag Platform support

Name	Format	Size	Description
CON_SYS_CODE[N]	array of Byte Sequences	variable (2 bytes × N)	If configured for the Type 3 Tag Platform, each element contains an ordered list of N system codes maintained by the adjacent upper layer (N>0). Otherwise this configuration parameter is not applicable.
CON_SENSF_RES[N]	array of Byte Sequences	variable (16 bytes × N)	If configured for the Type 3 Tag Platform, each element contains values for Byte 2-17 of the SENSF_RES Response (as specified in [DIGITAL]), consisting of NFCID2, PAD1, MRTI_{CHECK} , MRTI_{UPDATE} , and PAD2. Otherwise this configuration parameter is not applicable. N needs to be the same value as in CON_SYS_CODE[N].

- NOTE If the Listener responds to a single Poll Command with multiple Responses, then, if the Listener cannot send all possible Responses, it is recommended that the Listener foresees Configuration Parameters for each corresponding state machine and criteria for deciding which subset of Responses to send.
- NOTE For NFC-B and NFC-F, when the Listener sends multiple Responses, it is recommended that the Listener send each single Response within a single timeslot.

6 Listen Mode – State Machine

This section defines the listen side behavior of the T3T and T4T Platforms and the behavior of a Target by using a state machine. A Listener implementing all or part of this functionality needs to implement at least one such state machine, but might implement more than one. The Listen Mode state machine includes all possible State transitions caused by Commands specified in [DIGITAL] and [T3T].

Appendix B contains a state diagram visualizing the state machine defined in this section. The diagram is informative only.

NOTE Not all of the functionality of the Listen Mode state machine is necessarily mandatory for a Listener implementation.

NOTE The behavior of Type 2 Tag Commands are out of scope of this specification and are therefore not included in this state machine.

NOTE Other than in **NO_REMOTE_FIELD** State, it is assumed for Passive Communication Mode that the Operating Field is stable enough and provides enough energy to maintain state.

6.1 NO_REMOTE_FIELD State

The requirements in this section apply to the **NO_REMOTE_FIELD** State.

Requirements 6: Listen Mode – NO_REMOTE_FIELD State

Listen Mode

6.1.1.1 If Remote Field On the Listener SHALL enter the **IDLE** State.

Otherwise the Listener MAY conclude the state machine and therefore the Listen Mode.

NOTE After Remote Field On, the Listener needs to be able to respond within the Guard Times defined in [DIGITAL].

6.2 IDLE State

The requirements in this section apply to the **IDLE** State. In this State the Listener is ready to receive Poll Commands.

Requirements 7: Listen Mode – IDLE State

Listen Mode

- 6.2.1.1** The Listener SHALL become ready to respond to incoming Commands within the Guard Times (GT) defined per Technology in [DIGITAL].
 - 6.2.1.2** If the transition to **IDLE** State was from **READY_A**, **READY_A'**, **READY_A''** or **ACTIVE_A**, the Listener SHALL only respond to Commands in NFC-A Technology.
If the transition to **IDLE** State was from **SLEEP_B**, **READY_B_DECL** or **READY_B_REQU**, the Listener SHALL only respond to Commands in NFC-B Technology.
If the transition to **IDLE** State was from **NO_REMOTE_FIELD**, **ATR_READY_A**, **ATR_READY_F**, **TARGET_A** or **TARGET_F**, the Listener MAY respond in any Technology it is configured for.
 - 6.2.1.3** If CON_LISTEN_DEP_A is equal to 1b or CON_LISTEN_T4ATP is equal to 1b, and upon receipt of a Valid ALL_REQ Command, and after it has transmitted its SENS_RES Response, the Listener SHALL enter the **READY_A** State.
 - 6.2.1.4** If CON_LISTEN_DEP_A is equal to 1b or CON_LISTEN_T4ATP is equal to 1b, and upon receipt of a Valid SENS_REQ Command, and after it has transmitted its SENS_RES Response, the Listener SHALL enter the **READY_A** State.
 - 6.2.1.5** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid SENSB_REQ Command that contains an N equal to 1, an AFI that matches its own AFI and after it has transmitted its SENSB_RES Response, the Listener SHALL enter the **READY_B_DECL** State.
 - 6.2.1.6** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid ALLB_REQ Command that contains an N equal to 1, an AFI that matches its own AFI, and after it has transmitted its SENSB_RES Response, the Listener SHALL enter the **READY_B_DECL** State.
 - 6.2.1.7** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, its R is 1, and after it has transmitted its SENSB_RES Response, the Listener SHALL enter the **READY_B_DECL** State.
 - 6.2.1.8** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, its R is 1, and after it has transmitted its SENSB_RES Response, the Listener SHALL enter the **READY_B_DECL** State.
 - 6.2.1.9** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matched its own AFI, and its R is greater than 1, the Listener SHALL enter the **READY_B_REQU** State and SHALL NOT send a Response.
-

Listen Mode

-
- 6.2.1.10** If CON_LISTEN_T4BTP is equal to 1b, and upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is greater than 1, the Listener SHALL enter the **READY_B_REQ** State and SHALL NOT send a Response.
 - 6.2.1.11** If CON_LISTEN_T3TP is equal to 1b and the Listener has received a Valid CHECK or UPDATE Command (as defined in [T3T]), and the value of NFCID2 matches any of the NFCID2 values in the CON_SENSF_RES array, the Listener SHALL send its Response and SHALL then enter the **CARD_EMULATOR_3** State.
 - 6.2.1.12** If CON_LISTEN_T3TP is equal to 1b and upon receipt of a Proprietary Command for the Type 3 Tag Platform, the Listener MAY respond to this Proprietary Command. If it does, the Listener SHALL enter the **CARD_EMULATOR_3** State. If not, it SHALL be treated as OTHER.
-

Listen Mode

- 6.2.1.13** If CON_LISTEN_DEP_F is equal to 1b or CON_LISTEN_T3TP is equal to 1b and the Listener has received a Valid SENSF_REQ Command, the Listener SHALL handle the SENSF_REQ Command as specified below, and, if this results in transmitting a SENSF_RES Response, the Listener SHALL enter the **READY_F** State.

SENSF_REQ Command handling :

If CON_LISTEN_DEP_F is equal to 1b and the SENSF_REQ Command allows sending a SENSF_RES Response indicating NFC-DEP protocol (as defined in [DIGITAL]), the Listener SHALL send a corresponding SENSF_RES Command with an NFCID2 generated according to the rules defined in [DIGITAL]. This NFCID2 has to be remembered for usage in other States.

NOTE A SENSF_RES Response indicating NFC-DEP can only be sent if the corresponding SENSF_REQ Command has the SC set to FFFFh and the RC code set to 0h.

Otherwise, and if CON_LISTEN_T3TP is equal to 1b, compare the value of SC in the SENSF_REQ Command sequentially with the system code values contained in CON_SYS_CODE. If the values match according to the conditions defined below, the Listener SHALL stop the comparison and transmit its SENSF_RES Response. The SENSF_RES Response SHALL be coded using the values in CON_SENSF_RES at the same index as the CON_SYS_CODE entry that matched the SC in the SENSF_REQ Command.

If the Listener intends to include the RD bytes in the SENSF_RES Response according to the requirements given in [DIGITAL]:

- If the preceding SENSF_REQ Command contained an RC byte set to 01h, the value of the RD bytes SHALL be equal to the matching CON_SYS_CODE value.
- If the preceding SENSF_REQ Command contained an RC byte set to 02h the value of the RD bytes SHALL be set according to the rules defined in [DIGITAL] for the RD Format Advanced Protocol Features

An SC value in the SENSF_REQ Command corresponds to the value contained in CON_SYS_CODE at index X (with X being in the range from 1 to N) if any of the following is true:

- The value of SC in the SENSF_REQ Command is equal to FFFFh
- The value of SC in the SENSF_REQ Command is equal to the value of CON_SYS_CODE[X]
- The first byte of SC in the SENSF_REQ Command has a value of FFh and the value of the second byte equals the value of the second byte of CON_SYS_CODE[X], or
- The second byte of SC in the SENSF_REQ Command has a value of FFh and the value of the first byte equals the value of the first byte of the CON_SYS_CODE[X].

-
- 6.2.1.14** If CON_LISTEN_ACM is equal to 1b, and upon receipt of a Valid ATR_REQ Command that was sent using NFC-A, and after it has transmitted its ATR_RES Response, the Listener SHALL enter the **ATR_READY_A** State.
-

Listen Mode

- 6.2.1.15** If CON_LISTEN_ACM is equal to 1b, and upon receipt of a Valid ATR_REQ Command that was sent using NFC-F, and after it has transmitted its ATR_RES Response, the Listener SHALL enter the **ATR_READY_F** State.
- 6.2.1.16** Upon receipt of a Proprietary Command, the Listener MAY respond. If it does, further processing is out of scope of this specification. If not, it SHALL be treated as OTHER.
- 6.2.1.17** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **IDLE** State.
-

NOTE In contrast to ISO/IEC 18092, this specification does not define the **Sleep_AF** State for a Listener configured as a Target in Active Communication Mode.

6.3 READY_A State and READY_A* State

The requirements in this section apply to the **READY_A** and **READY_A*** States. In these States the Listener expects an SDD_REQ Command to retrieve the complete NFCID1.

Requirements 8: Listen Mode – READY_A State and READY_A* State

Listen Mode

- 6.3.1.1** Upon receipt of a Valid SDD_REQ CL1 Command, the Listener SHALL send its NFCID1 CL1 and stay in the **READY_A (READY_A*)** State.
- 6.3.1.2** Upon receipt of a Valid SEL_REQ CL1 Command with a matching NFCID1 CL1, a Listener with a single-size NFCID1 SHALL send its SEL_RES Response and SHALL enter the **ACTIVE_A (ACTIVE_A*)** State when it is selected with its complete NFCID1. The Listener SHALL indicate in its SEL_RES Response that the NFCID1 is complete.
- 6.3.1.3** Upon receipt of a Valid SEL_REQ CL1 Command with a matching NFCID1 CL1, a Listener with a double- or triple-size NFCID1 SHALL send its SEL_RES Response and SHALL enter the **READY_A' (READY_A'*)** State when it is selected with its complete NFCID1 CL1. The Listener SHALL indicate in its SEL_RES Response that the NFCID1 is NOT complete.
- 6.3.1.4** Upon receipt of a Proprietary Command, the Listener MAY respond. If it does, further processing is out of scope of this specification. If not, it SHALL be treated as OTHER.
- 6.3.1.5** If OTHER, the Listener SHALL NOT send a Response and
- When it is in the **READY_A** State, the Listener SHALL return to the **IDLE** State.
 - When it is in the **READY_A*** State, the Listener SHALL return to the **SLEEP_A** State.
-

6.4 READY_A' State and READY_A'* State

The requirements in this section apply to the **READY_A'** and **READY_A'*** States. The **READY_A'** and **READY_A'*** States are intermediate States that only exist for Listeners with double-size and triple-size NFCID1. In these States the cascade level 1 of the NFCID1 has been selected.

Requirements 9: Listen Mode – READY_A' State and READY_A'* State

Listen Mode

- 6.4.1.1** Upon receipt of a Valid SDD_REQ CL2 Command, a Listener SHALL send its NFCID1 CL2 and stay in the **READY_A' (READY_A'*)** State.
 - 6.4.1.2** Upon receipt of a Valid SEL_REQ CL2 Command with a matching NFCID1 CL2, a Listener with a double-size NFCID1 SHALL send its SEL_RES Response and SHALL enter the **ACTIVE_A (ACTIVE_A*)** State when it is selected with its complete NFCID1. The Listener SHALL indicate in its SEL_RES Response that the NFCID1 is complete.
 - 6.4.1.3** Upon receipt of a Valid SEL_REQ CL2 Command with a matching NFCID1 CL2, a Listener with a triple-size NFCID1 SHALL send its SEL_RES Response and SHALL enter the **READY_A'' (READY_A''*)** State when it is selected with its complete NFCID1 CL2. The Listener SHALL indicate in its SEL_RES Response that the NFCID1 is NOT complete.
 - 6.4.1.4** If OTHER, the Listener SHALL NOT send a Response.
When it is in the **READY_A'** State, the Listener SHALL return to the **IDLE** State.
When it is in the **READY_A'*** State, the Listener SHALL return to the **SLEEP_A** State.
-

6.5 READY_A" State and READY_A** State

The requirements in this section apply to the **READY_A"** and **READY_A**** States. The **READY_A"** and **READY_A**** States are intermediate States that only exist for Listeners with triple-size NFCID1. In these States the cascade level 1 and 2 of the NFCID1 have been selected.

Requirements 10: Listen Mode – READY_A" State and READY_A** State

Listen Mode

- 6.5.1.1** Upon receipt of a Valid SDD_REQ CL3 Command, a Listener SHALL send its NFCID1 CL3 and stay in the **READY_A" (READY_A**)** State.
 - 6.5.1.2** Upon receipt of a Valid SEL_REQ CL3 Command with a matching NFCID1 CL3, a Listener with a triple-size NFCID1 SHALL send its SEL_RES Response and SHALL enter the **ACTIVE_A (ACTIVE*)** State when it is selected with its complete NFCID1. The Listener SHALL indicate in its SEL_RES Response that the NFCID1 is complete.
 - 6.5.1.3** If OTHER, the Listener SHALL NOT send a Response.
When it is in the **READY_A"** State the Listener SHALL return to the **IDLE** State.
When it is in the **READY_A**** State the Listener SHALL return to the **SLEEP_A** State.
-

6.6 ACTIVE_A State and ACTIVE_A* State

The requirements in this section apply to the **ACTIVE_A** and **ACTIVE_A*** States. In these States the Listener expects Commands for protocol activation.

Requirements 11: Listen Mode – ACTIVE_A State and ACTIVE_A* State

Listen Mode

- 6.6.1.1** Upon receipt of a Valid SLP_REQ Command, the Listener SHALL enter the **SLEEP_A** State.
 - 6.6.1.2** If CON_LISTEN_DEP_A is equal to 1b, and upon receipt of a Valid ATR_REQ Command, the Listener SHALL send its ATR_RES Response and SHALL enter the **ATR_READY_A**.
 - 6.6.1.3** If CON_LISTEN_T4ATP is equal to 1b, and upon receipt of a Valid RATS Command, the Listener SHALL send its ATS Response and SHALL enter the **CARD_EMULATOR_4A** State.
 - 6.6.1.4** Upon receipt of a Valid Type 2 Tag Command or a Proprietary Command, the Listener MAY respond. If it does, further processing is out of scope of this specification. If not, it SHALL be treated as OTHER.
 - 6.6.1.5** If OTHER, the Listener SHALL NOT send a Response and
 - When it is in the **ACTIVE_A** State, the Listener SHALL return to the **IDLE** State.
 - When it is in the **ACTIVE_A*** State, the Listener SHALL return to the **SLEEP_A** State.
-

6.7 SLEEP_A State

The requirements in this section apply to the **SLEEP_A** State. In this State the Listener only responds to an ALL_REQ Command.

Requirements 12: Listen Mode – SLEEP_A State

Listen Mode

- 6.7.1.1** Upon receipt of a Valid ALL_REQ Command, the Listener SHALL send its SENS_RES Response and SHALL enter the **READY_A*** State.
 - 6.7.1.2** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **SLEEP_A** State.
-

6.8 ATR_READY_A State

The requirements in this section apply to the **ATR_READY_A** State. In this State the Listener expects a PSL_REQ Command or a DEP_REQ Command.

Requirements 13: Listen Mode – ATR_READY_A State

Listen Mode

- 6.8.1.1** Upon receipt of a Valid DEP_REQ Command with a matching DID value, the Listener SHALL send its DEP_RES Response and SHALL enter the **TARGET_A** State.
 - 6.8.1.2** Upon receipt of a Valid PSL_REQ Command with a matching DID value and with DSI and DRI set to 000b, the Listener SHALL send its PSL_RES Response and SHALL enter the **TARGET_A** State.
[DIGITAL] provides details on DSI and DRI coding.
 - 6.8.1.3** Upon receipt of a Valid PSL_REQ Command with a matching DID value and with DSI and DRI set to 001b or 010b, the Listener SHALL send its PSL_RES Response and SHALL enter the **TARGET_F** State.
[DIGITAL] provides details on DSI and DRI coding.
 - 6.8.1.4** Upon receipt of a Valid DSL_REQ Command with a matching DID value, the Listener SHALL send its DSL_RES Response and SHALL enter the **SLEEP_AF** State.
 - 6.8.1.5** Upon receipt of a Valid RLS_REQ Command with a matching DID value, the Listener SHALL send its RLS_RES Response and SHALL enter the **IDLE** State.
 - 6.8.1.6** If OTHER, the Listener SHALL NOT send a Response and SHALL enter the **TARGET_A** State.
-

6.9 TARGET_A State

The requirements in this section apply to the **TARGET_A** State. In this State the Listener expects higher layer messages.

Requirements 14: Listen Mode – TARGET_A State

Listen Mode

- 6.9.1.1** Upon receipt of a Valid DEP_REQ Command with a matching DID value, the Listener SHALL send its DEP_RES Response and stay in the **TARGET_A** State.
 - 6.9.1.2** Upon receipt of a Valid DSL_REQ Command with a matching DID value, the Listener SHALL send its DSL_RES Response and SHALL enter the **SLEEP_AF** State.
 - 6.9.1.3** Upon receipt of a Valid RLS_REQ Command with a matching DID value, the Listener SHALL send its RLS_RES Response and SHALL enter the **IDLE** State.
 - 6.9.1.4** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **TARGET_A** State.
-

6.10 CARD_EMULATOR_4A State

The requirements in this section apply to the **CARD_EMULATOR_4A** State. In this State the Listener expects higher layer messages or an S(DESELECT) Request (see [DIGITAL]).

Requirements 15: Listen Mode – CARD_EMULATOR_4A State

Listen Mode

- 6.10.1.1** If the first Command received in this state is a Proprietary Command, the Listener MAY respond to this Proprietary Command. In either case the Listener SHALL stay in the **CARD_EMULATOR_4A** State.
 - 6.10.1.2** Upon receipt of a Valid S(DESELECT) Request with a matching DID value (as defined in [DIGITAL]), the Listener SHALL send its S(DESELECT) Response and SHALL enter the **SLEEP_A** State.
 - 6.10.1.3** Upon receipt of a Valid Block in compliance with the ISO-DEP Protocol (as specified in [DIGITAL]), the Listener SHALL send its Response and SHALL stay in the **CARD_EMULATOR_4A** State.
 - 6.10.1.4** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **CARD_EMULATOR_4A** State.
-

6.11 READY_B_REQU State

The requirements in this section apply when the Listener is in the **READY_B_REQU** State. In this State the Listener expects an ALLB_REQ Command, a SENSB_REQ Command, or a corresponding SLOT_MARKER Command.

Requirements 16: Listen Mode – READY_B_REQU State

Listen Mode	
6.11.1.1	Upon receipt of a Valid SENSB_REQ Command that contains an N equal to 1 and an AFI that matches its own AFI, the Listener SHALL send its SENSB_RES Response and SHALL enter the READY_B_DECL State.
6.11.1.2	Upon receipt of a Valid ALLB_REQ Command that contains an N equal to 1 and an AFI that matches its own AFI, the Listener SHALL send its SENSB_RES Response and SHALL enter the READY_B_DECL State.
6.11.1.3	Upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and whose R is 1, the Listener SHALL send its SENSB_RES Response and SHALL enter the READY_B_DECL State
6.11.1.4	Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and whose R is 1, the Listener SHALL send its SENSB_RES Response and SHALL enter the READY_B_DECL State.
6.11.1.5	Upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and whose R is greater than 1, the Listener SHALL NOT send a Response and SHALL stay in the READY_B_REQU State.
6.11.1.6	Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and whose R is greater than 1, the Listener SHALL NOT send a Response and SHALL stay in the READY_B_REQU State.
6.11.1.7	Upon receipt of a Valid SLOT_MARKER Command indicating a Slot number matching R (as calculated at the reception of the last SENSB_REQ Command or ALLB_REQ Command), the Listener SHALL send its SENSB_RES Response and SHALL enter the READY_B_DECL State.
6.11.1.8	Upon receipt of a Valid SENSB_REQ Command that contains an AFI that does not match its own AFI, the Listener SHALL NOT send its SENSB_RES Response and SHALL enter the IDLE State.
6.11.1.9	Upon receipt of a Valid ALLB_REQ Command that contains an AFI that does not match its own AFI, the Listener SHALL NOT send a Response and SHALL enter the IDLE State.
6.11.1.10	If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the READY_B_REQU State.

6.12 READY_B_DECL State

The requirements in this section apply when the Listener is in the **READY_B_DECL** State. In this State the Listener expects an ATTRIB Command or a SLPB_REQ Command.

Requirements 17: Listen Mode – READY_B_DECL State

Listen Mode

- 6.12.1.1** Upon receipt of a Valid ATTRIB Command with a matching value of NFCID0, the Listener SHALL send its ATTRIB Response and SHALL enter the **CARD_EMULATOR_4B** State.
- 6.12.1.2** Upon receipt of a Valid SLPB_REQ Command with a matching value of NFCID0, the Listener SHALL send its SLPB_RES Response and SHALL enter the **SLEEP_B** State.
- 6.12.1.3** Upon receipt of a Valid SENSB_REQ Command that contains an N equal to 1 and an AFI that matches its own AFI, the Listener SHALL send its SENSB_RES Response and SHALL stay in the **READY_B_DECL** State.
- 6.12.1.4** Upon receipt of a Valid ALLB_REQ Command that contains an N equal to 1 and an AFI that matches its own AFI, the Listener SHALL send its SENSB_RES Response and SHALL stay in the **READY_B_DECL** State.
- 6.12.1.5** Upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is 1, the Listener SHALL send its SENSB_RES Response and SHALL stay in the **READY_B_DECL** State.
- 6.12.1.6** Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is 1, the Listener SHALL send its SENSB_RES Response and SHALL stay in the **READY_B_DECL** State.
- 6.12.1.7** Upon receipt of a Valid SENSB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is greater than 1, the Listener SHALL NOT send a Response and SHALL enter the **READY_B_REQU** State.
- 6.12.1.8** Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is greater than 1, the Listener SHALL NOT send a Response and SHALL enter the **READY_B_REQU** State.
- 6.12.1.9** Upon receipt of a Valid SENSB_REQ Command that contains an AFI that does not match its own AFI, the Listener SHALL NOT send a Response and SHALL enter the **IDLE** State.
- 6.12.1.10** Upon receipt of a Valid ALLB_REQ Command that contains an AFI that does not match its own AFI, the Listener SHALL NOT send a Response and SHALL enter the **IDLE** State.
- 6.12.1.11** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **READY_B_DECL** State.

6.13 SLEEP_B State

The requirements in this section apply when the Listener is in the **SLEEP_B** State. In this State the Listener expects an ALLB_REQ Command.

Requirements 18: Listen Mode – SLEEP_B State

Listen Mode

- 6.13.1.1** Upon receipt of a Valid ALLB_REQ Command that contains an N equal to 1 and an AFI that matches its own AFI, the Listener SHALL send its SENSB_RES Response and SHALL enter the **READY_B_DECL** State.
 - 6.13.1.2** Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is 1, the Listener SHALL send its SENSB_RES Response and SHALL enter the **READY_B_DECL** State.
 - 6.13.1.3** Upon receipt of a Valid ALLB_REQ Command that contains an N greater than 1, an AFI that matches its own AFI, and its R is greater than 1, the Listener SHALL NOT send a Response and SHALL enter the **READY_B_REQU** State.
 - 6.13.1.4** Upon receipt of a Valid ALLB_REQ Command that contains an AFI that does not match its own AFI, the Listener SHALL NOT send a Response and SHALL enter the **IDLE** State.
 - 6.13.1.5** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **SLEEP_B** State.
-

6.14 CARD_EMULATOR_4B State

The requirements in this section apply when the Listener is in the **CARD_EMULATOR_4B** State. In this State the Listener expects higher layer messages or an S(DESELECT) Request (see [DIGITAL]).

Requirements 19: Listen Mode – CARD_EMULATOR_4B State

Listen Mode

- 6.14.1.1** Upon receipt of a Valid S(DESELECT) Request with a matching DID value (as defined in [DIGITAL]), the Listener SHALL send its S(DESELECT) Response and SHALL enter the **SLEEP_B** State.
 - 6.14.1.2** Upon receipt of a Valid Block in compliance with the ISO-DEP Protocol (as specified in [DIGITAL]), the Listener SHALL send its Response and SHALL stay in the **CARD_EMULATOR_4B** State.
 - 6.14.1.3** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **CARD_EMULATOR_4B** State.
-

6.15 READY_F State

The requirements in this section apply to the **READY_F** State. In this State the Listener expects an ATR_REQ Command, a SENSF_REQ Command, a CHECK Command, or an UPDATE Command.

Requirements 20: Listen Mode – READY_F State

Listen Mode

- 6.15.1.1** If CON_LISTEN_DEP_F is equal to 1b, and upon receipt of a Valid ATR_REQ Command where the first 8 bytes of NFCID3i match the value of the NFCID2 of the last SENSF_RES Response sent that indicated support for NFC-DEP, the Listener SHALL send its ATR_RES Response and SHALL enter the **ATR_READY_F** State.
 - 6.15.1.2** If CON_LISTEN_T3TP is equal to 1b, and upon receipt of a Valid CHECK or UPDATE Command (as defined in [T3T]), and the value of NFCID2 matches any of the NFCID2 values in the CON_SENSF_RES array, the Listener SHALL send its Response and SHALL enter the **CARD_EMULATOR_3** State.
 - 6.15.1.3** Upon receipt of a Valid SENSF_REQ Command, the Listener SHALL handle the SENSF_REQ Command as specified in 6.2.1.13 and SHALL stay in the **READY_F** State.
 - 6.15.1.4** If CON_LISTEN_T3TP is equal to 1b, and upon receipt of a Proprietary Command for the Type 3 Tag Platform, the Listener MAY respond to this Proprietary Command. If it does, the Listener SHALL enter the **CARD_EMULATOR_3** State. If not, it SHALL be treated as OTHER.
 - 6.15.1.5** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **READY_F** State.
-

NOTE The **READY_F** State is known as MODE_0 in other, non-NFC-Forum, specifications (e.g., [JIS_X_6319-4]).

6.16 ATR_READY_F State

The requirements in this section apply to the **ATR_READY_F** State. In this State the Listener expects a PSL_REQ Command or a DEP_REQ Command.

Requirements 21: Listen Mode – ATR_READY_F State

Listen Mode

- 6.16.1.1** Upon receipt of a Valid DEP_REQ Command with a matching DID value, the Listener SHALL send its DEP_RES Response and SHALL enter the **TARGET_F** State.
 - 6.16.1.2** Upon receipt of a Valid PSL_REQ Command with a matching DID value and with DSI and DRI set to 001b or 010b, the Listener SHALL send its PSL_RES Response and SHALL enter the **TARGET_F** State.
[DIGITAL] provides details on DSI and DRI coding.
 - 6.16.1.3** Upon receipt of a Valid PSL_REQ Command with a matching DID value and with DSI and DRI set to 000b, the Listener SHALL send its PSL_RES Response and SHALL enter the **TARGET_A** State.
[DIGITAL] provides details on DSI and DRI coding.
 - 6.16.1.4** Upon receipt of a Valid RLS_REQ Command with a matching DID value, the Listener SHALL send its RLS_RES Response and SHALL enter the **IDLE** State.
 - 6.16.1.5** Upon receipt of a Valid DSL_REQ Command with a matching DID value, the Listener SHALL send its DSL_RES Response and SHALL enter the **SLEEP_AF** State.
 - 6.16.1.6** If OTHER, the Listener SHALL NOT send a Response and SHALL enter the **TARGET_F** State.
-

6.17 TARGET_F State

The requirements in this section apply to the **TARGET_F** State. In this State the Listener expects higher layer messages.

Requirements 22: Listen Mode – TARGET_F State

Listen Mode

- 6.17.1.1** Upon receipt of a Valid DEP_REQ Command with a matching DID value, the Listener SHALL send its DEP_RES Response and SHALL stay the **TARGET_F** State.
 - 6.17.1.2** Upon receipt of a Valid RLS_REQ Command with a matching DID value, the Listener SHALL send its RLS_RES Response and SHALL enter the **IDLE** State.
 - 6.17.1.3** Upon receipt of a Valid DSL_REQ Command with a matching DID value, the Listener SHALL send its DSL_RES Response and SHALL enter the **SLEEP_AF** State.
 - 6.17.1.4** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **TARGET_F** State.
-

6.18 CARD_EMULATOR_3 State

The requirements in this section apply to the **CARD_EMULATOR_3 State**. In this State the Listener expects Valid Commands for communication with a Type 3 Tag Platform (as defined in [DIGITAL]).

Requirements 23: Listen Mode – CARD_EMULATOR_3 State

Listen Mode

- 6.18.1.1** Upon receipt of a Valid CHECK or UPDATE Command (as defined in [T3T]), and, if the value of NFCID2 matches any of the NFCID2 values in the CON_SENSF_RES array, the Listener SHALL handle the Command and SHALL stay in the **CARD_EMULATOR_3 State**.
 - 6.18.1.2** Upon receipt of a Valid SENSF_REQ Command, the Listener SHALL handle the SENSF_REQ Command as specified in 6.2.1.13 and SHALL stay in the **CARD_EMULATOR_3 State**.
 - 6.18.1.3** Upon receipt of a Proprietary Command for the Type 3 Tag Platform, the Listener MAY respond to this Proprietary Command. If it does, the Listener SHALL stay in the **CARD_EMULATOR_3 State**. If not, it SHALL be treated as OTHER.
 - 6.18.1.4** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **CARD_EMULATOR_3 State**.
-

6.19 SLEEP_AF State

The requirements in this section apply to the **SLEEP_AF** State. In this State the Listener has been deselected by means of an NFC-DEP DSL_REQ Command. The Listener expects an ALL_REQ Command, a SENSF_REQ Command, a CHECK Command, an UPDATE Command or a Proprietary Command for the Type 3 Tag Platform.

Requirements 24: Listen Mode – SLEEP_AF State

Listen Mode

- 6.19.1.1** If CON_LISTEN_DEP_A is equal to 1b or CON_LISTEN_T4ATP is equal to 1b, and upon receipt of a Valid ALL_REQ Command, the Listener SHALL send its SENSF_RES Response and SHALL enter the **READY_A*** State.
- 6.19.1.2** If CON_LISTEN_DEP_F is equal to 1b or CON_LISTEN_T3TP is equal to 1b, and upon receipt of a Valid SENSF_REQ Command, the Listener SHALL handle the SENSF_REQ Command as specified in 6.2.1.13, and, if this results in transmitting a SENSF_RES Response, the Listener SHALL enter the **READY_F** State.
- 6.19.1.3** If CON_LISTEN_T3TP is equal to 1b and upon receipt of a Valid CHECK or UPDATE Command, as defined in [T3T], and the value of NFCID2 matches any of the NFCID2 values in the CON_SENSF_RES array, the Listener SHALL send its Response and SHALL enter the **CARD_EMULATOR_3** State.
- 6.19.1.4** If CON_LISTEN_T3TP is equal to 1b and upon receipt of a Proprietary Command for the Type 3 Tag Platform, the Listener MAY respond to this Proprietary Command. If it does, the Listener SHALL enter the **CARD_EMULATOR_3** State. If not, it SHALL be treated as OTHER.
- 6.19.1.5** If OTHER, the Listener SHALL NOT send a Response and SHALL stay in the **SLEEP_AF** State.

7 Poll Mode – Generic Requirements

This section contains generic requirements that need to be observed, independent of whether or not the Poller chooses to implement the Activities described in this document.

Requirements 25 contain the list of generic requirements.

Requirements 25: Generic

-
- 7.1.1.1** When the Poller sets the Operating Field to the Operating Field Off condition (carrier off, as defined in [ANALOG]) other than for:

- NFC-A modulation
- NFC-V modulation
- Active Communication Mode

then the Operating Field SHALL be set to the Operating Field Off condition for a time of at least t_{FIELD_OFF} .

Appendix C lists the Min and Max values of t_{FIELD_OFF} .

-
- 7.1.1.2** When the Poller generates a Poll Command initially after setting the Operating Field to Operating Field On condition or when generating subsequent Poll Command of different Technologies, these SHALL be preceded by a period during which the Poller sends an Unmodulated Carrier (as defined in [ANALOG]). The duration of this period is referred to as “Guard Time” and the Poller SHALL comply with the following Guard Times:

- GT_A for NFC-A
- GT_B for NFC-B
- GT_F for NFC-F
- GT_V for NFC-V
- GT_{ACM} for NFC-ACM.

If polling for NFC-F is preceded by polling for NFC-B, then GT_F is equal to GT_{BF} .

Otherwise GT_F is equal to GT_{FB} .

See [DIGITAL] for the Listen Mode Guard Time requirements for each Technology.

Appendix C lists the values of GT_A , GT_B , GT_{BF} , GT_{FB} , GT_V and GT_{ACM} .

NOTE This requirement does not apply to consecutive Poll Commands of the same Technology, nor does it apply to a Poll Command following a Sleep Command.

-
- 7.1.1.3** For the Poller, if the PSL_REQ Command is used, it SHALL be sent as the first Command of the NFC-DEP Protocol Data Exchange, i.e., before the first DEP_REQ Command:

- The PSL_REQ (A) as used in the state machine is a PSL_REQ Command with DS1 set to 000b.
- The PSL_REQ (F) as used in the state machine is a PSL_REQ Command with DS1 set to 001b or 010b.

[DIGITAL] specifies the coding of the PSL_REQ Command.

-
- 7.1.1.4** A Poller SHALL perform RF Collision Avoidance (see Section 3) before generating an Operating Field.

7.1.1.5 When the Poller includes Poll Commands for one or more Proprietary Technologies, then the Proprietary Technologies SHALL be polled after the NFC Technology(ies) defined by NFC Forum specifications.

7.1.1.6 For introducing Proprietary Technologies, the Poller SHALL wait with an Unmodulated Carrier for a period after a Poll Command. The duration of this period is the sum of FDT/FWT for the Poll Command and the Proprietary Technology Guard Time.

The resulting timing relationships are:

- If polling for Proprietary Technology is preceded immediately by polling for NFC-A, then the time $\text{PTGT}_A + \text{FDT}_{A,\text{LISTEN},\text{MAX}}$ SHALL be applied.
- If polling for Proprietary Technology is preceded immediately by polling for NFC-B, then the time $\text{PTGT}_B + \text{FWT}_{B,\text{SENSB}}$ SHALL be applied.
- If polling for Proprietary Technology is preceded immediately by polling for NFC-F, then the time $\text{PTGT}_F + \text{FDT}_{F,\text{LISTEN},\text{SENSF_REQ}}$ SHALL be applied.
- If polling for Proprietary Technology is preceded immediately by polling for NFC-V, then the time $\text{PTGT}_V + \text{FDT}_{V,\text{INVENT_NORES}}$ SHALL be applied.
- If polling for Proprietary Technology is preceded immediately by polling for NFC-ACM, then the time PTGT_{ACM} SHALL be applied.

[DIGITAL] provides details on $\text{FDT}_{A,\text{LISTEN},\text{MAX}}$, $\text{FWT}_{B,\text{SENSB}}$,
 $\text{FDT}_{F,\text{LISTEN},\text{SENSF_REQ}}$ and $\text{FDT}_{V,\text{INVENT_NORES}}$.

Appendix C lists the values of PTGT_A , PTGT_B , PTGT_F , PTGT_V and PTGT_{ACM} .

8 Poll Mode – Activity Model

Activities combine elementary blocks of [DIGITAL] into functional blocks. Each functional block has a dedicated purpose, with well-defined preconditions and post-conditions. It provides a level of detail on the Initiator/Reader functionality that is not already specified within [DIGITAL].

The Activity manages the dialogue with another device, using the Commands and Responses specified in [DIGITAL]. To perform its task, it has a well-defined set of algorithms, with one algorithm per Technology if necessary.

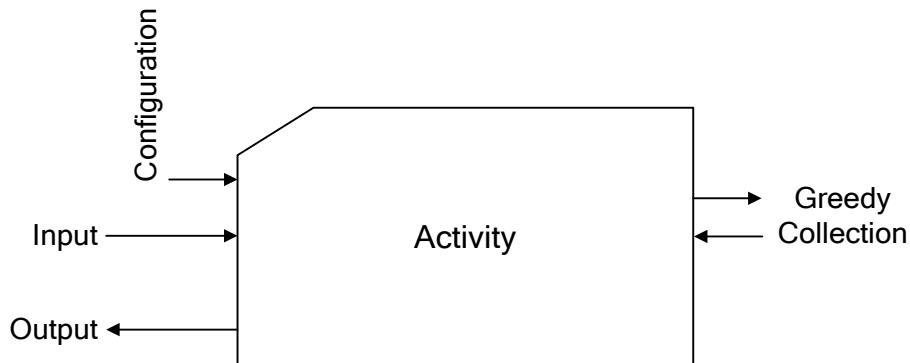


Figure 5: Activity

An Activity can have any of the following interfaces shown in Figure 5:

- Configuration Parameters
- Input Parameters
- Output Parameters
- Greedy Collection.

Configuration Parameters and Input Parameters provide the necessary flexibility on how to use the algorithm. As part of its processing, the Activity collects information on the other device. While this information might not be directly relevant for this Activity, it is stored in the Greedy Collection, so that it can be used by following Activities or the Resolution Process. Output Parameters provide the results of the Activity into the Resolution Process. For more information on the combination of Activities and the Resolution Process, see [PROFILES].

The following rules for the framework around Activities apply:

- During normal processing, Activities are not interrupted by the adjacent upper layer. If the adjacent upper layer interrupts an Activity, this is an exception.
- If an error occurs within an action block, then an error-handling task can interrupt the Activity processing. The error-handling task can choose to proceed inside the current Activity or to abort it. Error handling has to conform with [DIGITAL]. Care has to be taken with the integrity of the Greedy Collection.
- The Resolution Process can start an Activity only if the preconditions of the Activity are fulfilled. Before each execution of an Activity, the corresponding Configuration Parameters and Input Parameters need to be configured.

9 Poll Mode – Activities

An Activity uses:

- Technology-independent preconditions and post-conditions
- Configuration Parameters that can be Technology-dependent (as for the Technology Detection Activity, Collision Resolution Activity and the Device Activation Activity) or Technology-independent (as for the Data Exchange Activity and the Device Deactivation Activity)
- Technology-dependent Algorithms, using the Configuration Parameters in a Technology-specific manner

Configuration Parameters are independent from the other device and typically survive multiple transactions. This distinguishes them from the Greedy Collection, which stores information learned from the other device and therefore varies with each transaction.

The description of each Activity is structured as follows:

- The preconditions are described by:
 - The input from the Resolution Process
 - The Configuration Parameters
 - The information collected previously in the Greedy Collection
- The post-conditions are described by:
 - The information that can be used by the Resolution Process
 - The information currently in the Greedy Collection
- The algorithm is described through:
 - The flow chart
 - The requirements

The remainder of this section lists requirements and contains a detailed definition of each Activity.

9.1 Activities – Requirements

This section contains requirements that need to be observed when implementing the Activities described in this document.

Requirements 26: Activities – General

-
- 9.1.1.1** For each combination of Activities in Passive Communication Mode, the Operating Field SHALL be in the Operating Field On condition (see Section 3).
 - 9.1.1.2** For each combination of Activities, the first Activity SHALL be the Technology Detection Activity.
-

9.2 Technology Detection Activity

This section describes the Technology Detection Activity. The purpose of the Technology Detection Activity is to scan for devices of certain technologies that are within range.

9.2.1 Preconditions

The Configuration Parameters for the Technology Detection Activity are listed in Table 8.

Table 8: Technology Detection Activity – Configuration Parameters

Name	Format	Size	Description
CON_POLL_ACM	binary	1 bit	1b: Poll for NFC-ACM 0b: Do not poll for NFC-ACM
CON_POLL_A	binary	1 bit	1b: Poll for NFC-A 0b: Do not poll for NFC-A
CON_EXT_SENSB_RES	binary	1 bit	Controls whether the SENSB_REQ Command indicates support for the extended SENSB_RES byte (as defined in [DIGITAL]). 0b: Extended SENSB_RES byte is not supported 1b: Extended SENSB_RES byte is supported
CON_POLL_B	binary	1 bit	1b: Poll for NFC-B 0b: Do not poll for NFC-B
CON_POLL_F	binary	1 bit	1b: Poll for NFC-F 0b: Do not poll for NFC-F
CON_POLL_V	binary	1 bit	1b: Poll for NFC-V 0b: Do not poll for NFC-V
CON_POLL_P	binary	1 bit	1b: Poll for Proprietary Technology 0b: Do not poll for Proprietary Technology
CON_BAIL_OUT_A	binary	1 bit	1b: Bail-out after NFC-A 0b: No bail-out after NFC-A
CON_BAIL_OUT_B	binary	1 bit	1b: Bail-out after NFC-B 0b: No bail-out after NFC-B
CON_BAIL_OUT_F	binary	1 bit	1b: Bail-out after NFC-F 0b: No bail-out after NFC-F
CON_BITR	integer	1 byte	bit rate for NFC-F 2: 212 kbps 3: 424 kbps
CON_BITR_ACM	integer	1 byte	bit rate for NFC-ACM 1: 106 kbps 2: 212 kbps 3: 424 kbps
CON_TD_RESET	binary	1 bit	1b: perform a reset 0b: do no perform a reset

NOTE There is no need for a bail-out option for NFC-V, as bail-outs always occur before polling for a Proprietary Technology and the Poller always checks whether an NFC Technology defined by the NFC Forum has been detected. The bail-out options for NFC-A, NFC-B, and NFC-F are introduced to allow optimization.

There are no Input Parameters requested from the Resolution Process for this Activity.

There is no data needed from the Greedy Collection for this Activity.

9.2.2 Post-conditions

The output of the Technology Detection Activity is listed in Table 9.

Table 9: Technology Detection Activity – Output Parameters

Name	Format	Size	Description
FOUND_ACM	binary	1 bit	1b: NFC-ACM found 0b: NFC-ACM not found
FOUND_A	binary	1 bit	1b: NFC-A found 0b: NFC-A not found
FOUND_B	binary	1 bit	1b: NFC-B found 0b: NFC-B not found
FOUND_F	binary	1 bit	1b: NFC-F found 0b: NFC-F not found
FOUND_V	binary	1 bit	1b: NFC-V found 0b: NFC-V not found

NOTE The outcome of polling for Proprietary Technology is outside of the scope of this specification and therefore such result does not appear as an Output Parameter.

The data returned to the Greedy Collection are listed in Table 10.

Table 10: Technology Detection Activity – Output into Greedy Collection

Name	Format	Size	Description
GRE_POLL_ACM[]	array of Byte Sequences	variable	Each element contains a Response to an ATR_REQ Command. For NFC-ACM the array is limited to one element.
GRE_POLL_A[]	array of Byte Sequences	variable	Each element contains a Response to an ALL_REQ Command or SENS_REQ Command. For NFC-A the array is limited to one element.
GRE_POLL_B[]	array of Byte Sequences	variable	Each element contains a Response to an ALLB_REQ Command or SENSB_REQ Command. For NFC-B the array is limited to one element.
GRE_POLL_F[]	array of Byte Sequences	variable	Each element contains a Response to an SENSF_REQ Command. For NFC-F the array is limited to four elements.
GRE_POLL_V[]	array of Byte Sequences	variable	Each element contains a Response to an INVENTORY_REQ Command. For NFC-V the array is limited to one element.

9.2.3 Flow Chart and Requirements

The Poller uses a fixed polling order: NFC-ACM, NFC-A, NFC-B, NFC-F, NFC-V.

If bail-out is set for a particular Technology, except for NFC-ACM, the Poller checks whether this Technology or a Technology polled for earlier has been detected. If so, the Poller stops further polling; if not, the Poller continues polling for the remaining technologies.

After polling for NFC-V, and therefore, before polling for a Proprietary Technology, the Poller always checks whether a Technology has been detected.

Figure 6 shows the processing flow for the Poller during the Technology Detection Activity.

Poll Mode – Activities

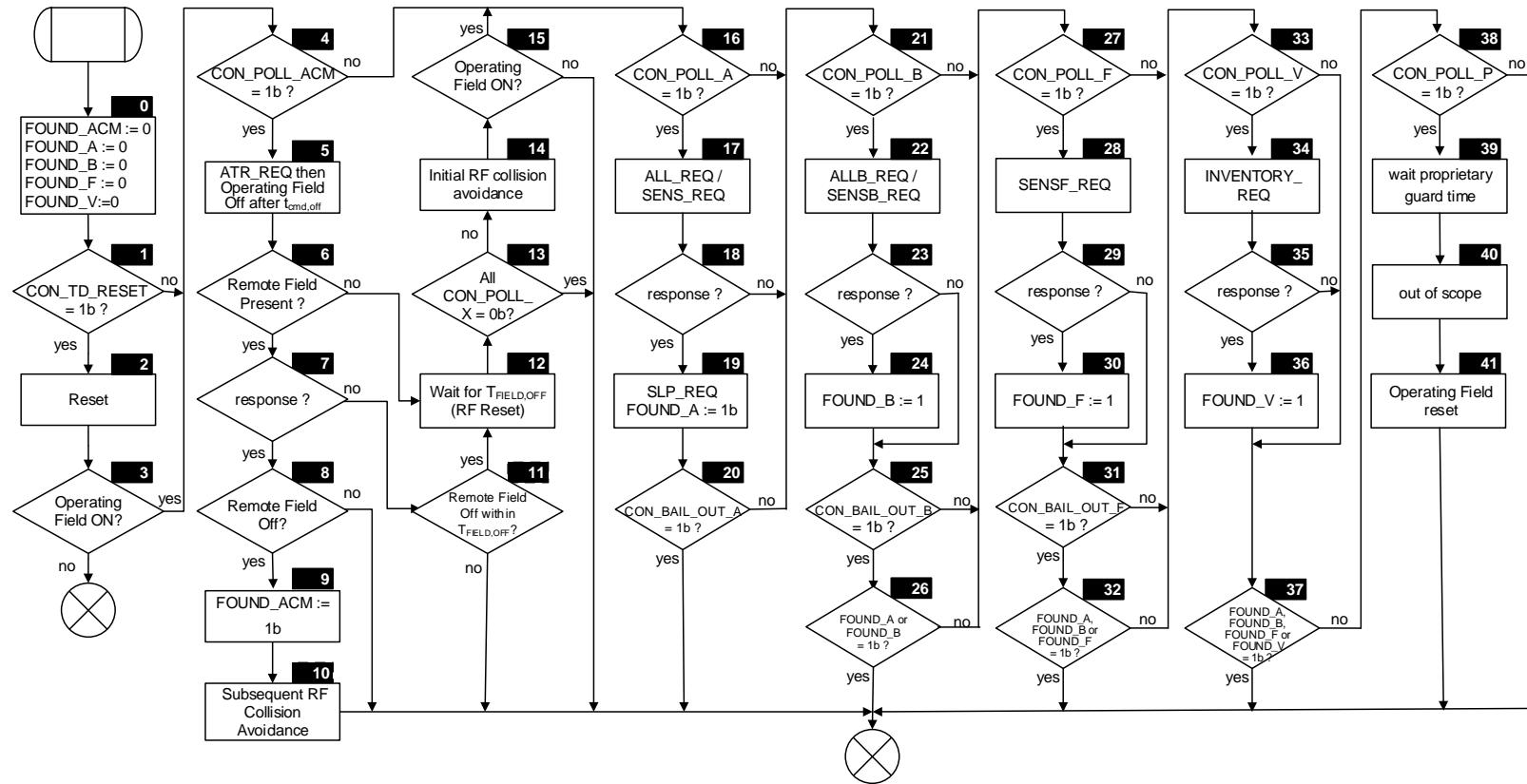


Figure 6: Technology Detection Activity – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 6.

Requirements 27: Technology Detection Activity

Poll Mode

9.2.3.1 Symbol 0:

The Poller SHALL initialize the following flags to 0b:

- FOUND_ACM
- FOUND_A
- FOUND_B
- FOUND_F
- FOUND_V.

9.2.3.2 Symbol 1:

If CON_TD_RESET is equal to 1b, the Poller SHALL proceed to Symbol 2.
Otherwise it SHALL proceed to Symbol 4.

9.2.3.3 Symbol 2:

In the following order the Poller SHALL

- Turn the Operating Field to the Operating Field Off condition (see [ANALOG])
- Wait for a time of $t_{TECHSWITCH}$
- Attempt to turn the Operating Field again to the Operating Field On condition by performing the Initial RF Collision Avoidance (see Section 0).

NOTE This symbol performs a reset of the Operating Field, which is needed when switching Technology during communication with a remote device. In this case this Activity is not the first in a Profile and is not configured for the Technology that has been used so far.

NOTE The time t_{FIELD_OFF} defines a minimum value. It is recommended that implementations use a reset time that is as close as possible to this value. If the reset time is extended significantly, the chance increases that the remote device switches to poll mode.

NOTE Due to specific NFC-F Card Emulator implementations deployed in Japan, a longer reset time is recommended for this market prior to switching to NFC-F Technology. NFC-F Card Emulator implementations in other markets do not require a longer reset time. See [PROFILES] for more information on the recommended value.

9.2.3.4 Symbol 3:

If the Operating Field is in the Operating Field On condition (as a result of the Initial RF Collision Avoidance in Symbol 2), the Poller SHALL proceed to Symbol 4.

Otherwise the Poller SHALL conclude the Technology Detection Activity.

9.2.3.5 Symbol 4:

If CON_POLL_ACM is equal to 1b, the Poller SHALL proceed to Symbol 5.

Otherwise the Poller SHALL proceed to Symbol 16.

Poll Mode

9.2.3.6 Symbol 5:

The Poller SHALL send an ATR_REQ Command using the bit rate as configured by CON_BITR_ACN, SHALL switch the Operating Field OFF after $t_{CMD,OFF}$ and then SHALL wait to detect the Remote Field generated by the Target (as defined in Section 3.4).

NOTE Before it sends an ATR_REQ Command, the Poller needs to maintain an Unmodulated Carrier for at least GT_{ACM} (as defined in requirement 7.1.1.2).

9.2.3.7 Symbol 6:

If the Poller does not detect a Remote Field within $T_{NRF,MAX}$, then it SHALL proceed to Symbol 12.

Otherwise the Poller SHALL proceed to Symbol 7.

9.2.3.8 Symbol 7:

If the Poller does not receive any Response to the ATR_REQ Command (as defined in [DIGITAL]), then the Poller SHALL proceed to Symbol 11.

Otherwise the Poller SHALL store the Response in GRE_POLL_ACN[] and SHALL proceed to Symbol 8.

9.2.3.9 Symbol 8:

If the Operating Field of the Target is not turned OFF after $t_{CMD,OFF,MAX} + T_{RFW}$ after the end of the frame the Poller SHALL conclude the Technology Detection Activity.

NOTE In this case it is recommended that the Poller change to Listen mode. Otherwise the Poller SHALL proceed to Symbol 9.

9.2.3.10 Symbol 9:

The Poller SHALL set FOUND_ACN to 1b.

9.2.3.11 Symbol 10:

The Poller SHALL perform Subsequent RF Collision avoidance and SHALL conclude the Technology Detection Activity afterwards.

9.2.3.12 Symbol 11:

The Poller SHALL wait until it detects Remote Field OFF.

If the Remote Field is not turned OFF within a duration of $t_{FIELD,OFF}$ the Poller SHALL conclude the Technology Detection Activity.

NOTE In this case it is recommended that the Poller change to Listen mode. Otherwise the Poller SHALL proceed to Symbol 12.

9.2.3.13 Symbol 12:

The Poller SHALL wait for $t_{FIELD,OFF}$ with Operating Field Off. Afterwards, the Poller SHALL proceed to Symbol 13.

Poll Mode

- 9.2.3.14** Symbol 13:
 If CON_POLL_A and CON_POLL_B and CON_POLL_F and CON_POLL_V and CON_POLL_P are equal to 0b, the Poller SHALL conclude the Technology Detection Activity.
 Otherwise Poller SHALL proceed to Symbol 14.
-
- 9.2.3.15** Symbol 14:
 The Poller SHALL perform Initial RF Collision Avoidance (see Section 0).
-
- 9.2.3.16** Symbol 15:
 If the Operating Field of the Poller is not in the Operating Field ON condition, then the Poller SHALL conclude the Technology Detection Activity.
 Otherwise the Poller SHALL proceed to Symbol 16.
-
- 9.2.3.17** Symbol 16:
 If CON_POLL_A is equal to 1b, the Poller SHALL proceed to Symbol 17.
 Otherwise the Poller SHALL proceed to Symbol 21.
-
- 9.2.3.18** Symbol 17:
 The Poller SHALL send an ALL_REQ Command or SENS_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]).
 NOTE Before sending an ALL_REQ Command or SENS_REQ Command, the Poller needs to maintain an Unmodulated Carrier for at least **GT_A** (as defined in requirement 7.1.1.2).
-
- 9.2.3.19** Symbol 18:
 If the Poller does not receive a Response to the ALL_REQ Command or SENS_REQ Commands, then Poller SHALL proceed to Symbol 21.
 Otherwise, if the Poller detects collisions in b1-b5 of Byte 1 it SHALL store the SENS_RES Response in GRE_POLL_A[] with a value that indicates bit frame SDD support (as defined in [DIGITAL]), and SHALL proceed to Symbol 19.
 Otherwise the Poller SHALL store the received SENS_RES Response in GRE_POLL_A[] and SHALL proceed to Symbol 19.
-
- 9.2.3.20** Symbol 19:
 The Poller SHALL send a SLP_REQ Command.
 The Poller SHALL set FOUND_A to 1b.
-
- 9.2.3.21** Symbol 20:
 If CON_BAIL_OUT_A is equal to 1b, the Poller SHALL conclude the Technology Detection Activity.
 Otherwise the Poller SHALL proceed to Symbol 21.
-
- 9.2.3.22** Symbol 21:
 If CON_POLL_B is equal to 1b, the Poller SHALL proceed to Symbol 22.
 Otherwise the Poller SHALL proceed to Symbol 27.
-

Poll Mode

- 9.2.3.23** Symbol 22:
The Poller SHALL send an ALLB_REQ Command or a SENS_B_REQ Command with number of slots set equal to 1 (N=1) and indicating Extended SENS_B_RES support as configured by CON_EXT_SENSB_RES and SHALL wait for a Response afterward (as defined in [DIGITAL]).
- NOTE** Before sending an ALLB_REQ Command or SENS_B_REQ Command, the Poller needs to maintain an Unmodulated Carrier for at least **GT_B** (as defined in requirement 7.1.1.2).
-

- 9.2.3.24** Symbol 23:
If the Poller does not receive a Response to the ALLB_REQ Command or SENS_B_REQ Commands, then the Poller SHALL proceed to Symbol 25.
Otherwise the Poller SHALL store the Response in GRE_POLL_B[] and SHALL proceed to Symbol 24.
-

- 9.2.3.25** Symbol 24:
The Poller SHALL set FOUND_B to 1b.
-

- 9.2.3.26** Symbol 25:
If CON_BAIL_OUT_B is equal to 1b, the Poller SHALL proceed to Symbol 26.
Otherwise the Poller SHALL proceed to Symbol 27.
-

- 9.2.3.27** Symbol 26:
If FOUND_A or FOUND_B has a value equal to 1b, the Poller SHALL conclude the Technology Detection Activity.
Otherwise the Poller SHALL proceed to Symbol 27.
-

- 9.2.3.28** Symbol 27:
If CON_POLL_F is equal to 1b, the Poller SHALL proceed to Symbol 28.
Otherwise the Poller SHALL proceed to Symbol 33.
-

- 9.2.3.29** Symbol 28:
The Poller SHALL send a SENSF_REQ Command with number of slots equal to 4 (TSN = 03h), SC = FFFFh, RC = 00h at the bit rate configured by CON_BITR and SHALL wait for a Response afterward (as defined in [DIGITAL]).
- NOTE** Before sending a SENSF_REQ Command, the Poller needs to maintain an Unmodulated Carrier for at least **GTF** (as defined in requirement 7.1.1.2).
-

- 9.2.3.30** Symbol 29:
If the Poller does not receive a Response to the SENSF_REQ Command, then the Poller SHALL proceed to Symbol 31.
Otherwise the Poller SHALL store the Response(s) in GRE_POLL_F[] and SHALL proceed to Symbol 30.
-

- 9.2.3.31** Symbol 30:
The Poller SHALL set FOUND_F to 1b.
-

Poll Mode

- 9.2.3.32** Symbol 31:
 If CON_BAIL_OUT_F is equal to 1b, the Poller SHALL proceed to Symbol 32.
 Otherwise the Poller SHALL proceed to Symbol 33.
- 9.2.3.33** Symbol 32:
 If FOUND_A or FOUND_B or FOUND_F is equal to 1b, the Poller SHALL conclude the Technology Detection Activity.
 Otherwise the Poller SHALL proceed to Symbol 33.
- 9.2.3.34** Symbol 33:
 If CON_POLL_V is equal to 1b, the Poller SHALL proceed to Symbol 34.
 Otherwise the Poller SHALL proceed to Symbol 37.
- 9.2.3.35** Symbol 34:
 The Poller SHALL send an INVENTORY_REQ Command with number of slots equal to 1, and MASK_LEN = 00h, and SHALL wait for a Response afterward (as defined in [DIGITAL]).
 NOTE Before sending an INVENTORY_REQ Command, the Poller needs to maintain an Unmodulated Carrier for at least GT_V (as defined in requirement 7.1.1.2).
- 9.2.3.36** Symbol 35:
 If the Poller does not receive a Response to the INVENTORY_REQ Command, then the Poller SHALL proceed to Symbol 37.
 Otherwise the Poller SHALL store the Response in GRE_POLL_V[] and SHALL proceed to Symbol 37.
- 9.2.3.37** Symbol 36:
 The Poller SHALL set FOUND_V to 1b.
- 9.2.3.38** Symbol 37:
 If FOUND_A or FOUND_B or FOUND_F or FOUND_V is equal to 1b, the Poller SHALL conclude the Technology Detection Activity.
 Otherwise the Poller SHALL proceed to Symbol 38.
- 9.2.3.39** Symbol 38:
 If CON_POLL_P is equal to 1, the Poller SHALL proceed to Symbol 39.
 Otherwise the Poller SHALL conclude the Technology Detection Activity.
- 9.2.3.40** Symbol 39:
 Before proceeding, the Poller SHALL maintain an Unmodulated Carrier for a guard time (as specified in requirement 7.1.1.6).
- 9.2.3.41** Symbol 40:
Processing and output parameters of Proprietary Technology is out of scope of this specification.
 Following processing, the Poller SHALL proceed to Symbol 41.
-

Poll Mode

9.2.3.42 Symbol 41:

The Poller SHALL set the Operating Field to the Operating Field Off condition (as defined in requirement 7.1.1.1).

Before proceeding, the Poller SHALL set the Operating Field to the Operating Field On condition and SHALL maintain an Unmodulated Carrier for a guard time (as specified in requirement 7.1.1.2).

9.3 Collision Resolution Activity

This section describes the Collision Resolution Activity.

9.3.1 Preconditions

The Configuration Parameters for the Collision Resolution Activity are listed in Table 11.

Table 11: Collision Resolution Activity – Configuration Parameters

Name	Format	Size	Description
CON_DEVICES_LIMIT_A	integer	1 byte	If CON_DEVICES_LIMIT_A is equal to 0: No identifier has to be resolved when a collision is detected. If CON_DEVICES_LIMIT_A is greater than 0: Number of resolved NFCID1 device identifiers beyond which the collision Resolution Process can stop resolving when collisions are still pending.
CON_DEVICES_LIMIT_B	integer	1 byte	If CON_DEVICES_LIMIT_B is equal to 0: No identifier has to be resolved when a collision is detected. If CON_DEVICES_LIMIT_B is greater than 0: Number of resolved NFCID0 device identifiers beyond which the collision Resolution Process can stop resolving when collisions are still pending.
CON_DEVICES_LIMIT_F	integer	1 byte	If CON_DEVICES_LIMIT_F is equal to 0: No additional identifier is required for collecting devices in Listen Mode. If CON_DEVICES_LIMIT_F is greater than 0: Attempt to resolve more devices if the number of resolved NFCID2 device identifiers during Technology Detection is lower than the CON_DEVICES_LIMIT_F value.
CON_DEVICES_LIMIT_V	integer	1 byte	If CON_DEVICES_LIMIT_V is equal to 0: No identifier has to be resolved when a collision is detected. If CON_DEVICES_LIMIT_V is greater than 0: Number of resolved device identifiers (UIDs) beyond which the collision Resolution Process can stop resolving when collisions are still pending.
CON_EXT_SENSB_RES	binary	1 bit	Controls whether the SENSB_REQ Command indicates support for the extended SENSB_RES byte (as defined in [DIGITAL]). 0b: Extended SENSB_RES byte is not supported 1b: Extended SENSB_RES byte is supported
CON_ANTICOLL	binary	1 bit	0b: Do not use anti-collision 1b: Use anti-collision

The Input Parameters for the Collision Resolution Activity are listed in Table 12.

Table 12: Collision Resolution Activity – Input Parameters

Name	Format	Size	Description
INT_TECH_SEL	binary	3 bits	000b: Resolve NFC-A 001b: Resolve NFC-B 010b: Resolve NFC-F 011b: Resolve NFC-V 100b: Resolve NFC-ACM

The data requested from the Greedy Collection are listed in Table 13.

Table 13: Collision Resolution Activity – Input from Greedy Collection

Name	Format	Size	Description
GRE_POLL_ACM[]	array of Byte Sequences	variable	Each element contains a Response to an ATR_REQ Command. For NFC-ACM the array is limited to one element.
GRE_POLL_F[]	array of Byte Sequences	variable	Each element contains a Response to an SENSF_REQ Command. For NFC-F the array is limited to four elements.
GRE_POLL_V[]	array of Byte Sequences	variable	Each element contains a Response to an INVENTORY_REQ Command. For NFC-V the array is limited to one element.

9.3.2 Post-conditions

The Output Parameters returned to the Resolution Process are listed in Table 14.

Table 14: Collision Resolution Activity – Output Parameters

Name	Format	Size	Description
INT_NFCIDX[m], m = 0 to M	array of identifiers	variable	Contains NFCIDX identifiers of the devices resolved. M denotes the number of devices resolved, M being less than or equal to CON_DEVICES_LIMIT_A, CON_DEVICES_LIMIT_B, CON_DEVICES_LIMIT_F or CON_DEVICES_LIMIT_V, depending on the Technology.
INT_NFCIDX_SLEEP[m], m = 0 to M	array of binary values	1 bit	Each element contains information about the sleep state of the device with the corresponding index in INT_NFCIDX: 0b: Device is not in sleep state. 1b: Device is in sleep state. M denotes the number of devices resolved.
INT_COLL_PEND	binary	1 bit	0b: No collision pending 1b: Collisions pending

NOTE For INT_NFCIDX and INT_NFCIDX_SLEEP the value of m will be 0b if collision resolution is not performed.

The data returned to the Greedy Collection are listed in Table 15.

Table 15: Collision Resolution Activity – Output into Greedy Collection

Name	Format	Size	Description
GRE_ATR_RES[]	array of Byte Sequences	variable	Response of the identified device. The single element contains a Response to an ATR_REQ Command from a Listener that uses NFC-ACM. For NFC-ACM the array is limited to one element.
GRE_SEL_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a SEL_RES Response from an NFC-A device.
GRE_SENSB_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a Response to an ALLB_REQ Command or SENSB_REQ Command from an NFC-B device.
GRE_SENSF_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a Response to an SENSF_REQ Command from an NFC-F device.
GRE_INVENTORY_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains an INVENTORY_RES Response from an NFC-V device.

9.3.3 Flow Chart (Normative)

The Collision Resolution Activity to be performed depends on the value of the INT_TECH_SEL parameter and is defined in the normative Figure 7.

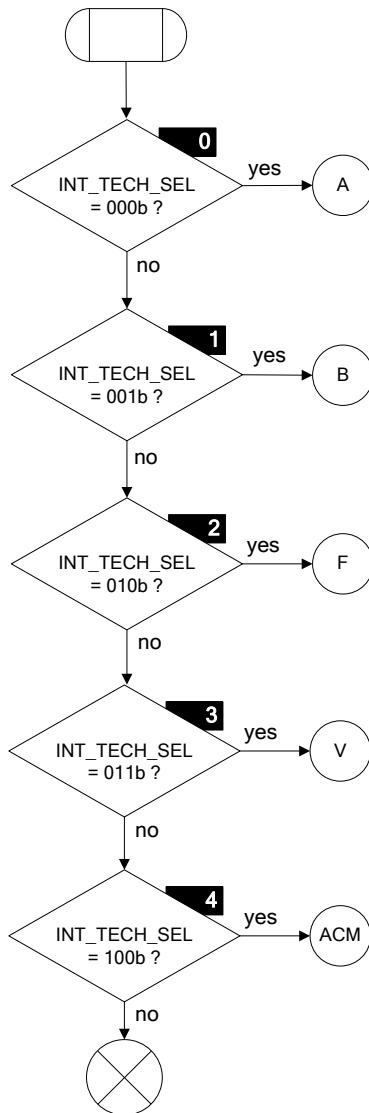


Figure 7: Collision Resolution Activity (Sheet 1, Entry) – Normative Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 7.

Requirements 28: Collision Resolution Activity

Poll Mode

- 9.3.3.1** Symbol 0:
If INT_TECH_SEL is equal to 000b, then continue with A (Flow Chart and Requirements for NFC-A).
Otherwise the Poller SHALL proceed to Symbol 1.
-

- 9.3.3.2** Symbol 1:
If INT_TECH_SEL is equal to 001b, then continue with B (Flow Chart and Requirements for NFC-B).
Otherwise the Poller SHALL proceed to Symbol 2.
-

- 9.3.3.3** Symbol 2:
If INT_TECH_SEL is equal to 010b, then continue with F (Flow Chart and Requirements for NFC-F).
Otherwise the Poller SHALL proceed to Symbol 3.
-

- 9.3.3.4** Symbol 3:
If INT_TECH_SEL is equal to 011b, then continue with V (Flow Chart and Requirements for NFC-V).
Otherwise the Poller SHALL proceed to Symbol 4.
-

- 9.3.3.5** Symbol 4:
If INT_TECH_SEL is equal to 100b, then continue with ACM (Flow Chart and Requirements for NFC-ACM).
Otherwise the Poller SHALL conclude the Collision Resolution Activity.
-

9.3.4 Flow Chart and Requirements for NFC-A

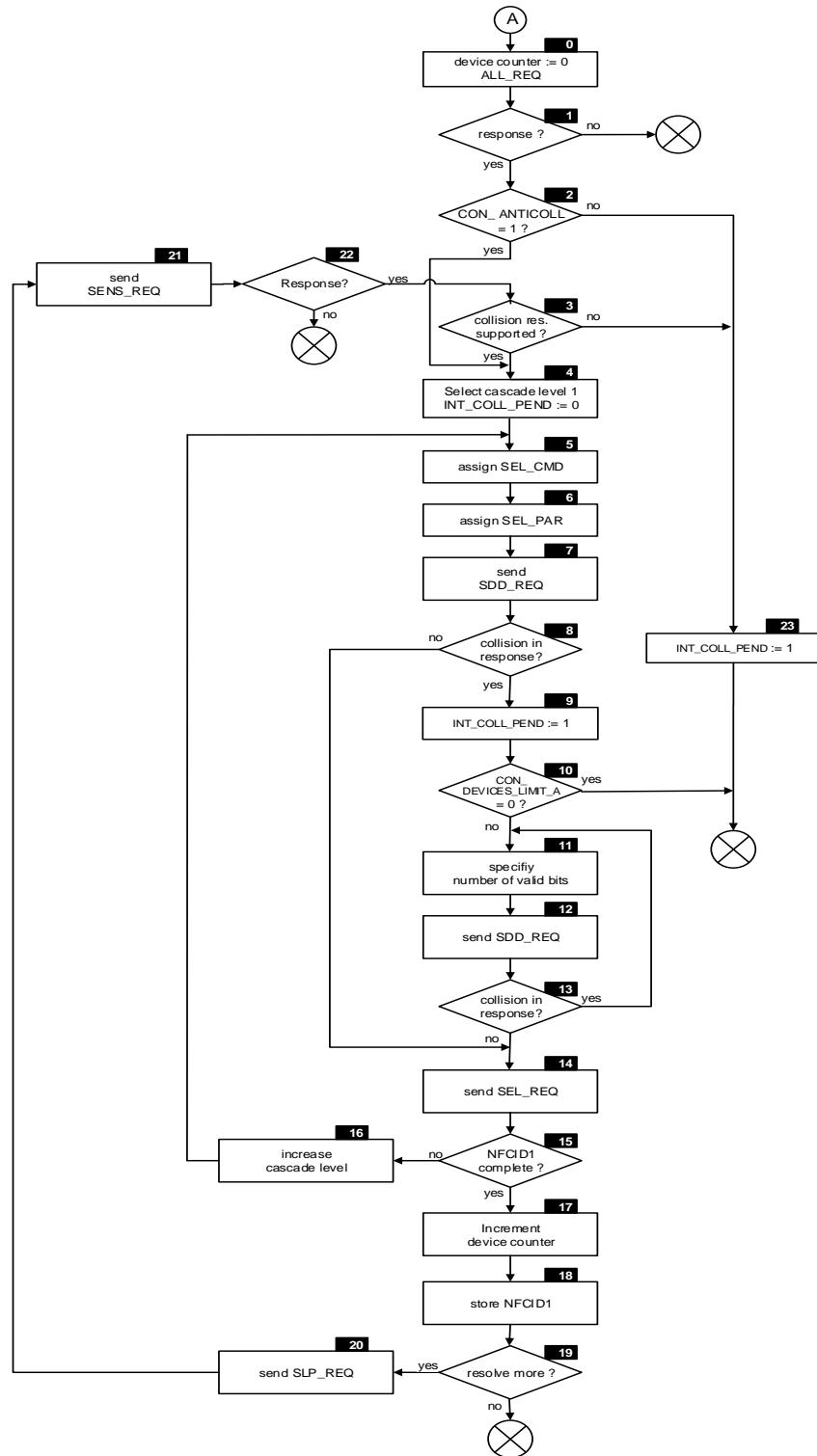
The purpose of the NFC-A-related part of the Collision Resolution Activity is to identify a device within range that has activated support for NFC-A (subset). The algorithm works as follows:

The algorithm selects one device after another. Every time a collision is detected, the algorithm continues with the valid bits of the NFCID1 CLn followed by a single bit that is set to any value. In this way multiple devices can be identified by selecting all cascade levels of one device before restarting the algorithm to select the next device. Before restarting the algorithm, the device identified is sent to **SLEEP_A** State to exclude it from the remaining collision resolution process.

The Poller can be configured to shorten the process by using the CON_DEVICES_LIMIT_A Configuration Parameters. The CON_DEVICES_LIMIT_A is used to conclude the Collision Resolution Activity after identification of a set number of devices, even if collisions are still pending.

If the CON_DEVICES_LIMIT_A is set to zero, then collision detection only is performed. That is, if a collision is detected, the Poller concludes the Collision Resolution Activity indicating a collision without identifying any device. During the Collision Resolution Activity, the number of collisions in each SENS_RES Response will diminish. This information can be used by the Resolution Process to determine the SENS_RES Response for some of the resolved devices.

Figure 8 describes the NFC-A-related part of the Collision Resolution Activity.


Figure 8: Collision Resolution Activity (Sheet 2, connector A, NFC-A) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 8.

Requirements 29: Collision Resolution Activity – NFC-A

Poll Mode

9.3.4.1 Symbol 0:

The Poller SHALL assign a parameter containing the device counter and SHALL initialize the parameter with a value of 0. The Poller SHALL initialize INT_NFCIDX_SLEEP[] with 0b.

The Poller SHALL send an ALL_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]).

9.3.4.2 Symbol 1:

If the Poller does not receive a Response to the ALL_REQ Command, then Poller SHALL clear INT_NFCIDX, set INT_COLL_PEND to 0b and conclude the Collision Resolution Activity.

Otherwise, if the Poller SHALL proceed to Symbol 2.

9.3.4.3 Symbol 2:

If CON_ANTICOLL is equal to 1b, the Poller SHALL proceed to Symbol 4.

Otherwise the Poller proceeds with Symbol 23.

9.3.4.4 Symbol 3:

If the SENS_RES Response is a Valid Response and does not indicate bit frame SDD support (as defined in [DIGITAL]), the Poller SHALL proceed to Symbol 23.

Otherwise the Poller SHALL proceed to Symbol 4.

9.3.4.5 Symbol 4:

The Poller SHALL select SDD cascade level 1 and set INT_COLL_PEND to 0b.

9.3.4.6 Symbol 5:

The Poller SHALL assign SEL_CMD with the code for the selected SDD cascade level.

9.3.4.7 Symbol 6:

The Poller SHALL set SEL_PAR to the value of 20h, indicating that no data bits are following.

9.3.4.8 Symbol 7:

The Poller SHALL send the SDD_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]). [DIGITAL] specifies the coding of the SDD_REQ Command.

9.3.4.9 Symbol 8:

If the Poller detects a Collision in the Response to the SDD_REQ Command, the Poller SHALL proceed to Symbol 9.

Otherwise the Poller SHALL proceed to Symbol 14.

9.3.4.10 Symbol 9:

The Poller SHALL set INT_COLL_PEND to 1b to indicate a pending collision.

Poll Mode

- 9.3.4.11** Symbol 10:
If CON_DEVICES_LIMIT_A is equal to 0, then the Poller SHALL conclude the Collision Resolution Activity (i.e., the Poller is configured to perform collision detection only).
Otherwise the Poller SHALL proceed to Symbol 11.
- 9.3.4.12** Symbol 11:
The Poller SHALL set SEL_PAR to a value that specifies the number of valid bits of NFCID1 CL_n. The valid bits are part of the NFCID1 CL_n that was received before a collision occurred, followed by a single bit that is set to any value (i.e., the position of the first collision in the Response to the previous SDD_REQ Command can take the value 1b or 0b).
- 9.3.4.13** Symbol 12:
The Poller SHALL send the SDD_REQ Command, including the data bits indicated by SEL_PAR, and SHALL wait for a Response afterward (as defined in [DIGITAL]).
- 9.3.4.14** Symbol 13:
If the Poller detects a Collision in the Response to the SDD_REQ Command, the Poller SHALL proceed to Symbol 11.
Otherwise the Poller SHALL proceed to Symbol 14.
- 9.3.4.15** Symbol 14:
The Poller SHALL send the SEL_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]). [DIGITAL] specifies the coding of the SEL_REQ Command.
- 9.3.4.16** Symbol 15:
The Poller SHALL check the Cascade bit of the SEL_RES Response.
If the Cascade bit indicates that NFCID1 is complete, then the Poller SHALL proceed to Symbol 17.
Otherwise the Poller SHALL proceed to Symbol 16.
- 9.3.4.17** Symbol 16:
The Poller SHALL increase the cascade level.
The Poller SHALL proceed to Symbol 5.
- 9.3.4.18** Symbol 17:
The Poller SHALL increment the device counter by 1.
- 9.3.4.19** Symbol 18:
The Poller SHALL store the SEL_RES Response in GRE_SEL_RES[] and SHALL store the NFCID1 identifier in INT_NFCIDX[device counter-1].
-

Poll Mode

- 9.3.4.20** Symbol 19:
 If the device counter is lower than CON_DEVICES_LIMIT_A and either INT_COLL_PEND is equal to 1b or the Poller wants to check for further devices in the operating volume, then the Poller SHALL proceed to Symbol 20.
 Otherwise the Poller SHALL conclude the Collision Resolution Activity.
-
- 9.3.4.21** Symbol 20:
 The Poller SHALL send a SLP_REQ Command to put the device with identifier INT_NFCIDX[device counter – 1] in the **SLEEP_A** State and SHALL wait for a Response afterward (as defined in [DIGITAL]).
 The Poller SHALL memorize the information that a SLP_REQ Command has been sent to the device by setting INT_NFCIDX_SLEEP[device counter – 1] equal to 1b.
-
- 9.3.4.22** Symbol 21:
 The Poller SHALL send the SENS_REQ Command and SHALL wait for the SENS_RES Response afterward (as specified in [DIGITAL]).
-
- 9.3.4.23** Symbol 22:
 If the Poller does not receive a Response to the SENS_REQ Command, then the Poller SHOULD conclude the Collision Resolution Activity. Before concluding, the Poller SHALL set INT_COLL_PEND to 0b.
 Otherwise, the Poller SHALL proceed to Symbol 3.
-
- 9.3.4.24** Symbol 23:
 The Poller SHALL set INT_COLL_PEND to 1b and SHALL conclude the Collision Resolution Activity.
-

9.3.5 Flow Chart and Requirements for NFC-B

The purpose of the NFC-B-related part of the Collision Resolution Activity is to identify a device within range that has activated support for NFC-B. The algorithm works as follows:

If the Technology Detection resulted in a Valid SENSB_RES Response (i.e., no collisions), then the Poller extracts the identifier and stores it in INT_NFCIDX[]. If CON_DEVICES_LIMIT_B is equal to 1 or there is just one device resolved, then the device is left in the **READY_B_DECL** State.

If the Technology Detection resulted in an invalid SENSB_RES Response (i.e., collisions), then the Poller polls with the number of timeslots set to 1. The Poller saves each Valid Response to a SENSB_REQ Command or SLOT_MARKER Command in GRE_SENSB_RES[]. Each Valid Response results in an identifier that is stored in INT_NFCIDX[] and each device identified is subsequently put in the **SLEEP_B** State, except the last resolved device (it stays in the **READY_B_DECL** State).

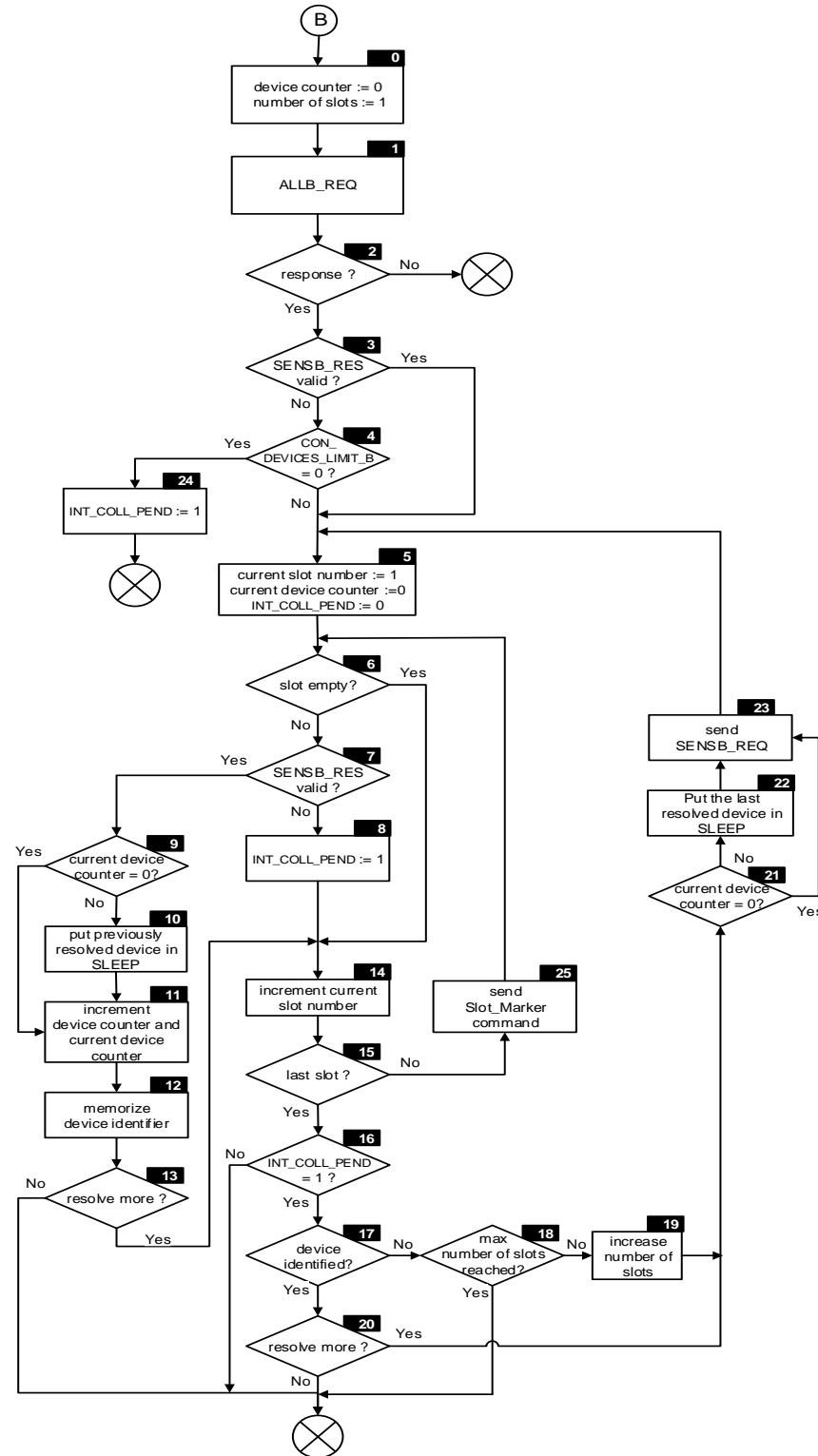
As long as collisions occur and no device has been identified yet, the Poller increments the number of time slots and sends new SENSB_REQ Commands. If there are still collisions after having completed the collision resolution with the maximum number of time slots, no further attempt is made to isolate the identifiers.

When at least one device has already been identified, the number of slots is not further incremented.

The Poller can be configured to shorten the process by using the CON_DEVICES_LIMIT_B parameter. The parameter is used to conclude the Collision Resolution Activity after identification of a set number of devices, even if collisions are still pending.

If this parameter is set to zero, then collision detection only is performed. That is, if a collision is detected, the Poller concludes the Collision Resolution Activity.

Figure 9 describes the NFC-B-related part of the Collision Resolution Activity.


Figure 9: Collision Resolution Activity (Sheet 3, connector B, NFC-B) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 9.

Requirements 30: Collision Resolution Activity – NFC-B

Poll Mode

9.3.5.1 Symbol 0:

The Poller SHALL assign a parameter containing the device counter and SHALL initialize this parameter with 0.

The Poller SHALL assign a parameter containing the number of slots and SHALL initialize this parameter with 1.

The Poller SHALL initialize INT_NFCIDX_SLEEP[] with 0b.

9.3.5.2 Symbol 1:

The Poller SHALL send an ALLB_REQ Command with number of slots set equal to 1 (N=1) and indicating Extended SENSB_RES support as configured by CON_EXT_SENSB_RES and SHALL wait for a Response afterward (as defined in [DIGITAL]).

9.3.5.3 Symbol 2:

If the Poller does not receive a Response to the ALLB_REQ Command, then the Poller SHALL clear INT_NFCIDX, set INT_COLL_PEND to 0b and conclude the Collision Resolution Activity.

Otherwise the Poller SHALL proceed to Symbol 3.

9.3.5.4 Symbol 3:

If the SENSB_RES Response received in Symbol 1 is Valid, then the Poller SHALL proceed to Symbol 5.

Otherwise the Poller SHALL proceed to Symbol 4.

9.3.5.5 Symbol 4:

If CON_DEVICES_LIMIT_B is equal to 0 (i.e., the Poller is configured to perform collision detection only), then the Poller SHALL proceed to Symbol 24.

Otherwise the Poller SHALL proceed to Symbol 5.

9.3.5.6 Symbol 5:

The Poller SHALL assign a parameter containing the current slot number and SHALL initialize this parameter with 1.

The Poller SHALL assign a parameter containing the current device counter and SHALL initialize this parameter with 0.

The Poller SHALL initialize INT_COLL_PEND with 0b.

9.3.5.7 Symbol 6:

If the Poller did not receive a Response in the slot corresponding to the current slot number, then the Poller SHALL proceed to Symbol 14.

Otherwise the Poller SHALL proceed to Symbol 7.

Poll Mode

- 9.3.5.8** Symbol 7:
 If the last SENSB_RES Response that the Poller has memorized is Valid, then the Poller SHALL proceed to Symbol 9.
 Otherwise the Poller SHALL proceed to Symbol 8.
- 9.3.5.9** Symbol 8:
 The Poller SHALL set INT_COLL_PEND to 1b.
 The Poller SHALL proceed to Symbol 14.
- 9.3.5.10** Symbol 9:
 If the current device counter is equal to 0, then the Poller SHALL proceed to Symbol 11.
 Otherwise the Poller SHALL proceed to Symbol 10.
- 9.3.5.11** Symbol 10:
 The Poller SHALL send a SLPB_REQ Command to put the previously resolved device (INT_NFCIDX[device_counter-1]) in the **SLEEP_B** State and SHALL wait for a Response afterward (as defined in [DIGITAL]).
 The Poller SHALL set INT_NFCIDX_SLEEP[device counter-1] to 1b.
- 9.3.5.12** Symbol 11:
 The Poller SHALL increment the device counter and the current device counter.
- 9.3.5.13** Symbol 12:
 The Poller SHALL store the NFCID0 identifier to INT_NFCIDX[device counter-1] and the last received SENSB_RES Command in GRE_SENSB_RES[].
- 9.3.5.14** Symbol 13:
 If the device counter is lower than CON_DEVICES_LIMIT_B and the number of slots is higher than 1, then the Poller SHALL proceed to Symbol 14.
 Otherwise the Poller SHALL conclude the Collision Resolution Activity.
- 9.3.5.15** Symbol 14:
 The Poller SHALL increment the current slot number, indicating the current slot in which to receive SENSB_RES Responses.
- 9.3.5.16** Symbol 15:
 If the current slot number is equal to the last slot, then the Poller SHALL proceed to Symbol 16.
 Otherwise the Poller SHALL proceed to Symbol 25.
-

Poll Mode

9.3.5.17 Symbol 16:

If INT_COLL_PEND is equal to 1b, then the Poller SHALL proceed to Symbol 17.

Otherwise the Poller SHALL conclude the Collision Resolution Activity.

9.3.5.18 Symbol 17:

If subsequent to the last SENSB_REQ Command, the Poller resolved an identifier of a responding device (i.e., the identifier of the responding device has been memorized), then the Poller SHALL proceed to Symbol 20.

Otherwise (i.e., no identifier was resolved), the Poller SHALL proceed to Symbol 18.

9.3.5.19 Symbol 18:

If the number of slots coded within NI is equal to the maximum NI value supported by the Poller, then the Poller SHALL conclude the Collision Resolution Activity.

Otherwise the Poller SHALL proceed to Symbol 19.

NI is specified in [DIGITAL] within the SENSB_REQ Command.

9.3.5.20 Symbol 19:

The Poller SHALL increase the number of slots to the next value allowed. The values allowed for the number of slots are specified in [DIGITAL], within the SENSB_REQ Command.

The Poller SHALL proceed to Symbol 21.

9.3.5.21 Symbol 20:

If the device counter is lower than CON_DEVICES_LIMIT_B, then the Poller SHALL proceed to Symbol 21.

Otherwise the Poller SHALL conclude the Collision Resolution Activity.

9.3.5.22 Symbol 21:

If the current device counter is equal to 0, then the Poller SHALL proceed to Symbol 23.

Otherwise the Poller SHALL proceed to Symbol 22.

9.3.5.23 Symbol 22:

The Poller SHALL send a SLPB_REQ Command to put the last resolved device (INT_NFCIDX[device_counter-1]) in the **SLEEP_B** State and SHALL wait for a Response afterward (as defined in [DIGITAL]).

The Poller SHALL set INT_NFCIDX_SLEEP[device counter-1] equal to 1b.

Poll Mode

9.3.5.24 Symbol 23:

The Poller SHALL send a SENSB_REQ Command indicating Extended SENSB_RES support as configured by CON_EXT_SENSB_RES and SHALL wait for the SENSB_RES Response afterward (as specified in [DIGITAL]).

The Poller SHALL proceed to Symbol 5.

9.3.5.25 Symbol 24:

The Poller SHALL set INT_COLL_PEND to 1b and SHALL conclude the Collision Resolution Activity.

9.3.5.26 Symbol 25:

The Poller SHALL send a SLOT_MARKER Command indicating the current slot and SHALL wait for a Response afterward (as defined in [DIGITAL]).

The Poller SHALL proceed to Symbol 6.

9.3.6 Flow Chart and Requirements for NFC-F

The purpose of the NFC-F-related part of the Collision Resolution Activity is to identify a device that has activated support for NFC-F (subset). The algorithm works as follows:

The Poller retrieves the number of devices that already have been identified. If this number is lower than the value of CON_DEVICES_LIMIT_F, then the Poller polls again by sending a SENSF_REQ Command with the maximum number of time slots set.

Figure 10 describes the NFC-F related part of the Collision Resolution Activity.

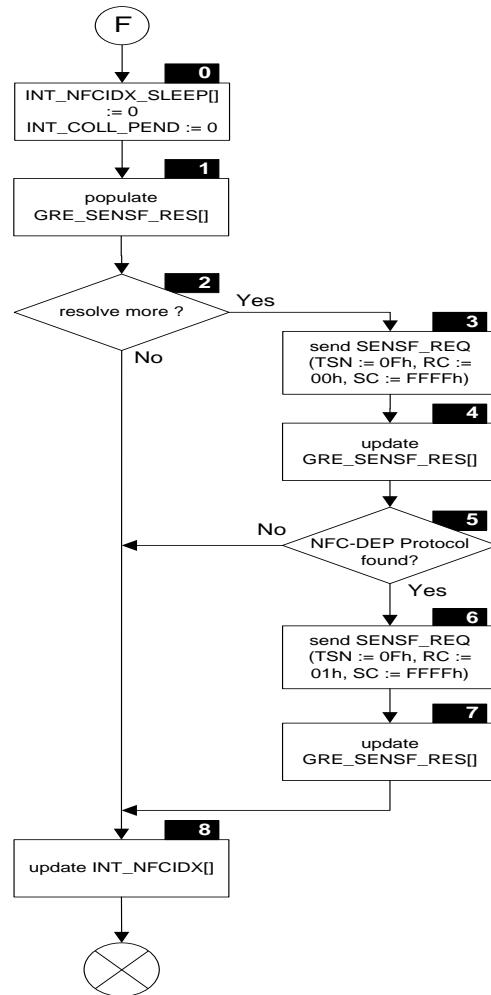


Figure 10: Collision Resolution Activity (Sheet 4, connector F, NFC-F) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 10.

Requirements 31: Collision Resolution Activity – NFC-F

Poll Mode

9.3.6.1 Symbol 0:

The Poller SHALL set INT_COLL_PEND to 0b and SHALL initialize INT_NFCIDX_SLEEP[] with 0b.

The Poller SHALL remove all entries from GRE_SENSF_RES[].

9.3.6.2 Symbol 1:

The Poller SHALL assign a parameter device counter and SHALL initialize this parameter with 0.

The Poller SHALL read GRE_POLL_F[], which contains the SENSF_RES Response(s) to the preceding SENSF_REQ Commands. For each Valid SENSF_RES Response the Poller SHALL increment the device counter by 1, it SHALL copy each Valid SENSF_RES Response contained in GRE_POLL_F[] into GRE_SENSF_RES[].

9.3.6.3 Symbol 2:

If the value of device counter (the number of Valid SENSF_RES Responses retrieved from Greedy Collection) is lower than the value of CON_DEVICES_LIMIT_F, the Poller SHALL proceed to Symbol 3.

Otherwise the Poller SHALL proceed to Symbol 8.

9.3.6.4 Symbol 3:

The Poller SHALL send a SENSF_REQ Command with TSN set to 0Fh, RC set to 00h and SC set to FFFFh and SHALL wait for a Response afterward (as defined in [DIGITAL]).

9.3.6.5 Symbol 4:

The Poller SHALL check for any Valid SENSF_RES Response(s) received during processing of Symbol 3 whether an entry with the same NFCID2 already exists in GRE_SENSF_RES[], and if not, store the SENSF_RES Response in GRE_SENSF_RES[].

9.3.6.6 Symbol 5:

If the NFCID2 of at least one SENSF_RES Response received in Symbol 3 indicates support for the NFC-DEP Protocol, it SHALL proceed to Symbol 6.

Otherwise the Poller SHALL proceed to Symbol 8.

9.3.6.7 Symbol 6:

The Poller SHALL send a SENSF_REQ Command with TSN set to 0Fh, RC set to 01h and SC set to FFFFh and SHALL wait for a Response afterward (as defined in [DIGITAL]).

9.3.6.8 Symbol 7:

The Poller SHALL check for any Valid SENSF_RES Response(s) received during processing of Symbol 4 whether an entry with the same NFCID2 already exists in GRE_SENSF_RES[], and if not, store the SENSF_RES Response in GRE_SENSF_RES[]

Poll Mode

9.3.6.9 Symbol 8:

The Poller SHALL remove all entries from INT_NFCIDX[]. Then, the Poller SHALL extract the NFCID2 for each entry in GRE_SENSF_RES[] and SHALL store NFCID2 in INT_NFCIDX[].

Afterward, the Poller SHALL conclude the Collision Resolution Activity.

9.3.7 Flow Chart and Requirements for NFC-V

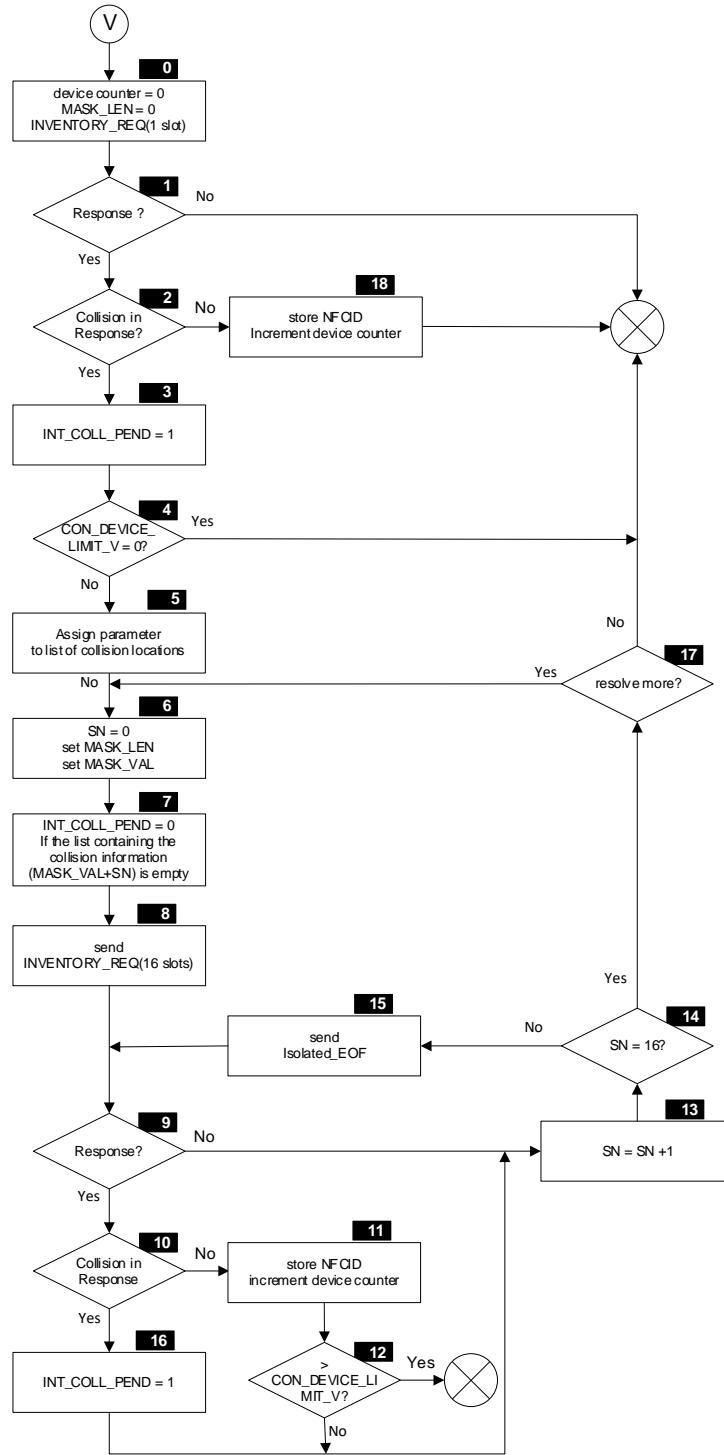
The purpose of the NFC-V related part of the Collision Resolution Activity is to identify a device within range that has activated support for NFC-V (subset). The algorithm works as follows:

The algorithm detects one device after the other. Every time a collision is detected, the algorithm continues filtering devices with collisions by extending the mask by a nibble corresponding to the current slot number for the following INVENTORY_REQ Command. In this way multiple devices can be identified.

The Poller can be configured to shorten the process by using the CON_DEVICES_LIMIT_V Configuration Parameter. CON_DEVICES_LIMIT_V is used to conclude the Collision Resolution Activity after identification of a number of devices, even if collisions are still pending.

If the CON_DEVICES_LIMIT_V is set to zero, then only the collision detection is performed. That is, if a collision is detected, the Poller concludes the Collision Resolution Activity indicating a collision without identifying any device.

Figure 11 describes the NFC-V-related part of the Collision Resolution Activity.


Figure 11: Collision Resolution Activity (Sheet 5, connector V, NFC-V) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 11.

Requirements 32: Collision Resolution Activity – NFC-V

Poll Mode

9.3.7.1 Symbol 0:

The Poller SHALL assign a parameter containing the device counter and SHALL initialize the parameter with a value of 0. The Poller SHALL initialize INT_NFCIDX_SLEEP[] with 0b.

The Poller SHALL set INT_COLL_PEND to 0b.

The Poller SHALL send an INVENTORY_REQ Command with 1 Slot Mode and a MASK_LEN of 0 and an empty MAS_VAL, and SHALL wait for a Response afterwards (as defined in [DIGITAL]).

9.3.7.2 Symbol 1:

If the Poller does not receive a Response to the INVENTORY_REQ Command, then the Poller SHALL clear INT_NFCIDX, and SHALL conclude the Collision Resolution Activity.

Otherwise the Poller SHALL proceed to Symbol 2.

9.3.7.3 Symbol 2:

If the INVENTORY_RES Response is a Correct Frame, the Poller SHALL proceed to Symbol 18.

Otherwise the Poller detects a Collision in the Response to the INVENTORY_REQ Command and SHALL proceed to Symbol 3.

9.3.7.4 Symbol 3:

The Poller SHALL set INT_COLL_PEND to 1b.

9.3.7.5 Symbol 4:

If CON_DEVICES_LIMIT_V is not equal to 0, the Poller SHALL proceed to Symbol 5.

Otherwise the Poller SHALL conclude the Collision Resolution Activity.

9.3.7.6 Symbol 5:

The Poller SHALL assign a parameter to contain a list of collision information and SHALL either

- initialize the list to be empty
- add the collision information acquired in Symbol 2

Subsequently, the Poller SHALL proceed to Symbol 6

Poll Mode

9.3.7.7 Symbol 6:

The Poller SHALL assign a parameter SN to hold the slot number, and SHALL initialize this parameter with the value ‘00h’.

The Poller SHALL assign a parameter MASK_VAL to hold the mask bits to be sent in further INVENTORY_REQ Commands.

If the list containing collision information (MASK_VAL+SN) is empty then the Poller SHALL initialize the parameter MASK_VAL to empty. Otherwise the Poller SHALL generate the current MASK_VAL from the MASK_VAL and the SN value of a previous collision which is stored in the list containing the collision information.

If the slotted collision resolution mechanism is used, the entry associated to the current MASK_VAL and MASK_LEN SHALL be removed from the list containing the collision information.

The Poller SHALL assign the number of bits of the current MASK_VAL to the MASK_LEN parameter.

NOTE The process of selecting which of the stored collision information will be processed in the restarting 16 slot loop is implementation dependent and outside of the scope of this specification.

9.3.7.8 Symbol 7:

If the list containing collision information (MASK_VAL+SN) is empty, the Poller SHALL set INT_COLL_PEND to 0b.

Otherwise the Poller SHALL leave the value unchanged.

9.3.7.9 Symbol 8:

The Poller SHALL send the INVENTORY_REQ Command with 16 slot mode and the current values of MASK_VAL and MASK_LEN, and SHALL wait for a Response afterwards (as defined in [DIGITAL]). [DIGITAL] specifies the coding of the INVENTORY_REQ Command.

9.3.7.10 Symbol 9:

If the Poller does not receive a Response to the INVENTORY_REQ Command or isolated EOF frame, then the Poller SHALL proceed to Symbol 13.

Otherwise the Poller SHALL proceed to Symbol 10.

9.3.7.11 Symbol 10:

If the INVENTORY_RES Response is a Correct Frame, the Poller SHALL proceed to Symbol 11.

Otherwise the Poller detects a Collision in the Response to the INVENTORY_REQ Command and SHALL proceed to Symbol 16.

9.3.7.12 Symbol 11:

If the INVENTORY_RES Response is a Valid Response, the Poller SHALL increment the device counter by 1, SHALL store the INVENTORY_RSP Response in GRE_INV_RSP[] and SHALL store the UID in INT_NFCIDX[device counter-1].

Poll Mode

- 9.3.7.13** Symbol 12:
If the device counter reached the value of CON_DEVICES_LIMIT_V then the Poller SHALL conclude the Collision Resolution activity.
Otherwise the Poller SHALL proceed to Symbol 13.
-
- 9.3.7.14** Symbol 13:
The Poller SHALL increment the slot number SN counter by 1.
-
- 9.3.7.15** Symbol 14:
If SN is equal to 16, the Poller SHALL proceed to Symbol 17.
Otherwise the Poller SHALL proceed to Symbol 15.
-
- 9.3.7.16** Symbol 15:
The Poller SHALL send an Isolated EOF frame and SHALL wait for a Response afterward (as defined in [DIGITAL]). The Poller SHALL proceed to Symbol 9.
-
- 9.3.7.17** Symbol 16:
The Poller SHALL set INT_COLL_PEND to 1b, SHALL add the collision information (MASK_VAL + SN) to the list containing the collision information and SHALL proceed to Symbol 13.
-
- 9.3.7.18** Symbol 17:
If INT_COLL_PEND is equal to 1b, then the Poller SHALL proceed to Symbol 6.
Otherwise the Poller SHALL conclude the Collision Resolution Activity.
-
- 9.3.7.19** Symbol 18:
If the INVENTORY_RES Response is a Valid Response, the Poller SHALL increment the device counter by 1, SHALL store the INVENTORY_RSP Response in GRE_INV_RSP[] and SHALL store the UID in INT_NFCIDX[device counter-1].
Afterwards, the Poller SHALL conclude the Collision Resolution Activity.
-

9.3.8 Flow Chart and Requirements for NFC-ACM

The purpose of the NFC-ACM related part of the Collision resolution Activity is to identify a device in range that has activated support for Active Communication Mode.

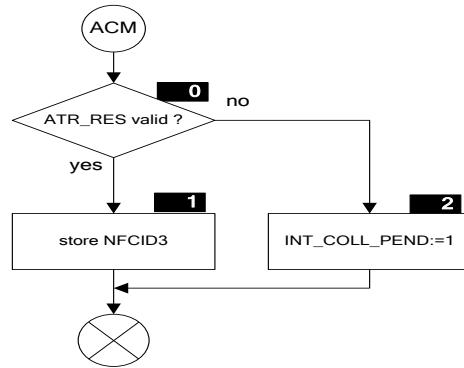


Figure 12: Collision Resolution Activity (Sheet 6, connector ACM, NFC-ACM) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 12.

Requirements 33: Collision Resolution Activity – NFC-ACM

Poll Mode

9.3.8.1 Symbol 0:

The Poller SHALL set INT_COLL_PEND to 0b.

If the ATR_RES Response received by the Poller is valid, the Poller SHALL proceed to Symbol 1.

Otherwise the Poller SHALL proceed to Symbol 2.

9.3.8.2 Symbol 1:

The Poller SHALL store the ATR_RES Response in GRE_ATR_RES[device_counter-1] and the NFCID3 identifier in INT_NFCIDX[device_counter-1].

Afterwards, the Poller SHALL conclude the Collision Resolution Activity.

9.3.8.3 Symbol 2:

The Poller SHALL set INT_COLL_PEND to 1b to indicate a pending collision and SHALL conclude the Collision Resolution Activity.

9.4 Device Activation Activity

This section describes the Device Activation Activity.

The Device Activation Activity activates one device out of the set of devices identified during Technology Detection Activity and Collision Resolution Activity. The Resolution Process decides which device to activate.

9.4.1 Preconditions

The Configuration Parameters for the Device Activation Activity are listed in Table 16.

Table 16: Device Activation Activity – Configuration Parameters

Name	Format	Size	Description
CON_EXT_SEN SB_RES	binary	1 bit	Controls whether the SENSB_REQ Command indicates support for the extended SENSB_RES byte (as defined in [DIGITAL]). 0b: Extended SENSB_RES byte is not supported 1b: Extended SENSB_RES byte is supported
CON_ATR	Byte Sequence	4 bytes	ATR_REQ Command parameter. Contains Byte 13 to 16 of the ATR_REQ Command (as defined in [DIGITAL]).
CON_GB	Byte Sequence	n bytes	General bytes of the ATR_REQ Command or Higher Layer INF of the ATTRIB Command See [DIGITAL] for information on Byte 17+n of the ATR_REQ Command and [DIGITAL]Byte 10+n of the ATTRIB Command. For the ATR_REQ Command these bytes contain the General Bytes ($G_T0 \dots G_Tn$) as information for LLCP. For the ATTRIB Command these bytes contain High Layer INF.
CON_RATS	hexadecimal	1 byte	RATS Command Parameters [DIGITAL] (Byte 2 of RATS Command) specifies the coding of Byte 1.
CON_ATTRIB	Byte Sequence	4 bytes	ATTRIB Command Parameters. Contains Byte 6 to 9 of the ATTRIB Command (as defined in [DIGITAL]).
CON_BITR_NF C_DEP	integer	1 byte	Desired bit rate for NFC-DEP: <ul style="list-style-type: none">• 0: maintain the bit rate• 1: 106 kbps• 2: 212 kbps• 3: 424 kbps

The Input Parameters for the Device Activation Activity are listed in Table 17.

Table 17: Device Activation Activity – Input Parameters

Name	Format	Size	Description
INT_TECH_SEL	binary	3 bits	Technology to activate: <ul style="list-style-type: none">• 000b: NFC-A• 001b: NFC-B• 010b: NFC-F• 011b: NFC-V• 100b: NFC-ACM
INT_INDEX	integer	1 byte	Contains the index to the identifier in INT_NFCIDX of the device to be activated
INT_NFCIDX[m], m = 1 to M	array of identifiers	variable	Contains the NFCIDX identifiers of the devices resolved. M denotes the number of devices resolved, M being less than or equal to one of: CON_DEVICES_LIMIT_A CON_DEVICES_LIMIT_B CON_DEVICES_LIMIT_F CON_DEVICES_LIMIT_V depending on the Technology.
INT_NFCIDX_SL EEP[m], m = 1 to M	binary	1 bit	Each element contains information about the sleep state of the device with the corresponding index in INT_NFCIDX: 0b: Device is not in sleep state. 1b: Device is in sleep state . M denotes the number of devices resolved
INT_PROTOCOL	binary	3 bits	Protocol of device to be activated: <ul style="list-style-type: none">• 000b: Use NFC-DEP• 001b: Use ISO-DEP• 011b: Use Type 2 Tag Platform• 100b: Use Type 3 Tag Platform• 101b: Use Type 5 Tag Platform

NOTE Use of Type 4 Tag Platform is covered by use of ISO-DEP.

The data requested from the Greedy Collection are listed in Table 18.

Table 18: Device Activation Activity – Input from Greedy Collection

Name	Format	Size	Description
GRE_SEL_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a SEL_RES Response from an NFC-A device.
GRE_SENSB_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a Response to an ALLB_REQ Command or SENSB_REQ Command from an NFC-B device.
GRE_SENSF_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains a Response to an SENSF_REQ Command from an NFC-F device.
GRE_INVENTORY_RES[]	array of Byte Sequences	variable	Response of each identified device. Each element contains an INVENTORY_RES Response from an NFC-V device.

9.4.2 Post-conditions

The Output Parameters returned to the Resolution Process are listed in Table 19.

Table 19: Device Activation Activity – Output Parameters

Name	Format	Size	Description
INT_INDEX	integer	1 byte	Index to the identifier in INT_NFCIDX of the device activated.
INT_NFCIDX_SLEEP[m], m = 1 to M	array of binary values	1 bit	Each element contains information about the sleep state of the device with the corresponding index in INT_NFCIDX: 0b: Device is not in sleep state. 1b: Device is in sleep state. M denotes the number of devices resolved
INT_DX_TECHNOLOGY	binary	3 bits	Technology that has been selected for use during data exchange <ul style="list-style-type: none"> • 000b: NFC-A • 001b: NFC-B • 010b: NFC-F • 011b: NFC-V • 100b: NFC-ACM
INT_DX_BIT_RATE	integer	1 byte	Current bit rate in case of NFC-DEP activation: <ul style="list-style-type: none"> • 0: unchanged bit rate • 1: 106 kbps • 2: 212 kbps • 3: 424 kbps

The data returned to the Greedy Collection are listed in Table 20.

Table 20: Device Activation Activity – Output into Greedy Collection

Name	Format	Size	Description
GRE_ATR	Byte Sequence	≥ 17 bytes	ATR_RES Response of device activated
GRE_RATS	Byte Sequence	≥ 2 bytes	RATS Response of device activated
GRE_ATTRIB	Byte Sequence	≥ 1 byte	ATTRIB Response of device activated

NOTE There is no Greedy Collection for the Type 2 Tag Platform.

NOTE For the Type 2 Tag Platform the outcome of the device activation (by means of a Valid READ or WRITE Command) is part of the Data Exchange Activity.

9.4.3 Flow Chart (Normative)

The Device Activation Activity to be performed depends on the value of the INT_TECH_SEL parameter and is defined in the normative Figure 13.

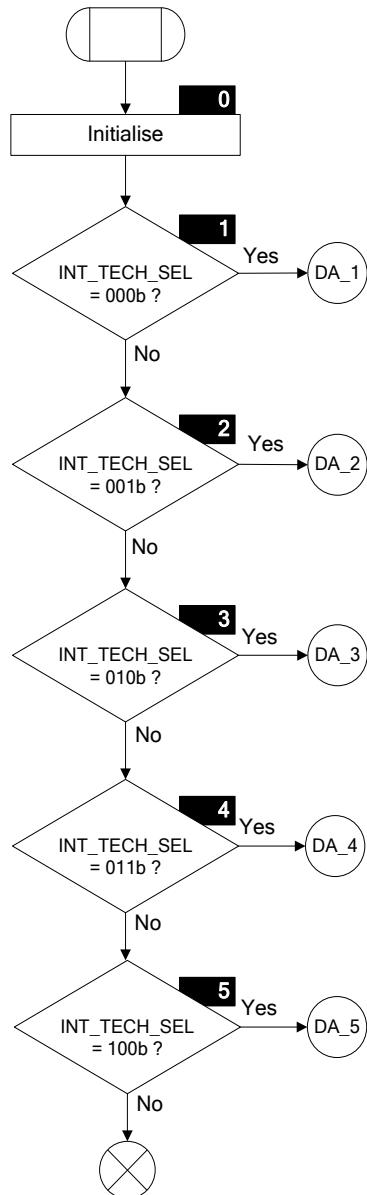


Figure 13: Device Activation Activity (Sheet 1, Entry) – Normative Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 13.

Requirements 34: Device Activation Activity

Poll Mode

- 9.4.3.1** Symbol 0:
The Poller SHALL set INT_DX_TECHNOLOGY to the value of INT_TECH_SEL and SHALL set INT_DX_BIT_RATE to 00b.
- 9.4.3.2** Symbol 1:
If INT_TECH_SEL is equal to 000b, then continue with DA_1 (Flow Chart and Requirements for NFC-A).
Otherwise the Poller SHALL proceed to Symbol 2.
- 9.4.3.3** Symbol 2:
If INT_TECH_SEL is equal to 001b, then continue with DA_2 (Flow Chart and Requirements for NFC-B).
Otherwise the Poller SHALL proceed to Symbol 3.
- 9.4.3.4** Symbol 3:
If INT_TECH_SEL is equal to 010b, then continue with DA_3 (Flow Chart and Requirements for NFC-F).
Otherwise the Poller SHALL proceed to Symbol 4.
- 9.4.3.5** Symbol 4:
If INT_TECH_SEL is equal to 011b, then continue with DA_4 (Flow Chart and Requirements for NFC-V).
Otherwise the Poller SHALL proceed to Symbol 5.
- 9.4.3.6** Symbol 5:
If INT_TECH_SEL is equal to 100b, then continue with DA_5 (Flow Chart and Requirements for NFC-ACM).
Otherwise the Poller SHALL conclude the Device Activation Activity.
-

9.4.4 Flow Chart and Requirements for NFC-A

The purpose of the NFC-A-related part of the Device Activation Activity is to activate a device within range that has activated support for NFC-A (subset). Depending on the outcome of the Resolution Process of the previous Activity, the device to be activated supports NFC-DEP Protocol, Type 4A Tag Platform or Type 2 Tag Platform.

Figure 14 illustrates the NFC-DEP Protocol (NFC-A), Type 4A Tag Platform and Type 2 Tag Platform related parts of the Device Activation Activity.

NOTE There is no specific action for the Type 2 Tag Platform because this platform is activated implicitly upon completion of the Collision Resolution Activity, unless it is in the **SLEEP_A** State.

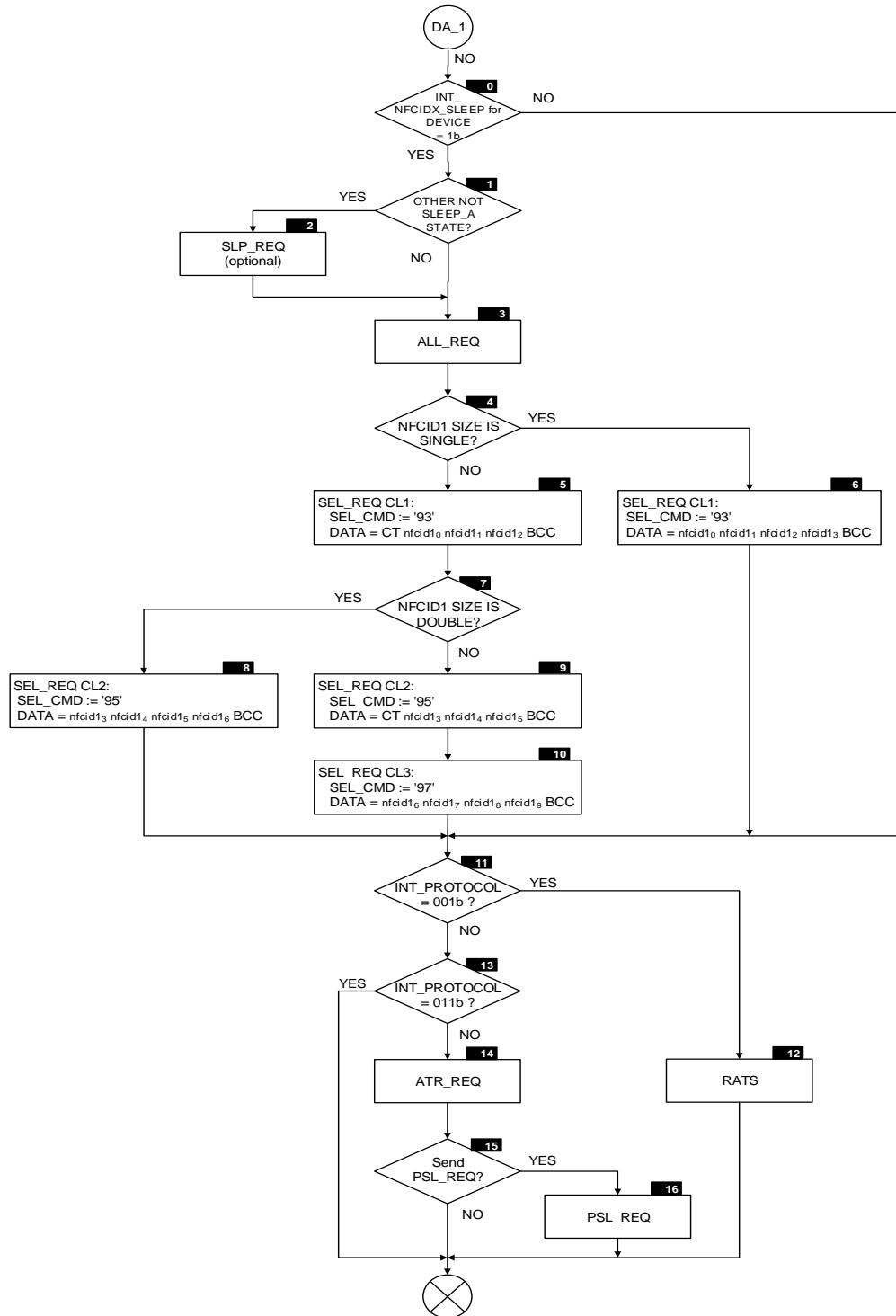


Figure 14: Device Activation Activity (Sheet 2, Connector DA_1, NFC-DEP (NFC-A), Type 2 & 4A Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 14.

Requirements 35: Device Activation Activity – NFC-DEP (NFC-A), Type 2, & 4A Tag Platform

Poll Mode

- 9.4.4.1** Symbol 0:
 If INT_NFCIDX_SLEEP[INT_INDEX] is equal to 1b (i.e., the device is in **SLEEP_A** State or a T2T may be in IDLE State after an error as defined in [DIGITAL]), then the Poller SHALL proceed to Symbol 1.
 Otherwise the Poller SHALL proceed to Symbol 11.
- 9.4.4.2** Symbol 1:
 If, for any value of x ($x \geq 1, x \leq M$) other than INT_INDEX, INT_NFCIDX_SLEEP[x] is equal to 0b (i.e. at least one other device is not in **SLEEP_A** State), then the Poller SHALL proceed to Symbol 2.
 Otherwise the Poller SHALL proceed to Symbol 3.
- 9.4.4.3** Symbol 2:
 The Poller MAY send a SLP_REQ Command to ensure that all devices are in the **SLEEP_A** State. If the SLP_REQ Command is sent, then INT_NFCIDX_SLEEP[M] SHALL be set to 1b.
- 9.4.4.4** Symbol 3:
 The Poller SHALL send an ALL_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]).
- 9.4.4.5** Symbol 4:
 If INT_NFCIDX [INT_INDEX] indicates a single size NFCID1, then the Poller SHALL proceed to Symbol 6.
 Otherwise the Poller SHALL proceed to Symbol 5.
- 9.4.4.6** Symbol 5:
 If INT_NFCIDX [INT_INDEX] indicates a double- or triple-size NFCID1, then the Poller SHALL first select cascade level 1 by sending a SEL_REQ Command with SEL_CMD = 93h and NFCID1 CL1, before continuing with cascade level 2.
 The Poller SHALL proceed to Symbol 7.
- 9.4.4.7** Symbol 6:
 The Poller SHALL send a SEL_REQ Command with SEL_CMD = 93h and NFCID1 CL1 (of INT_NFCIDX [INT_INDEX]) and SHALL wait for a Response afterward (as defined in [DIGITAL]).
 The Poller SHALL proceed to Symbol 11.
- 9.4.4.8** Symbol 7:
 If INT_NFCIDX[INT_INDEX] indicates a double-size NFCID1, then the Poller SHALL proceed to Symbol 8.
 Otherwise the Poller SHALL proceed to Symbol 9.
-

Poll Mode

- 9.4.4.9** Symbol 8:
The Poller SHALL send a SEL_REQ Command with SEL_CMD = 95h and NFCID1 CL2 (of INT_NFCIDX [INT_INDEX]) and SHALL wait for a Response afterward (as defined in [DIGITAL]).
The Poller SHALL proceed to Symbol 11.
- 9.4.4.10** Symbol 9:
If INT_NFCIDX[INT_INDEX] indicates a triple-size NFCID1, then the Poller SHALL first select cascade level 2 by sending a SEL_REQ Command with SEL_CMD = 95h and NFCID1 CL2, before continuing with cascade level 3.
- 9.4.4.11** Symbol 10:
The Poller SHALL send a SEL_REQ Command with SEL_CMD = 97h and NFCID1 CL3 (of INT_NFCIDX [INT_INDEX]) and SHALL wait for a Response afterward (as defined in [DIGITAL]).
- 9.4.4.12** Symbol 11:
If INT_PROTOCOL is equal to 001b, then the Poller SHALL proceed to Symbol 13.
Otherwise the Poller SHALL proceed to Symbol 13.
- 9.4.4.13** Symbol 12:
The Poller SHALL send a RATS Command (as specified in [DIGITAL]), containing the CON_RATS and SHALL wait for a Response afterward (as defined in [DIGITAL]).
The Poller SHALL handle the RATS Response (as specified in [DIGITAL]).
If a Valid RATS Response is received, the Poller SHALL:
 - Set INT_NFCIDX_SLEEP[INT_INDEX] to 0b
 - Store the RATS Response in GRE_RATS.
The Poller SHALL conclude the Device Activation Activity.
- 9.4.4.14** Symbol 13:
If INT_PROTOCOL is equal to 011b, then the Poller SHALL:
 - Set INT_NFCIDX_SLEEP[INT_INDEX] to 0b
 - Conclude the Device Activation Activity.
Otherwise the Poller SHALL proceed to Symbol 14.
- 9.4.4.15** Symbol 14:
The Poller SHALL send an ATR_REQ Command (as specified in [DIGITAL]), containing CON_ATR, CON_GB and the identifier INT_NFCIDX [INT_INDEX] and SHALL wait for a Response afterward (as defined in [DIGITAL]). The Poller SHALL handle the ATR_RES Response (as specified in [DIGITAL]). If a Valid ATR_RES Response is received, the Poller SHALL:
 - Set INT_NFCIDX_SLEEP[INT_INDEX] to 0b
 - Store the ATR_RES Response in GRE_ATR.
-

Poll Mode**9.4.4.16** Symbol 15:

The Poller SHALL proceed to Symbol 17 if all of the following conditions apply:

- PSL_REQ Command is supported.
- The Poller wants to change the Length Reduction Values by using the FSL parameter of the PSL_REQ Command (as defined in [DIGITAL]), and/or CON_BITR_NFC_DEP is equal to or greater than 2.
- The device identified by INT_INDEX is the only device that the Poller activates during execution of the active Poll Profile.

Otherwise the Poller SHALL conclude the Device Activation Activity.

9.4.4.17 Symbol 16:

The Poller SHALL send a PSL_REQ Command:

- If CON_BITR_NFC_DEP is equal to either 0 or 1, then the Poller SHALL set DS1 and DRI equal to 000b.
- If CON_BITR_NFC_DEP is equal to 2, then the Poller SHALL set DS1 and DRI equal to 001b.
- If CON_BITR_NFC_DEP is equal to or greater than 3, then the Poller SHALL set DS1 and DRI equal to 010b.

In addition it SHALL wait for a Response afterward (as defined in [DIGITAL]).

The PSL_REQ Command SHALL be coded as specified in [DIGITAL].

The Poller SHALL handle the PSL_RES Response (as specified in [DIGITAL]).

If a Valid PSL_RES Response is received, the Poller SHALL:

- Set INT_RX_BIT_RATE according to the bit rate specified by the DS1 and DRI parameter of the PSL_REQ Command
- If CON_BITR_NFC_DEP is equal to 2 or 3, set INT_RX_TECHNOLOGY to 010b, indicating NFC-F.

The Poller SHALL conclude the Device Activation Activity.

NOTE For activation of a Type 2 Tag Platform the Poller sends a Valid Read or Write Command in compliance with the Type 2 Tag Platform (as specified in [DIGITAL]), handles the Response (also as specified in [DIGITAL]), and concludes the Device Activation Activity.

NOTE If DS1 has been set to 001 or 010b and a valid PSL_RES Response has been received, the further communication uses NFC-F.

9.4.5 Flow Chart and Requirements for NFC-B

The purpose of the NFC-B Device Activation Activity is to activate a device within range that has activated support for the Type 4B Tag Platform.

Figure 15 illustrates the Type 4B Tag Platform related part of the Device Activation Activity.

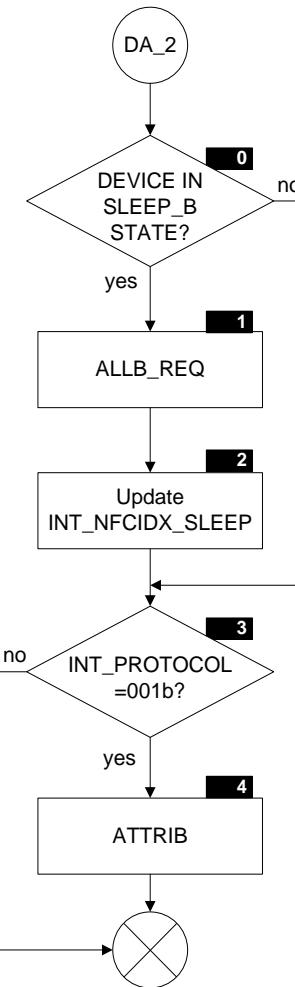


Figure 15: Device Activation Activity (Sheet 3, Connector DA_2, Type 4B Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 15.

Requirements 36: Device Activation Activity – Type 4B Tag Platform

Poll Mode

9.4.5.1 Symbol 0:

If INT_NFCIDX_SLEEP[INT_INDEX] is equal to 1b (i.e. the device is in **SLEEP_B** State), then the Poller SHALL proceed to Symbol 1.

Otherwise the Poller SHALL proceed to Symbol 3.

9.4.5.2 Symbol 1:

The Poller SHALL send an ALLB_REQ Command (as specified in [DIGITAL]) with the number of slots set equal to 1 (N=1) and indicating Extended SENSB_RES support as configured by CON_EXT_SENSB_RES and SHALL wait for a Response afterward (as defined in [DIGITAL]).

NOTE The Response, if any, is not used.

9.4.5.3 Symbol 2:

The Poller SHALL set INT_NFCIDX_SLEEP[0:m] to 0b

9.4.5.4 Symbol 3:

If INT_PROTOCOL is equal to 001b, the Poller SHALL proceed to Symbol 4.

Otherwise the Poller SHALL conclude the Device Activation Activity.

9.4.5.5 Symbol 4:

The Poller SHALL send an ATTRIB Command (as specified in [DIGITAL]), containing CON_ATTRIB and the NFCID0 included in INT_NFCIDX and SHALL wait for a Response afterward (as defined in [DIGITAL]).

If a Valid ATTRIB Response is received, the Poller SHALL store the ATTRIB Response in GRE_ATTRIB.

[DIGITAL] defines the ATTRIB Command and Response.

The Poller SHALL conclude the Device Activation Activity.

9.4.6 Flow Chart and Requirements for NFC-F

The purpose of the NFC-F Device Activation Activity is to activate a device within range that has activated support for NFC-F (subset). Such devices can support NFC-DEP Protocol or Type 3 Tag Platform.

Figure 16 illustrates the NFC-DEP Protocol (NFC-F) and Type 3 Tag Platform part of the Device Activation Activity.

NOTE There is no specific action for the Type 3 Tag Platform as this platform is activated implicitly upon completion of the NFC-F Collision Resolution Activity.

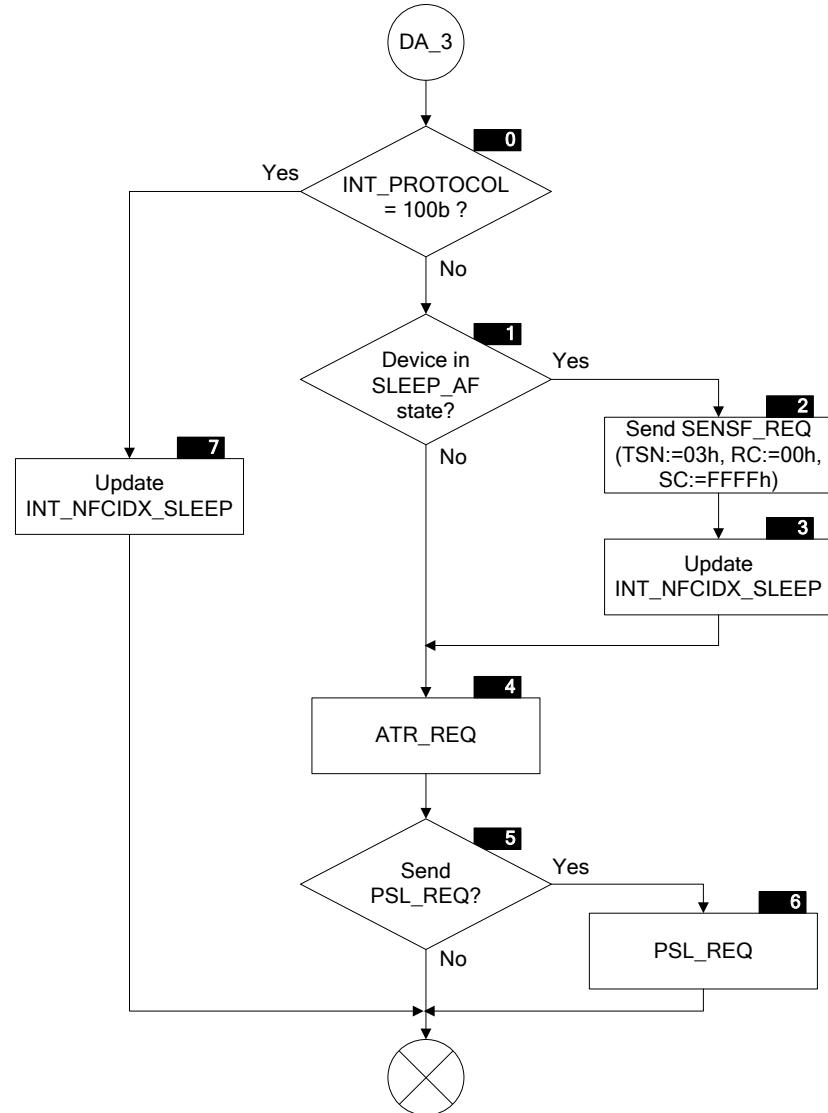


Figure 16: Device Activation Activity (Sheet 4, Connector DA_3, NFC-DEP (NFC-F), Type 3 Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 16.

Requirements 37: Device Activation Activity – NFC-DEP (NFC-F), Type 3 Tag Platform

Poll Mode

9.4.6.1 Symbol 0:

If INT_PROTOCOL is equal to 100b, then the Poller SHALL proceed to Symbol 7.

Otherwise the Poller SHALL proceed to Symbol 1.

9.4.6.2 Symbol 1:

If INT_NFCIDX_SLEEP[INT_INDEX] is equal to 1b (i.e. the device is in **SLEEP_AF** State), then the Poller SHALL proceed to Symbol 2.

Otherwise the Poller SHALL proceed to Symbol 4.

9.4.6.3 Symbol 2:

The Poller SHALL send a SENSF_REQ Command with TSN set to 03h, RC set to 00h and SC set to FFFFh and SHALL wait for a Response afterward (as defined in [DIGITAL]).

9.4.6.4 Symbol 3:

If a valid SENSF_RES Response is received indicating support for NFC-DEP protocol, the Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 0b, and SHALL proceed to Symbol 4.

Otherwise the Poller SHALL conclude the Device Activation Activity.

9.4.6.5 Symbol 4:

The Poller SHALL send an ATR_REQ Command (as specified in [DIGITAL]), containing CON_ATR, CON_GB and the identifier included in INT_NFCIDX and SHALL wait for a Response afterward (as defined in [DIGITAL]).

The Poller SHALL handle the ATR_RES Response (as specified in [DIGITAL]).

If a Valid ATR_RES Response is received, the Poller SHALL store the ATR_RES Response in GRE_ATR.

9.4.6.6 Symbol 5:

The Poller SHALL proceed to Symbol 6 if all of the following conditions apply:

- PSL_REQ Command is supported.
- The Poller wants to change the Length Reduction Values by using the FSL parameter of the PSL_REQ Command (as defined in [DIGITAL]) and/or the bit rate specified by CON_BITR_NFC_DEP is different than the current bit rate.
- The device identified by INT_INDEX is the only device that the Poller activates during execution of the active Poll Profile.

Otherwise the Poller SHALL conclude the Device Activation Activity.

Poll Mode**9.4.6.7** Symbol 6:

The Poller SHALL send a PSL_REQ Command:

- If CON_BITR_NFC_DEP is equal to 0, then the Poller SHALL set DSI and DRI equal to the current bit rate.
- If CON_BITR_NFC_DEP is equal to 1, then the Poller SHALL set DSI and DRI equal to 000b.
- If CON_BITR_NFC_DEP is equal to 2, then the Poller SHALL set DSI and DRI equal to 001b.
- If CON_BITR_NFC_DEP is equal to or greater than 3, then the Poller SHALL set DSI equal to 010b.

In addition it SHALL wait for a Response afterward (as defined in [DIGITAL]).

The PSL_REQ Command SHALL be coded as specified in [DIGITAL].

The Poller SHALL handle the PSL_RES Response as specified in [DIGITAL].

If a Valid PSL_RES Response is received, the Poller SHALL:

- Set INT_DX_BIT_RATE according to the bit rate specified by the DSI and DRI parameter in the PSL_REQ Command
- If CON_BITR_NFC_DEP is equal to 1, set INT_DX_TECHNOLOGY to 000b, indicating NFC-A.

The Poller SHALL conclude the Device Activation Activity.

9.4.6.8 Symbol 7:

The Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 0b, then conclude the Device Activation Activity.

NOTE If DSI has been set to 000 and a valid PSL_RES Response has been received, all further communication uses NFC-A.

9.4.7 Flow Chart and Requirements for NFC-V

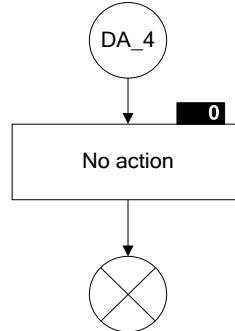


Figure 17: Device Activation Activity (Sheet 5, Connector DA_4, Type 5 Tag Platform – Flow Chart)

Symbols in this section refer to the corresponding symbols in Figure 17.

Requirements 38: Device Activation Activity – Type 5 Tag Platform

Poll Mode

9.4.7.1 Symbol 0:

There is no action for the Type 5 Tag, as this platform is activated after completion of NFC_V Collision Resolution Activity.

9.4.8 Flow Chart and Requirements for NFC-ACM

The purpose of the NFC-ACM Device Activation Activity is to activate a device within range that has activated support for the NFC-DEP protocol.

Figure 18 illustrates the NFC-DEP protocol related part of the Activation Activity.

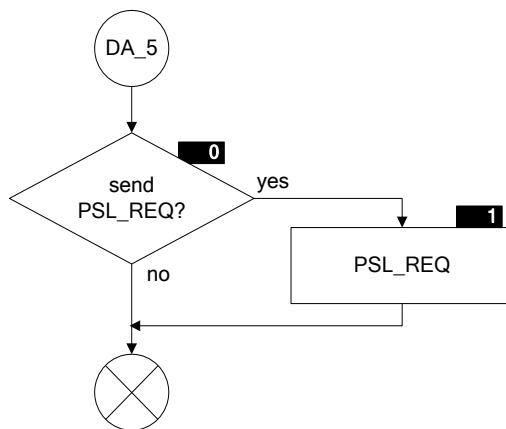


Figure 18: Device Activation Activity (Sheet 6, Connector DA_5, NFC-DEP (NFC-ACM)) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 18.

Requirements 39: Device Activation Activity – NFC-ACM

Poll Mode

9.4.8.1 Symbol 0:

The Poller SHALL proceed to Symbol 1 if all of the following conditions apply:

- PSL_REQ Command is supported.
- The Poller wants to change the Length Reduction Values by using the FSL parameter of PSL_REQ Command (as defined in [DIGITAL]), and/or the bit rate specified by CON_BITR_NFC_DEP is different than the current bit rate.

Otherwise the Poller SHALL conclude the Device Activation Activity.

9.4.8.2 Symbol 1:

The Poller SHALL send a PSL_REQ Command:

- If CON_BITR_NFC_DEP is equal to 0, then the Poller SHALL set DSI and DRI equal to the current bit rate.
- If CON_BITR_NFC_DEP is equal to 1, then the Poller SHALL set DSI and DRI equal to 000b.
- If CON_BITR_NFC_DEP is equal to 2, then the Poller SHALL set DSI and DRI equal to 001b.
- If CON_BITR_NFC_DEP is equal to or greater than 3, then the Poller SHALL set DSI equal to 010b.

In addition it SHALL wait for a Response afterward (as defined in [DIGITAL]).

The PSL_REQ Command SHALL be coded (as specified in [DIGITAL]).

The Poller SHALL handle the PSL_RES Response (as specified in [DIGITAL]).

If a Valid PSL_RES Response is received, the Poller SHALL set INT_RX_BIT_RATE according to the bit rate specified by the DSI and DRI parameter in the PSL_REQ Command.

The Poller SHALL conclude the Device Activation Activity.

9.5 Data Exchange Activity

This section describes the Data Exchange Activity.

9.5.1 Preconditions

There are no Configuration Parameters defined for this Activity.

The Input Parameters for the Data Exchange Activity are listed in Table 21.

Table 21: Data Exchange Activity – Input Parameters

Name	Format	Size	Description
INT_INDEX	integer	1 byte	Index to the identifier in INT_NFCIDX of the device to exchange data with.
INT_PROTOCOL	binary	3 bits	Protocol of device to exchange data with: <ul style="list-style-type: none"> • 000b: Use NFC-DEP • 001b: Use ISO-DEP • 011b: Use Type 2 Tag Platform • 100b: Use Type 3 Tag Platform • 101b: Use Type 5 Tag Platform

NOTE Use of Type 4 Tag Platform is covered by use of ISO-DEP.

The data requested from the Greedy Collection are listed in Table 22.

Table 22: Device Activation Activity – Input from Greedy Collection

Name	Format	Size	Description
GRE_ATR	Byte Sequence	≥ 17 bytes	ATR_RES Response of device activated
GRE_RATS	Byte Sequence	≥ 2 bytes	RATS Response of device activated
GRE_ATTRIB	Byte Sequence	≥ 1 byte	ATTRIB Response of device activated

9.5.2 Post-conditions

The Output Parameters returned to the Resolution Process are listed in Table 23.

Table 23: Data Exchange Activity – Output Parameters

Name	Format	Size	Description
INT_INDEX	integer	1 byte	Index to the identifier in INT_NFCIDX of the active device

No data are returned to the Greedy Collection by this Activity.

9.5.3 Flow Chart (Normative)

The Data Exchange Activity to be performed depends on the value of the INT_PROTOCOL parameter and is defined in the normative Figure 19.

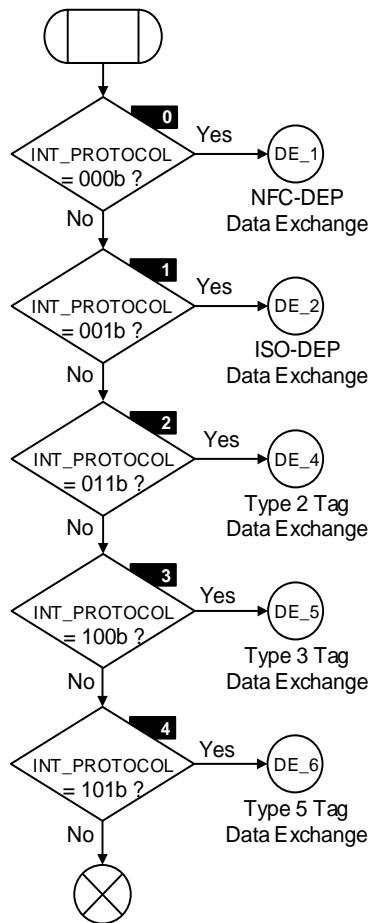


Figure 19: Data Exchange Activity (Sheet 1, entry) – Normative Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 19.

Requirements 40: Data Exchange Activity

Poll Mode

- 9.5.3.1** Symbol 0:
If INT_PROTOCOL is equal to 000b, then continue with DE_1 (Flow Chart and Requirements for NFC-DEP).
Otherwise the Poller SHALL proceed to Symbol 1.
- 9.5.3.2** Symbol 1:
If INT_PROTOCOL is equal to 001b, then continue with DE_2 (Flow Chart and Requirements for ISO-DEP).
Otherwise the Poller SHALL proceed to Symbol 2.
- 9.5.3.3** Symbol 2:
If INT_PROTOCOL is equal to 011b, then continue with DE_4 (Flow Chart and Requirements for Type 2 Tag Platform).
Otherwise the Poller SHALL proceed to Symbol 3.
- 9.5.3.4** Symbol 3:
If INT_PROTOCOL is equal to 100b, then continue with DE_5 (Flow Chart and Requirements for Type 3 Tag Platform).
Otherwise the Poller SHALL proceed to Symbol 4.
- 9.5.3.5** Symbol 4:
If INT_PROTOCOL is equal to 101b, then continue with DE_6 (Flow Chart and Requirements for Type 5 Tag Platform).
Otherwise the Poller SHALL conclude the Data Exchange Activity.
-

9.5.4 Flow Chart and Requirements for NFC-DEP

The purpose of the NFC-DEP Data Exchange Activity is to exchange data with a device within range, communicating over NFC-DEP.

Figure 20 illustrates the NFC-DEP-related part of the Data Exchange Activity.

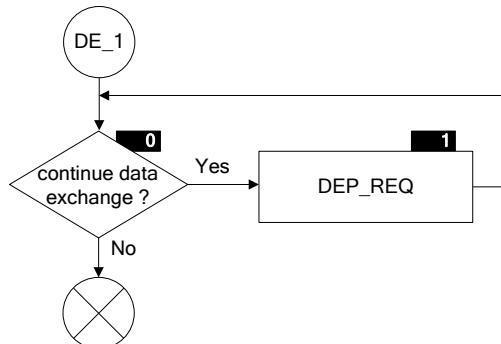


Figure 20: Data Exchange Activity (Sheet 2, connector DE_1, NFC-DEP using Passive or Active Communication Mode) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 20.

Requirements 41: Data Exchange Activity – NFC-DEP

Poll Mode

9.5.4.1 Symbol 0, Symbol 1:

As long as data exchange continues, as controlled by the adjacent upper layer, the Poller SHALL send a DEP_REQ Command and SHALL wait for the DEP_RES Response afterward (as specified in [DIGITAL]).

Otherwise the Poller SHALL conclude the Data Exchange Activity.

9.5.5 Flow Chart and Requirements for ISO-DEP

The purpose of the ISO-DEP Data Exchange Activity is to exchange data with a device within range, communicating over ISO-DEP.

Figure 21 illustrates the ISO-DEP-related part of the Data Exchange Activity.

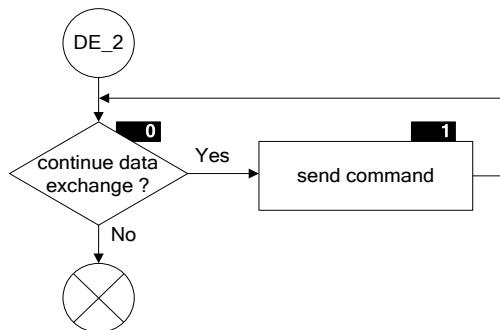


Figure 21: Data Exchange Activity (Sheet 3, Connector DE_2, ISO-DEP) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 21.

Requirements 42: Data Exchange Activity – ISO-DEP

Poll Mode

9.5.5.1 Symbol 0, Symbol 1:

As long as data exchange continues, as controlled by the adjacent upper layer, the Poller SHALL send Commands and receive Responses (as specified in [T4T]) using ISO-DEP (as specified in [DIGITAL]).

Otherwise the Poller SHALL conclude the Data Exchange Activity.

The Poller MAY send Commands and receive Responses that are proprietary for the Type 4 Tag.

9.5.6 Flow Chart and Requirements for Type 2 Tag Platform

The purpose of the Type 2 Tag Platform Data Exchange Activity is to exchange data with a device within range that has activated support for the Type 2 Tag Platform.

Figure 22 illustrates the Type 2 Tag Platform-related part of the Data Exchange Activity.

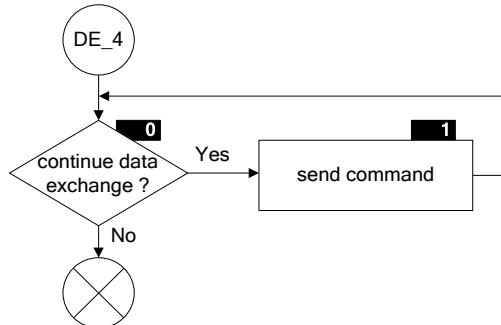


Figure 22: Data Exchange Activity (Sheet 5, connector DE_4, Type 2 Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 22.

Requirements 43: Data Exchange Activity – Type 2 Tag Platform

Poll Mode

9.5.6.1

Symbol 0, Symbol 1:

As long as data exchange continues, as controlled by the adjacent upper layer, the Poller SHALL send Commands and receive Responses (as specified in [T2T]).

Otherwise the Poller SHALL conclude the Data Exchange Activity.

The Poller MAY send Commands and receive Responses that are proprietary for the Type 2 Tag.

9.5.7 Flow Chart and Requirements for Type 3 Tag Platform

The purpose of the Type 3 Tag Platform Data Exchange Activity is to exchange data with a device within range that has activated support for the Type 3 Tag Platform.

Figure 23 illustrates the Type 3 Tag Platform related part of the Data Exchange Activity, including the selection of a Type 3 Tag Platform with a specific system code, e.g. 12FCh for NDEF-enabled Type 3 Tag Platform.

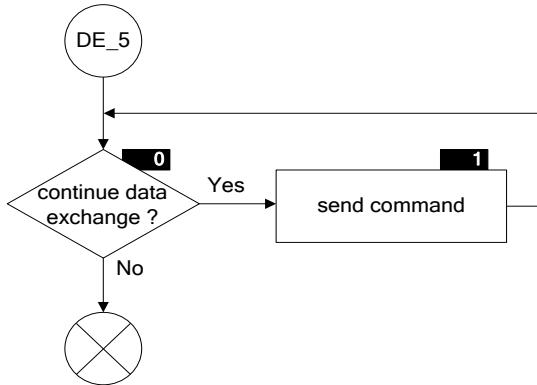


Figure 23: Data Exchange Activity (Sheet 6, connector DE_5, Type 3 Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 23.

Requirements 44: Data Exchange Activity – Type 3 Tag Platform

Poll Mode

9.5.7.1

Symbol 0, Symbol 1:

As long as data exchange continues, as controlled by the adjacent upper layer, the Poller SHALL send Commands and receive Responses (as specified in [T3T]).

Otherwise the Poller SHALL conclude the Data Exchange Activity.

If the Poller intends to select a Type 3 Tag Platform for NDEF Message Data Exchange (as defined in [T3T]), it SHALL send a SENSF_REQ Command with SC set to 12FCh and RC set to 00h.

When sending a SENSF_REQ Command using a System Code of FFFFh to select a Type 3 Tag Platform for any other Data Exchange, the Poller SHALL set RC to 01h, which excludes responses indicating NFC-DEP support.

If Valid SENSF_RES Responses are received, the Poller MAY use the data contained in a SENSF_RES Response (e.g. NFCID2) during the further communication in the Data Exchange Activity instead of the data associated to INT_INDEX.

NOTE The above requirements do not forbid a Poller to send SENSF_REQ Commands with other System Code values during the Data Exchange Activity. For such SENSF_REQ Commands RC can be set to any allowed value.

The Poller MAY send Commands and receive Responses that are proprietary for the Type 3 Tag.

9.5.8 Flow Chart and Requirements for Type 5 Tag Platform

The purpose of the Type 5 Tag Platform Data Exchange Activity is to exchange data with a device within range that has activated support for the Type 5 Tag Platform.

Figure 24 illustrates the Type 5 Tag Platform-related part of the Data Exchange Activity.

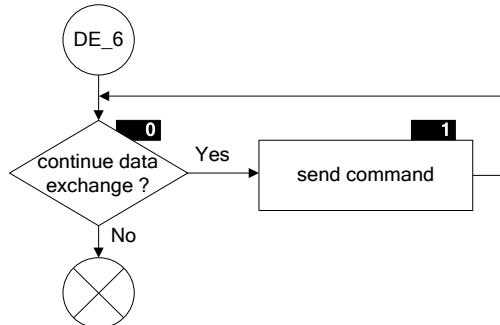


Figure 24: Data Exchange Activity (Sheet 5, connector DE_6, Type 5 Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 24.

Requirements 45: Data Exchange Activity – Type 5 Tag Platform

Poll Mode

9.5.8.1

Symbol 0, Symbol 1:

As long as data exchange continues, as controlled by the adjacent upper layer, the Poller SHALL send Commands and receive Responses (as specified in [TST]).

Otherwise the Poller SHALL conclude the Data Exchange Activity.

The Poller MAY send Commands and receive Responses that are proprietary for the Type 5 Tag.

9.6 Device Deactivation Activity

This section describes the Device Deactivation Activity.

9.6.1 Preconditions

There are no Configuration Parameters defined for this Activity.

The Parameters requested from Resolution for the Device Deactivation Activity are listed in Table 24.

Table 24: Device Deactivation Activity – Input Parameters

Name	Format	Size	Description
INT_INDEX	integer	1 byte	Index to the identifier in INT_NFCIDX of the device to be deactivated.
INT_NFCIDX_SLEEP[m], m = 1 to M	array of binary values	1 bit	Each element contains information about the sleep state of the device with the corresponding index in INT_NFCIDX: 0b: Device is not in sleep state. 1b: Device is in sleep state. M denotes the number of devices resolved
INT_PROTOCOL	binary	3 bits	Protocol to be deactivated: <ul style="list-style-type: none"> • 000b: Using NFC-DEP • 001b: Using ISO-DEP • 011b: Using Type 2 Tag Platform protocol • 100b: Using Type 3 Tag Platform protocol • 101b: Using Type 5 Tag Platform protocol.

No data are needed from the Greedy Collection for this Activity.

9.6.2 Post-conditions

The Output Parameters returned to the Resolution Process are listed in Table 25.

Table 25: Device Deactivation Activity – Output Parameters

Name	Format	Size	Description
INT_INDEX	integer	1 byte	Index to the identifier in INT_NFCIDX of the device deactivated
INT_NFCIDX_SLEEP[m], m = 1 to M	array of binary values	1 bit	Each element contains information about the sleep state of the device with the corresponding index in INT_NFCIDX: 0b: Device is not in sleep state. 1b: Device is in sleep state. M denotes the number of devices resolved

No data are returned to the Greedy Collection by this Activity.

9.6.3 Flow Chart (Normative)

The Device Deactivation Activity to be performed depends on the value of the INT_PROTOCOL parameter and is defined in the normative Figure 25.

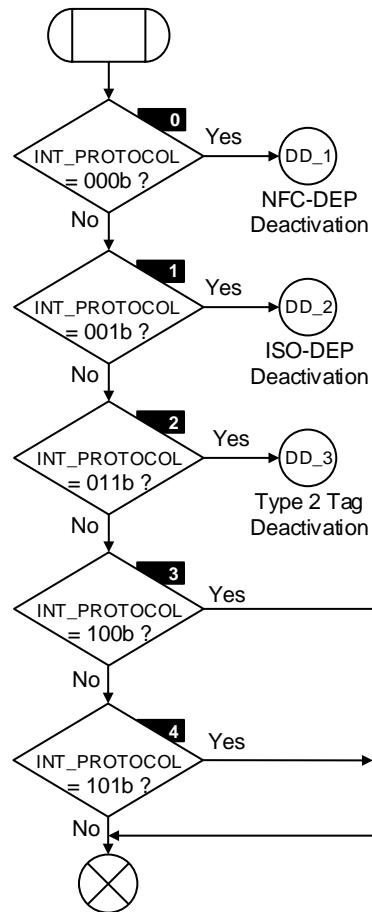


Figure 25: Device Deactivation Activity (Sheet 1, Entry) – Normative Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 25.

Requirements 46: Device Deactivation Activity

Poll Mode

- 9.6.3.1** Symbol 0:
If INT_PROTOCOL is equal to 000b, then continue with DD_1 (Flow Chart and Requirements for NFC-DEP).
Otherwise the Poller SHALL proceed to Symbol 1.
- 9.6.3.2** Symbol 1:
If INT_PROTOCOL is equal to 001b, then continue with DD_2 (Flow Chart and Requirements for ISO-DEP).
Otherwise the Poller SHALL proceed to Symbol 2.
- 9.6.3.3** Symbol 2:
If INT_PROTOCOL is equal to 011b, then continue with DD_3 (Flow Chart and Requirements for Type 2 Tag Platform).
Otherwise the Poller SHALL proceed to Symbol 3.
- 9.6.3.4** Symbol 3:
If INT_PROTOCOL is equal to 100b, then Poller SHALL conclude the Device Deactivation Activity (Flow Chart and Requirements for Type 3 Tag Platform).
Otherwise the Poller SHALL proceed to Symbol 4.
- 9.6.3.5** Symbol 4:
If INT_PROTOCOL is equal to 101b, then Poller SHALL conclude the Device Deactivation Activity (Flow Chart and Requirements for Type 5 Tag Platform).
Otherwise the Poller SHALL conclude the Device Deactivation Activity.
-

9.6.4 Flow Chart and Requirements for NFC-DEP

The purpose of the NFC-DEP Device Deactivation Activity is to deactivate a device within range, communicating over NFC-DEP.

Figure 26 illustrates the NFC-DEP-related part of the Device Deactivation Activity.

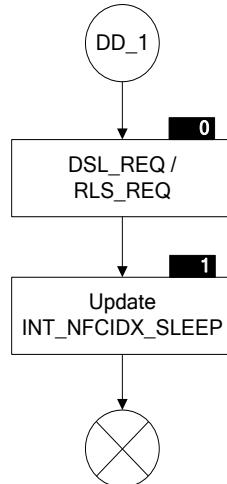


Figure 26: Device Deactivation Activity (Sheet 2, Connector DD_1, NFC-DEP) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 26.

Requirements 47: Device Deactivation Activity – NFC-DEP

Poll Mode

9.6.4.1 Symbol 0:

If in Passive Communication Mode, the Poller SHALL send an RLS_REQ Command or a DSL_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]).

If in Active Communication Mode, the Poller SHALL send an RLS_REQ Command and SHALL wait for a Response afterward (as defined in [DIGITAL]).

NOTE The definition for Active Communication Mode deviates from ISO/IEC 18092.

9.6.4.2 Symbol 1:

If in Passive Communication Mode and upon receipt of a valid RLS_RES Response, the Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 0b.

Otherwise, upon receipt of a valid DSL_RES Response, the Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 1b.

The Poller SHALL conclude the Device Deactivation Activity.

9.6.5 Flow Chart and Requirements for ISO-DEP

The purpose of the ISO-DEP Device Deactivation Activity is to deactivate a device within range, communicating over ISO-DEP.

Figure 27 illustrates the ISO-DEP-related part of the Device Deactivation Activity.

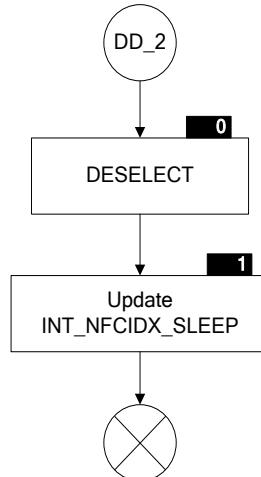


Figure 27: Device Deactivation Activity (Sheet 3, connector DD_2, ISO-DEP) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 27.

Requirements 48: Device Deactivation Activity – ISO-DEP

Poll Mode

9.6.5.1 Symbol 0:

The Poller SHALL send an S(DESELECT) Request (as specified in [DIGITAL]) and SHALL wait for a Response afterward (as specified in [DIGITAL]).

9.6.5.2 Symbol 1:

Upon receipt of the S(DESELECT) Response, the Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 1b.

The Poller SHALL conclude the Device Deactivation Activity.

9.6.6 Flow Chart and Requirements for Type 2 Tag Platform

The purpose of the Type 2 Tag Platform Device Deactivation Activity is to deactivate a Type 2 Tag Platform within range.

Figure 28 illustrates the Type 2 Tag Platform-related part of the Device Deactivation Activity.

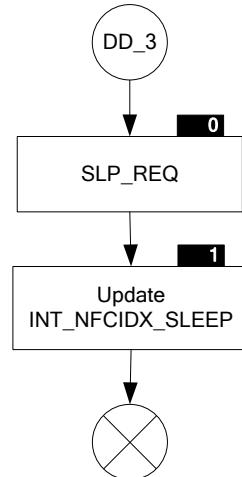


Figure 28: Device Deactivation Activity (Sheet 4, connector DD_3, Type 2 Tag Platform) – Flow Chart

Symbols in this section refer to the corresponding symbols in Figure 28.

Requirements 49: Device Deactivation Activity – Type 2 Tag Platform

Poll Mode

9.6.6.1 Symbol 0:

The Poller SHALL send a SLP_REQ Command (as specified in [DIGITAL]).

9.6.6.2 Symbol 1:

The Poller SHALL set INT_NFCIDX_SLEEP[INT_INDEX] to 1b.

The Poller SHALL conclude the Device Deactivation Activity.

9.6.7 Flow Chart and Requirements for Type 3 Tag Platform

For a Type 3 Tag Platform there is no particular Device Deactivation Activity.

9.6.8 Flow Chart and Requirements for Type 5 Tag Platform

For a Type 5 Tag Platform there is no particular Device Deactivation Activity.

A. Exhibit A

No items have been included in Exhibit A.

B. Listen Mode – State Diagram (Informative)

Figure 29 shows a graphical representation of a Listener.

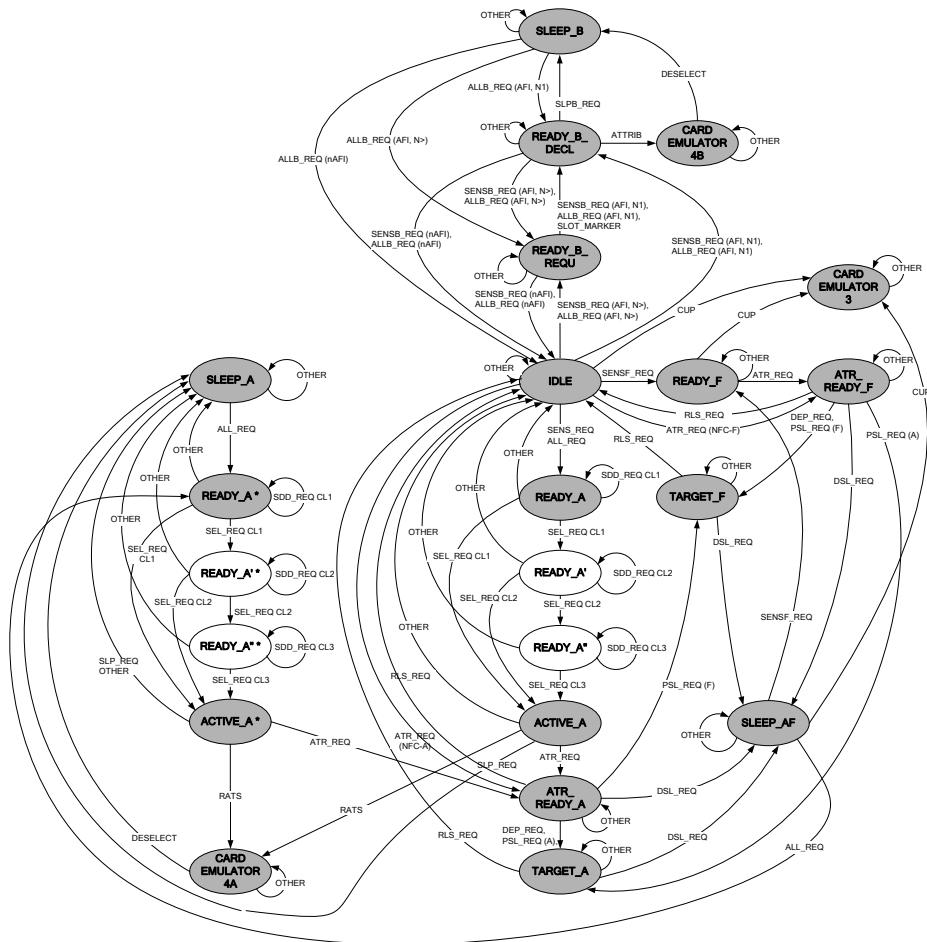


Figure 29: Listen Mode – State Diagram (Informative)

C. Values

Throughout this document, symbols are used to identify the values of Parameters. The actual values of the Parameters are listed in Table 26. A minimum value and a maximum value are defined for some of the Parameters. A single value is defined for other Parameters.

Parameters have one value for the Poller and another for the Listener. Unless otherwise specified, the value for Poll Mode has to be used when the Parameter is referenced in a Poll Mode requirement. The value for Listen Mode has to be used when the Parameter referenced in a Listen Mode requirement.

Table 26: Poll Mode and Listen Mode Parameter Values

Parameter	Poll Mode Value			Listen Mode Value			Units
	Min	Nominal	Max	Min	Nominal	Max	
t_{FIELD_OFF}	5.1					5.0	ms
$t_{TECHSWITCH}$	5.1						ms
$PTGT_A$	0.5						ms
$PTGT_B$	3.8						ms
$PTGT_F$	0.5						ms
$PTGT_V$	0.5						ms
$PTGT_{ACM}$	0.5						ms
GT_A	5.1			See [DIGITAL]			ms
GT_B	5.1			See [DIGITAL]			ms
GT_{BF}	15.3			See GT_F in [DIGITAL]			ms
GT_{FB}	20.4			See GT_F in [DIGITAL]			ms
GT_v	5.1			See [DIGITAL]			ms
GT_{ACM}	5.1					5	ms
T_{AD}	768		2559	768		2559	$1/f_c$
$T_{NRF,MAX}$		4607			4607		$1/f_c$
$t_{A,CMD,OFF}$	350		2559	350		2559	$1/f_c$
$t_{F,CMD,OFF}$	215		2559	215		2559	$1/f_c$
$t_{CMD,OFF,MAX}$		2559			2559		$1/f_c$
$T_{ARFG,MIN}$		1025			1025		$1/f_c$
$T_{ID,MIN}$		4097					$1/f_c$
T_{RFW}		512					$1/f_c$
n_{MIN}		0					
n_{MAX}		3					

NOTE $T_{NRF,MAX}$ is calculated using the formula $T_{AD} + (n_{MAX} + 1) \times T_{RFW}$ (using the Max value for T_{AD}).

D. Revision History

Table 27 outlines the revision history of the Activity Technical Specification.

Table 27: Revision History

Document Name	Revision and Release Date	Status	Change Notice	Supersedes
Activity	Version 1.0, November 2010	Final		
Activity	Version 1.1, March 2014	Final	Multiple fixes and clarifications	Version 1.0, April 2011
Activity	Version 2.0 Candidate, July 2015	Candidate Specification	Addition of NFC-V and NFC-ACM. Multiple fixes and clarifications	
Activity	Version 2.0 Nov. 2016	Final Technical Specification	Addition of Listen Mode and inclusion of many fixes, followed by edit for the final draft.	Version 2.0 Candidate, July 2015
Activity	Version 2.0 April 2017	Final Technical Specification	Editorial updates, including some reference changes and renaming.	Version 2.0 Final, Nov. 2016
Activity	Version 2.1 December 2019	Final	Profile section moved to the new Profile specification. Minor editorial changes and clarifications	Version 2.0 April 2017
Activity Technical Specification	Version 2.1 January 2020	Final	Editorial change of copyright notice.	Version 2.0 December 2019
Activity Technical Specification	Version 2.2 July 2021	Final	T1T and T1T Platform removal; NFC-A Collision resolution extension; NFC-V Collision resolution correction; T2T Platform error handling clarification; Maintenance. Editorial update.	Version 2.1 January 2020
Activity Technical Specification	Version 2.3 February 2023	Final	NFC-A collision resolution update. The collision bit can be set to any value. Editorial update.	Version 2.2 July 2021