UM10936

PN7150 User Manual

Rev. 2.0 — 6 November 2020
348120

User manual
COMPANY PUBLIC

Document information

Info	Content
Keywords	PN7150, NFC, NFCC, NCI 1.0
Abstract	This is a user manual for the PN7150 NFC Controller. The aim of this document is to describe the PN7150 interfaces, modes of operation and possible configurations.

Revision history

Rev	Date	Description
2.0	20201106	• Removed obsolete chapter about EMVCo PCD deactivation procedure
1.9	20200626	• Added definition of PN7150B0HN/C11006 • Removed definition of PN7150B0HN/C11004 because of discontinuity of this reference
1.8	20190902	• Added new Anti-tearing mechanism description (PN7150B0HN/C11004)
1.7	20190710	• Fixed error in parameter default values • Removed XTAL_SETTINGS_CFG and TXLDO_CFG registers description because obsolete
1.6	20181213	• Fixed error in parameter default values
1.5	20181113	• Fixed erroneous TAG_DETECTOR_PERIOD_CFG default value • Added mention about EEPROM memory corruption risk
1.4	20180914	• Fix issue in some parameter description and default value • Editorial updates
1.3	20180115	• Fix syntax issue in configurations descriptions • Fix wrong description of TEST_GET_REGISTER_CMD
1.2	20170119	• Fix syntax issue about VBAT pins and typo error about TEST_GET_REGISTER_RSP • Correct confused sentence about PLL switch off for power consumption optimization
1.1	20160524	• Security status changed into COMPANY PUBLIC
1.0	20141124	First official release of the document

Contact information

For more information, please visit: http://www.nxp.com

1. Introduction

The PN7150 is a full features NFC controller for contactless communication at 13.56 MHz.

The User Manual describes the software interfaces (API), based on the NFC FORUM standard, NCI.

Note: this document includes cross-references, which can be used to directly access the section/chapter referenced in the text. These cross-references are indicated by the following sign: '→'. This sign is positioned right before the section/chapter reference. The way to jump to the referenced section/chapter depends on the file format:

- In the word format, you have to first press the key "Ctrl" on the key board and then to click on the section/chapter reference number pointed by the '→' sign. The mouse symbol changes to a small hand when it is positioned on the section/chapter reference number.
- In .pdf format, you only have to click on the section/chapter reference number pointed by the '→' sign: the mouse symbol automatically changes to a small hand when it is positioned on the section/chapter reference number

As this document assumes pre-knowledge on certain technologies please check section →15: References to find the appropriate documentation.

For further information please refer to the PN7150 data sheet [PN7150_DS].

In this document the term „MIFARE card“ refers to a contactless card using an IC out of the MIFARE Classic, MIFARE Plus, MIFARE Ultralight or MIFARE DESFire product family

The PN7150 architecture overview

The PN7150 is an NFC Controller, which is briefly described in Fig 1:

- The top part describes the Device Host (DH) architecture with Higher Layer Driver (e.g. Android stack) hosting the different kind of applications (Reader/Writer, Peer to Peer, Card Emulation in the DH-NFCEE), the NCI driver & the transport layer driver.
- The PN7150 is the NFCC in the Fig 1. It is connected to the DH through a physical interface which is an I²C. The PN7150 firmware supports the NCI specification but also provides support for additional extensions that are not contained in the NCI specification. These additional extensions are specific to the PN7150 chip and are proprietary to NXP.
- The bottom part of the figure contains the RF antenna connected to the PN7150, which can communicate over RF with a Tag (Card) and a Reader/Writer or a Peer device.

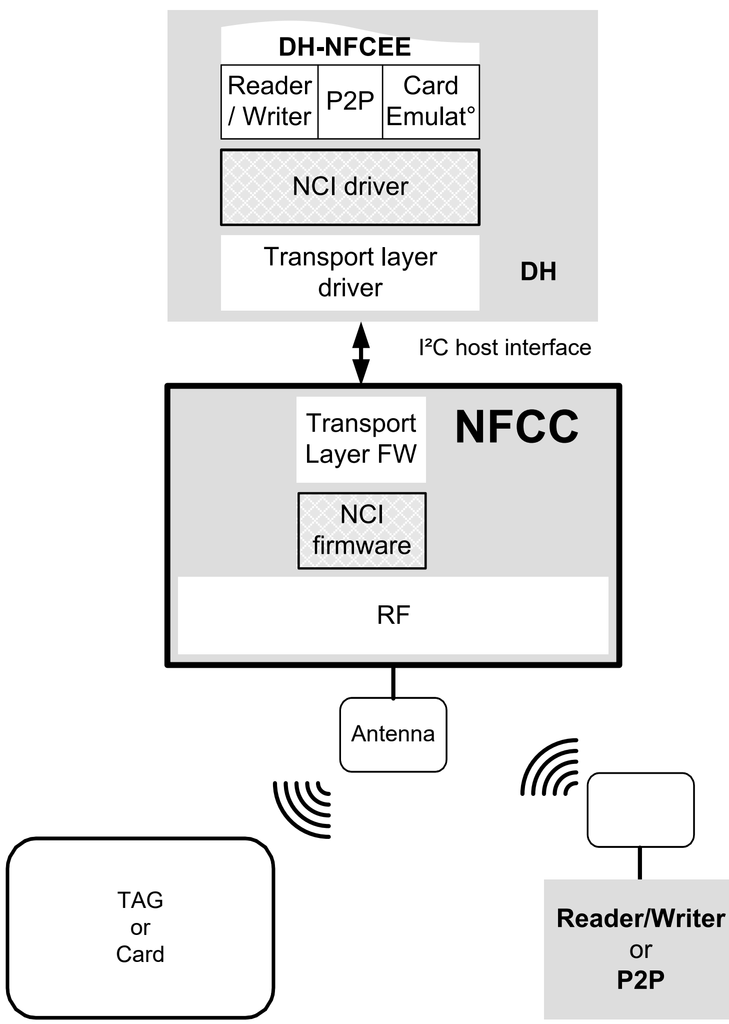

Fig 1. PN7150 system architecture

For contactless operation, several Modes of operation are possible, based on the overall system described above.

1.1 Reader/Writer Operation in Poll Mode

This mode of operation is further detailed in chapter →6.

The Reader/Writer application running on the DH is accessing a remote contactless Tag/Card, through the PN7150.

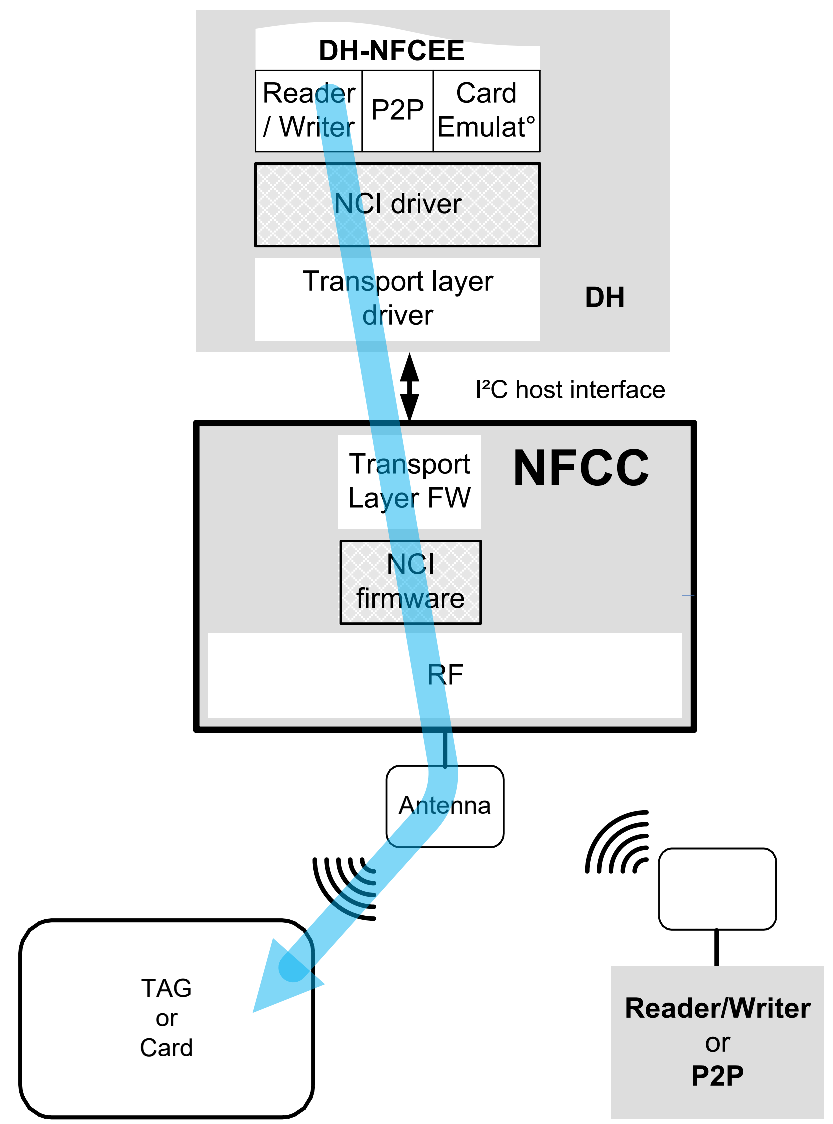

Fig 2. Reader/Writer

1.2 Card Emulation Operation in Listen Mode

This mode of operation is further detailed in chapter →7.

An external Reader/Writer accesses the DH-NFCEE emulating a contactless card, through the PN7150.

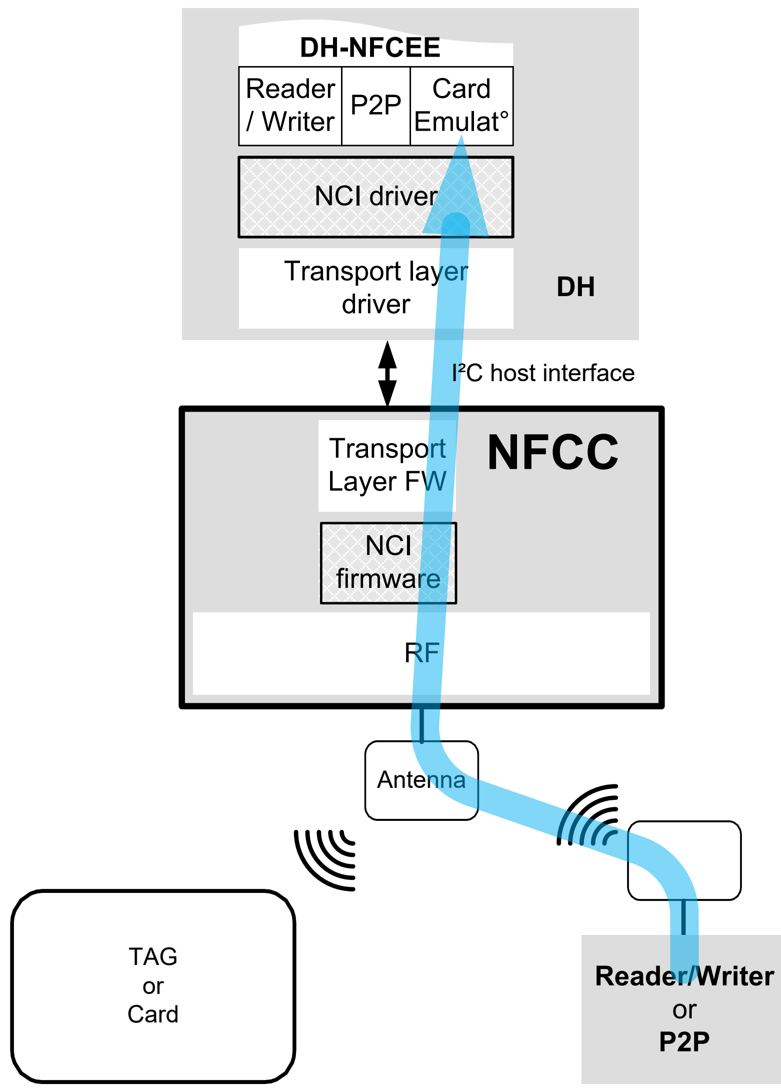

Fig 3. Card Emulation

1.3 Peer to Peer Operation in Listen & Poll Mode

This mode of operation is further detailed in chapter →8

The P2P application running on the DH is accessing a remote Peer device, through the PN7150.

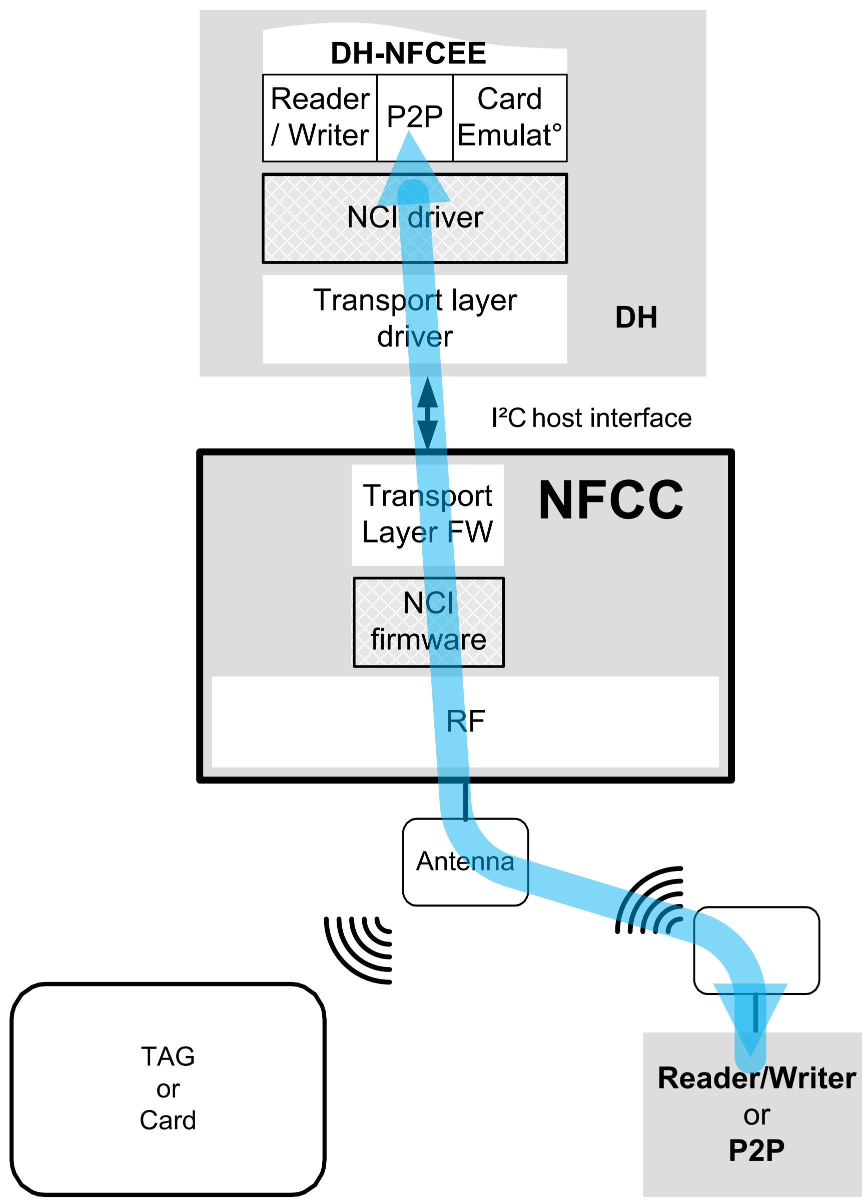

Fig 4. Peer to peer

1.4 Combined Modes of Operation

The PN7150 firmware is able to combine the basic modes of operation described above, using the RF Discovery as defined in [NCI]. As the PN7150 offers more features than what [NCI] addressed, NXP has defined some proprietary extensions.

The principle used to combine the various modes of operation is to build a cyclic activity which will sequentially activate various modes of operation. This cyclic activity is called the polling loop. This loop alternates listening phase (NFCC behaves as card or target) and polling phase (NFCC behaves as a reader/writer or an initiator). A cycle of the polling loop is called RF discovery sequence; it is made of 3 steps:

1. Start a Polling phase to look for a remote Tag/Card or a remote Target. If several technologies are enabled by the DH, PN7150 will poll sequentially for all the enabled technologies.
2. If no card or tag or target was detected, PN7150 enters a Listening phase, to potentially be activated as a Card / Tag emulator or a P2P target by an external Reader/Writer or external Initiator.
3. If no device to interact is detected during polling phase (step 1) or listening phase (step 2), then after a programmable timeout, PN7150 switches back to polling phase (step 1).

A combination of the 3 different steps defines a polling loop profile.

The RF discovery sequence is usually drawn as below (here applied for the NFC forum polling loop profile where technologies NFC-A, NFC-B & NFC-F are activated in Poll Mode):

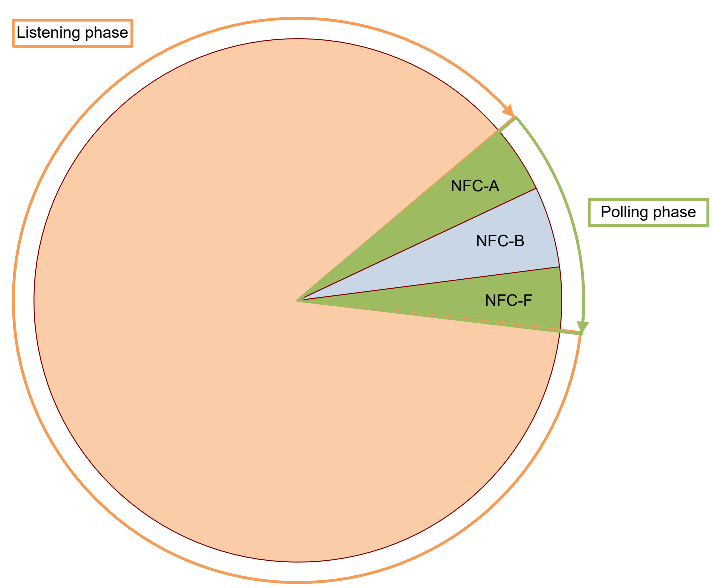

Fig 5. RF discovery sequence (NFC FORUM profile)

Please note that when the PN7150 is in Poll phase, it consumes a significant amount of current: in the range of 30mA (depending on the antenna characteristics). This applies at least for the 3 polled technologies drawn on the Fig 5, above (NFC-A, NFC-B and NFC_F) and it is due to the fact that the PN7150 has to generate the RF carrier (13.56MHz). However, during the Listen phase, the PN7150 current consumption is reduced to around 20µA when standby mode is enabled, due to the fact that it is waiting for the detection of an externally generated RF carrier.

Here is a figure illustrating a RF Discovery sequence, where polling is enabled only for NFC-A & NFC-B, for simplicity:

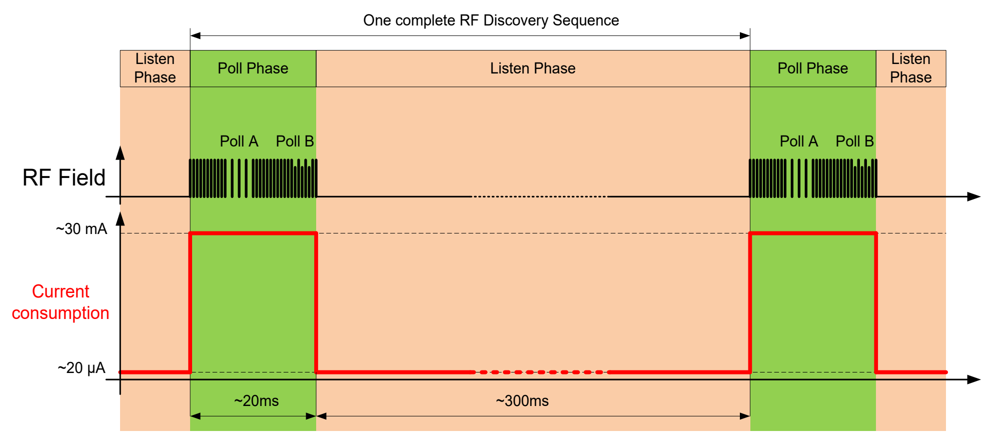

Fig 6. Power consumption during RF discovery sequence (NFC forum profile)

In a typical set-up, the polling phase is approximately 20ms long while the listening phase is usually in the range 300ms to 500ms long (this is configured thanks to the NCI parameter called TOTAL_DURATION).

For 500ms this gives an average power consumption of:

$$[30x20 + 0.02x500] / 520 = 1.17mA.$$

This average consumption can even be further optimized, using the PN7150 feature called “Tag Detector”. See chapter →9.4 for more details.

See chapter →9 for further details on the RF discovery activity.

2. NCI Overview

The aim of this section is to give an overview of the key points of the [NCI] specification.

2.1 NCI Components

Here below are described the NCI component as defined in [NCI] which are located in the NFCC embedded FW.

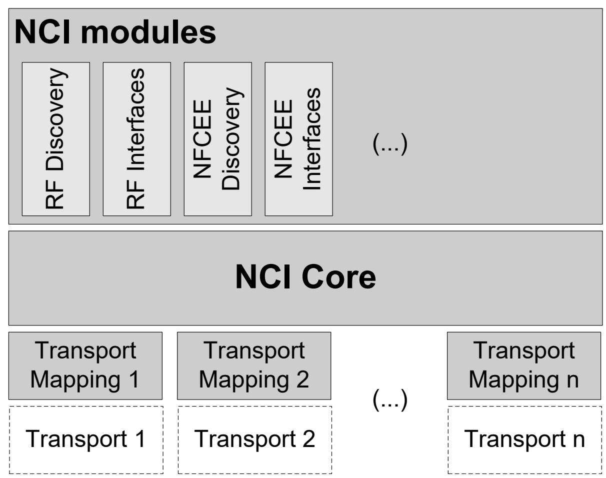

Fig 7. NCI components

2.1.1 NCI Modules

NCI modules are built on top of the functionality provided by the NCI Core. Each module provides a well-defined functionality to the DH. NCI modules provide the functionality to configure the NFCC and to discover and communicate with Remote NFC Endpoints (see [NCI] for definition) or with DH-NFCEEs.

Some NCI modules are mandatory parts of an NCI implementation, others are optional. There can also be dependencies between NCI modules in the sense that a module may only be useful if there are other modules implemented as well. For example, all modules that deal with communication with a Remote NFC Endpoint (the RF Interface modules) depend on the RF Discovery to be present.

2.1.2 NCI Core

The NCI Core defines the basic functionality of the communication between a Device Host (DH) and an NFC Controller (NFCC). This enables Control Message (Command, Response and Notification) and Data Message exchange between an NFCC and a DH.

2.1.3 Transport Mappings

Transport Mappings define how the NCI messaging is mapped to an underlying NCI Transport, which is a physical connection (and optional associated protocol) between the DH and the NFCC. Each Transport Mapping is associated with a specific NCI Transport (see [NCI] for definition).

2.2 NCI Concepts

This chapter outlines the basic concepts used in [NCI].

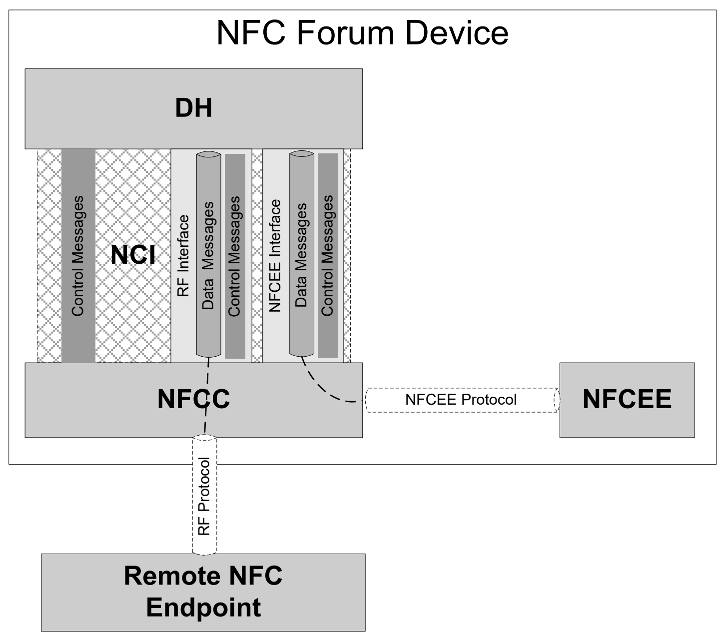

Fig 8. NCI concepts

2.2.1 Control Messages

A DH uses NCI Control Messages to control and configure an NFCC. Control Messages consist of Commands, Responses and Notifications. Commands are only allowed to be sent in the direction from DH to NFCC, Responses and Notifications are only allowed in the other direction. Control Messages are transmitted in NCI Control Packets, NCI supports segmentation of Control Messages into multiple Packets.

The NCI Core defines a basic set of Control Messages, e.g. for setting and retrieving of NFCC configuration parameters. NCI Modules can define additional Control Messages.

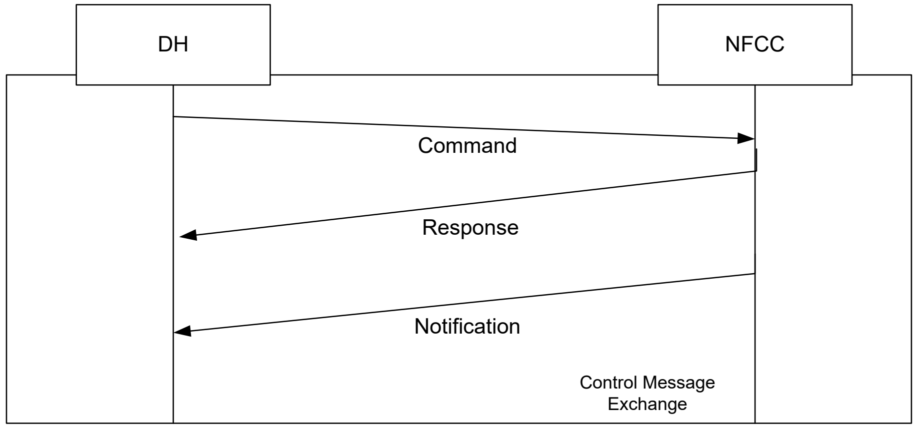

Fig 9. Control Message Exchange

2.2.2 Data Messages

Data Messages are used to transport data to either a Remote NFC Endpoint (named RF Communication in NCI) or to an NFCEE (named NFCEE Communication). NCI defines Data Packets enabling the segmentation of Data Messages into multiple Packets.

Data Messages can only be exchanged in the context of a Logical Connection. As a result, a Logical Connection must be established before any Data Messages can be sent. One Logical Connection, the Static RF Connection, is always established during initialization of NCI. The Static RF Connection is dedicated to be used for RF Communication. Additional Logical Connections can be created for RF and/or NFCEE Communication.

Logical Connections provide flow control for Data Messages in the direction from DH to NFCC.

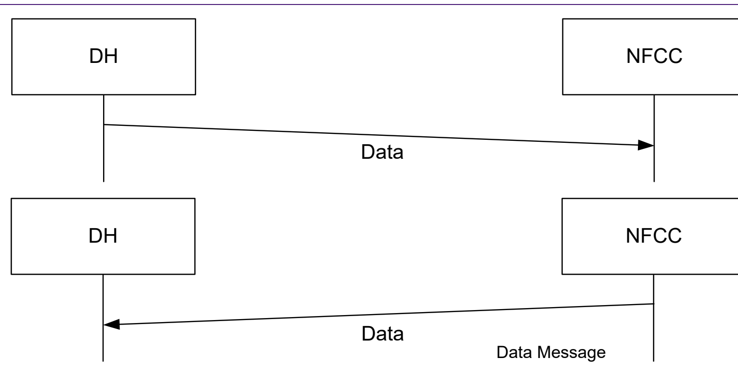

Fig 10. Data Message Exchange

2.2.3 Interfaces

An NCI Module may contain one Interface. An Interface defines how a DH can communicate via NCI with a Remote NFC Endpoint or NFCEE. Each Interface is defined to support specific protocols and can only be used for those protocols (the majority of Interfaces support exactly one protocol). NCI defines two types of Interfaces: RF Interfaces and NFCEE Interfaces.

Protocols used to communicate with a Remote NFC Endpoint are called RF Protocols. Protocols used to communicate with an NFCEE are called NFCEE Protocols.

An NFCEE Interface has a one-to-one relationship to an NFCEE Protocol, whereas there might be multiple RF Interfaces for one RF Protocol. The later allows NCI to support different splits of the protocol implementation between the NFCC and DH. An NCI implementation on an NFCC should include those RF Interfaces that match the functionality implemented on the NFCC.

Interfaces must be activated before they can be used and they must be deactivated when they are no longer used.

An Interface can define its own configuration parameters and Control Messages, but most importantly it must define how the payload of a Data Message maps to the payload of the respective RF or NFCEE Protocol and, in case of RF Communication, whether the Static RF Connection is used to exchange those Data Messages between the DH and the NFCC.

2.2.4 RF Communication

RF Communication is started by configuring and running the polling loop (RF discovery sequences in loops). The RF discovery sequence involved the NCI module called RF discovery. This module discovers and enumerates Remote NFC Endpoints.

For each Remote NFC Endpoint, the RF Discovery module provides the DH with the information about the Remote NFC Endpoint gathered during the RF Discovery sequence. One part of this information is the RF Protocol that is used to communicate with the Remote NFC Endpoint. During RF Discovery module configuration, the DH must configure a mapping that associates an RF Interface for each RF Protocol. If only a single Remote NFC Endpoint is detected during one discovery sequence, the RF Interface for this Endpoint is automatically activated. If there are multiple Remote NFC Endpoints detected during the Poll phase, the DH can select the Endpoint it wants to communicate with. This selection also triggers the activation of the mapped Interface.

After an RF Interface has been activated, the DH can communicate with the Remote NFC Endpoint using the activated RF Interface. An activated RF Interface can be deactivated by either the DH or the NFCC (e.g. on behalf of the Remote NFC Endpoint). However, each RF Interface can define which of those methods are allowed. Depending on which part of the protocol stack is executed on the DH there are different deactivation options. For example, if a protocol command to tear down the communication is handled on the DH, the DH will deactivate the RF Interface. If such a command is handled on the NFCC, the NFCC will deactivate the Interface.

This specification describes the possible Control Message sequences for RF Communication in the form of a state machine.

2.2.5 NFCEE Communication

The DH can learn about the NFCEEs connected to the NFCC by using the NFCEE Discovery module. During NFCEE Discovery the NFCC assigns an identifier for each NFCEE. When the DH wants to communicate with an NFCEE, it needs to open a Logical Connection to the NFCEE using the corresponding identifier and specifying the NFCEE Protocol to be used.

Opening a Logical Connection to an NFCEE automatically activates the NFCEE Interface associated to the protocol specified. As there is always a one-to-one relationship between an NFCEE Protocol and Interface, there is no mapping step required (different as for the RF Communication).

After the Interface has been activated, the DH can communicate with the NFCEE using the activated Interface.

Closing the connection to an NFCEE Interface deactivates that NFCEE Interface.

NCI also includes functionality to allow the DH to enable or disable the communication between an NFCEE and the NFCC.

2.2.6 Identifiers

The NFCC might only be used by the DH but also by the NFCEEs in the device (in such a case the NFCC is a shared resource). NFCEEs differ in the way they are connected to the NFCC and the protocol used on such a link determines how an NFCEE can use the NFCC. For example, some protocols allow the NFCEE to provide its own configuration for RF parameters to the NFCC (similar to the NCI Configuration Parameters for RF Discovery) in other cases the NFCEE might not provide such information.

NFCCs can have different implementation in how they deal with multiple configurations from DH and NFCEEs. They might for example switch between those configurations so that only one is active at a time or they might attempt to merge the different configurations. During initialization NCI provides information for the DH whether the configuration it provides is the only one or if the NFCC supports configuration by NFCEEs as well.

NCI includes a module, called Listen Mode Routing, with which the DH can define where to route received data when the device has been activated in Listen Mode. The Listen Mode Routing allows the DH to maintain a routing table on the NFCC. Routing can be done based on the technology or protocol of the incoming traffic or based on application identifiers in case [7816-4] APDU commands are used on top of ISO-DEP.

In case of PN7150 the only route is the DH-NFCEE, therefore no Listen Mode Routing programming supported.

In addition, NCI enables the DH to get informed if communication between an NFCEE and a Remote NFC Endpoint occurs.

2.3 NCI Packet Format

2.3.1 Common Packet Header

All Packets have a common header, consisting of an MT field and a PBF field:

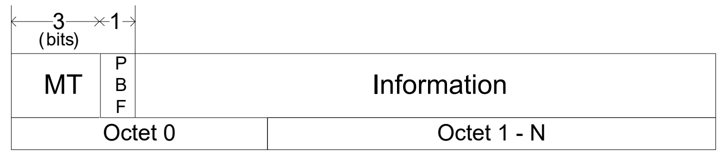

Fig 11. NCI Core Packet Format

- **Message Type (MT)**

The MT field indicates the contents of the Packet and SHALL be a 3 bit field containing one of the values listed in Table 1, below. The content of the Information field is dependent on the value of the MT field. The receiver of an MT designated as RFU SHALL silently discard the packet.

Table 1. MT values

MT	Description
000b	Data Packet
001b	Control Packet - Command Message as a payload
010b	Control Packet - Response Message as a payload
011b	Control Packet – Notification Message as a payload
100b-111b	RFU

- **Packet Boundary Flag (PBF)**

The Packet Boundary Flag (PBF) is used for Segmentation and Reassembly and SHALL be a 1 bit field containing one of the values listed in [NCI] specification.

Table 2. PBF Value

PBF	Description
0b	The Packet contains a complete Message, or the Packet contains the last segment of a segmented Message
1b	The Packet contains a segment of a Message which is not the last segment.

The following rules apply to the PBF flag in Packets:

- If the Packet contains a complete Message, the PBF SHALL be set to 0b.
- If the Packet contains the last segment of a segmented Message, the PBF SHALL be set to 0b.
- If the packet does not contain the last segment of a segmented Message, the PBF SHALL be set to 1b.

2.3.2 Control Packets

The Control Packet structure is detailed below.

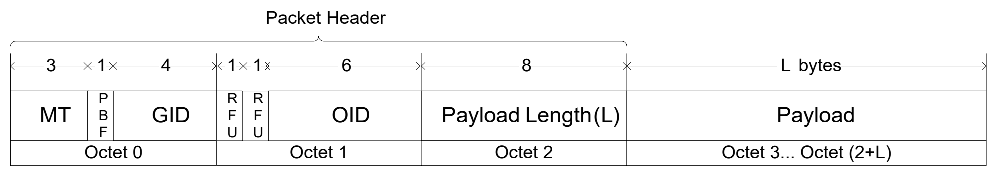

Fig 12. Control Packet Format

Each Control Packet SHALL have a 3 octet Packet Header and MAY have additional payload for carrying a Control Message or a segment of Control Message.

NOTE In the case of an 'empty' Control Message, only the Packet Header is sent.

- **Message Type (MT)**

Refer to section 2.3.1 for details of the MT field.

- **Packet Boundary Flag (PBF)**

Refer to section 2.3.1 for details of the PBF field.

- **Group Identifier (GID)**

The NCI supports Commands, Responses and Notifications which are categorized according their individual groups. The Group Identifier (GID) indicates the categorization of the message and SHALL be a 4 bit field containing one of the values listed in [NCI] specification.

All GID values not defined in [NCI] specification are RFU.

- **Opcode Identifier (OID)**

The Opcode Identifier (OID) indicates the identification of the Control Message and SHALL be a 6 bit field which is a unique identification of a set of Command, Response or Notification Messages within the group (GID). OID values are defined along with the definition of the respective Control Messages described in [NCI] specification.

- **Payload Length (L)**

The Payload Length SHALL indicate the number of octets present in the payload. The Payload Length field SHALL be an 8 bit field containing a value from 0 to 255.

2.3.3 Data Packets

The Data Packet structure is detailed below.

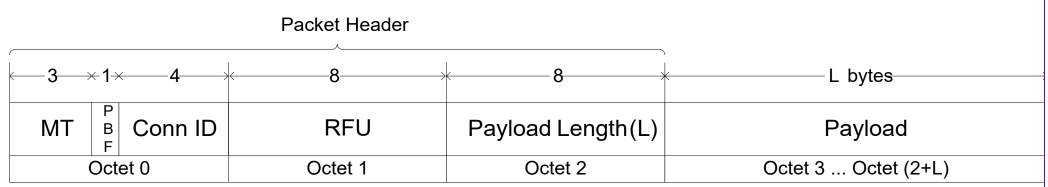

Fig 13. Data Packet Structure

Each Data Packet SHALL have a 3 octet Packet Header and MAY have additional Payload for carrying a Data Message or a segment of a Data Message.

NOTE: In the case of an ‘empty’ Data Message, only the Packet Header is sent.

- **Message Type (MT)**

Refer to section 2.3.1 for details of the MT field.

- **Packet Boundary Flag (PBF)**

Refer to section 2.3.1 for details of the PBF field.

- **Connection Identifier (Conn ID)**

The Connection Identifier (Conn ID) SHALL be used to indicate the previously setup Logical Connection to which this data belongs. The Conn ID is a 4 bit field containing a value from 0 to 15.

- **Payload Length (L)**

The Payload Length field indicates the number of Payload octets present. The Payload Length field is an 8 bit field containing a value from 0 to 255.

2.3.4 Segmentation and Reassembly

The Segmentation and Reassembly functionality SHALL be supported by both the DH and the NFCC.

Segmentation and Reassembly of Messages SHALL be performed independently for Control Packets and Data Packets of each Logical Connection.

Any NCI Transport Mapping is allowed to define a fixed Maximum Transmission Unit (MTU) size in octets. If such a Mapping is defined and used, then if either DH or NFCC needs to transmit a Message (either Control or Data Message) that would generate a Packet (including Packet Header) larger than the MTU, the Segmentation and Reassembly (SAR) feature SHALL be used on the Message.

The following rules apply to segmenting Control Messages:

- For each segment of a Control Message, the header of the Control Packet SHALL contain the same MT, GID and OID values.
- **From DH to NFCC:** the Segmentation and Reassembly feature SHALL be used when sending a Command Message from the DH to the NFCC that would generate a Control Packet with a payload larger than the “Max Control Packet Payload Size” reported by the NFCC at initialization. Each segment of a Command Message except for the last SHALL contain a payload with the length of “Max Control Packet Payload Size”.
- **From NFCC to DH:** when an NFCC sends a Control Message to the DH, regardless of the length, it MAY segment the Control Message into smaller Control Packets if needed for internal optimization purposes.

The following rules apply to segmenting Data Messages:

- For each segment of a Data Message, the header of the Data Packet SHALL contain the same MT and Conn ID.
- **From DH to NFCC:** if a Data Message payload size exceeds the Max Data Packet Payload Size, of the connection then the Segmentation and Reassembly feature SHALL be used on the Data Message.
- **From NFCC to DH:** when an NFCC sends a Data Message to the DH, regardless of the payload length it MAY segment the Data Message into smaller Data Packets for any internal reason, for example for transmission buffer optimization.

3. DH interface

3.1 Introduction

The I²C interface of the PN7150 is compliant with the I²C Bus Specification V3.0, including device ID and Soft Reset. It is slave-only, i.e. the SCL signal is an input driven by the host.

!	NCI packets can be as long as 258 Bytes. If the DH I²C peripheral has a buffer limitation which is below 258 Bytes, then a fragmentation mechanism SHALL be used at the I²C transport layer, as defined in →3.6.

The PN7150 I²C interface supports standard (up to 100kbps), fast-Speed mode (up to 400kbps) and High Speed mode (up to 3.4Mbit/s).

I²C defines two different modes of addressing (7-bit & 10-bit). The PN7150 only supports the 7-bit addressing mode.

The PN7150 I²C 7-bit address can be configured from 0x28 to 0x2B. The 2 least significant bits of the slave address are electrically forced by pins I2C_ADR0 and I2C_ADDR1 of the PN7150.

So, in binary format, the PN7150 slave 7-bit address is:

"0 1 0 1 0 I2C_ADDR1 I2C_ADDR0"

Table 3. PN7150 I²C slave address

Address Value	I2C_ADDR1 Pin	I2C_ADDR0 Pin
0x28	0	0
0x29	0	1
0x2A	1	0
0x2B	1	1

This can be easily configured through direct connection of pins I2C_ADDR0 and I2C_ADDR1 to either GND or PVDD at PCB level.

3.2 NCI Transport Mapping

In the PN7150, there is no additional framing added for I²C: an NCI packet (either data or control message, as defined in chapter →2.3) is transmitted over I²C "as is", i.e. without any additional Byte (no header, no CRC etc…).

3.3 Write Sequence from the DH

As the I²C clock is mastered by the DH, only the DH can initiate an I²C exchange.

A DH write sequence always starts with the sending of the PN7150 I²C Slave Address followed by the write bit (logical '0': 0b). Then the PN7150 I²C interface sends an I²C ACK back to the DH for each data byte written by the DH.

It may send an I²C NACK (negative acknowledge) when none of the buffers used by the NCI core in the PN7150 is free, which may happen in case PN7150 is in standby mode. If one single byte of a complete NCI frame is NACKed by the PN7150, the DH has to re-send the complete NCI frame and not only this single byte.

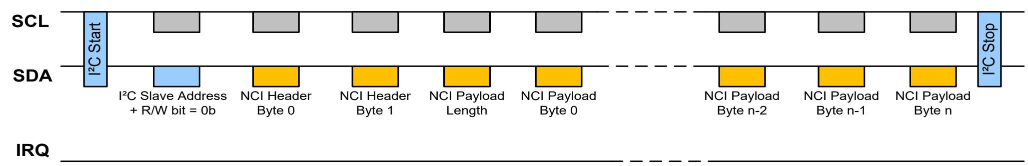

Fig 14. I²C Write sequence

> **!** It may happen that PN7150 has an NCI Message ready to be sent to the DH while it is receiving another NCI Message from the DH. In such a condition, the IRQ pin will be raised somewhere during the Write Sequence: this is not an error and has to be accepted by the DH: once the Write Sequence is completed, the DH has to start a Read Sequence (see →3.4).

3.4 Read Sequence from the DH

The DH shall never initiate a spontaneous I²C read request. The DH shall wait until it is triggered by the PN7150.

To trigger the DH, the PN7150 generates a logical transition from Low to High on its IRQ pin (if the IRQ pin is configured to be active High; see configuration chapter →10.1). So after writing any NCI command, the DH shall wait until the PN7150 raises its IRQ pin.

The DH can then transmit a Read request to fetch the NCI answer from the PN7150. When the PN7150 needs to send a spontaneous notification to the DH (for instance an RF Interface activation notification), the PN7150 raises the IRQ pin and the DH performs a normal read as described above.

A DH Read Sequence always starts by the sending of the PN7150 I²C Slave Address followed by the read bit (logical ‘1’). Then the DH I²C interface sends an ACK back to the PN7150 for each data Byte received.

Fig 15 is an example where the IRQ is raised so the DH can proceed a read.

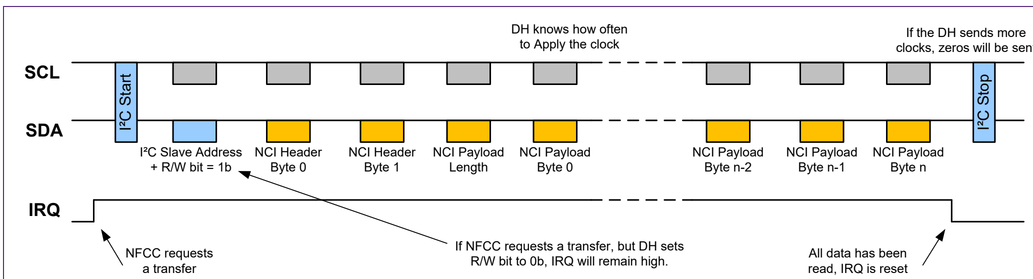

Fig 15. I²C Read sequence

As indicated on Fig 15, in case the PN7150 requests a data transfer by raising the IRQ pin and the DH tries to initiate a write sequence by positioning the write bit to 0b, the PN7150 keeps the IRQ active until the DH starts a read sequence.

The DH is not allowed to proceed with a write sequence once the PN7150 has set the IRQ pin to its active value (logical '1' in Fig 15).

If PN7150 has another message ready to be sent to the DH before the end of the on-going Read Sequence, the IRQ pin will be first deactivated at the end of the on-going Read Sequence and then re-activated to notify to the DH that a new message has to be read.

3.5 Split mode

The PN7150 supports the interruption of a frame transfer, as defined in [I²C]. This feature is only available in Read Mode; it is forbidden to use it in Write Mode.

This can be useful in a system where the I²C bus is shared between several peripherals: it allows the host to stop an on-going exchange, to switch to another peripheral (with a different slave address) and then to resume the communication with the PN7150.

Another typical use-case for the split mode is to have the DH reading first the NCI packet header, to know what the Payload length is. The DH can then allocate a buffer with an appropriate size and read the payload data to fill this buffer. This use-case is represented on Fig 16:

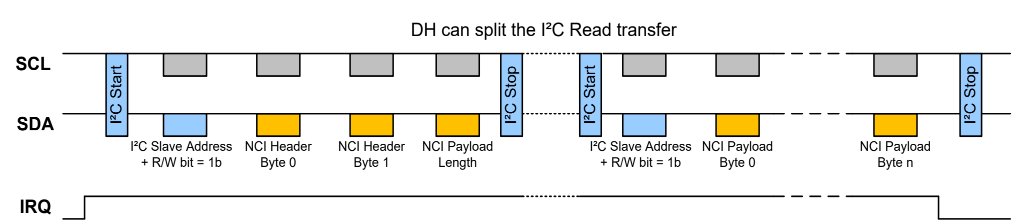

Fig 16. I²C Read sequence with split mode

3.6 Optional transport fragmentation

PN7150 comes with an optional transport fragmentation on I²C, which can be enabled/disabled thanks to bit b4 in *IRQ_POLARITY_CFG* (see →10.1).

This fragmentation can only be used from the DH to the PN7150: there is no fragmentation available from the PN7150 to the DH.

This fragmentation is purely implemented at the I²C transport layer and does not interfere with NCI segmentation, which remains possible on top.

> **!** The I²C fragmentation implemented on PN7150 requires that the DH waits until it has received a Control Message of type Response in response to a Control Message of type Command before it can send any Data Message.
>
> The DH also has to wait until it has received a Credit Notification to release the credit consumed by a previous Data Message it has sent, before it can send a new Control Message of type Command.

3.6.1 Description of the I²C fragmentation:

If the DH has limited capabilities to transport Frames of Bytes over I²C (so below the maximum frame size of an NCI packet which is equal to 258 Bytes), it SHALL send the NCI packet into several fragments, according to the following rules:

- The fragment size has to be an integer multiple of 4 Bytes (excluding the Slave Address Byte required by the I²C protocol).

- The minimum fragment size supported by the DH has to be long enough to transport the following sequence of commands, which is necessary to enable the feature by setting bit b4 in the *IRQ_POLARITY_CFG* parameter (see →10.1):
  - *CORE_RESET_CMD*
  - *CORE_INIT_CMD*
  - *NCI_PROPRIETARY_ACT_CMD*
  - *CORE_SET_CONFIG_CMD*

- To implement a flow control mechanism, the DH has to follow the following sequence:
  1. The DH sends a first fragment of an NCI data packet.
  2. The DH waits for WaitTime = 500µs
  3. The DH writes the [Address & R/Wn] Byte over the I²C bus; it has then to check the I²C ACK bit generated by PN7150:
     a. If the ACK bit is not set, this means that PN7150 is still processing the previous fragment of the NCI packet and it is not yet ready to receive the next fragment. The DH has to wait for an additional WaitTime, moving back to step 2.
     b. If the ACK bit is set, the DH can move to step 4.
  4. The DH transmits the next Fragment
  5. If the whole NCI packet has not yet been transmitted, the DH proceeds to step 2 with another fragment. If the whole NCI packet has been transmitted, the sequence is stopped.

The next figure shows this sequence:

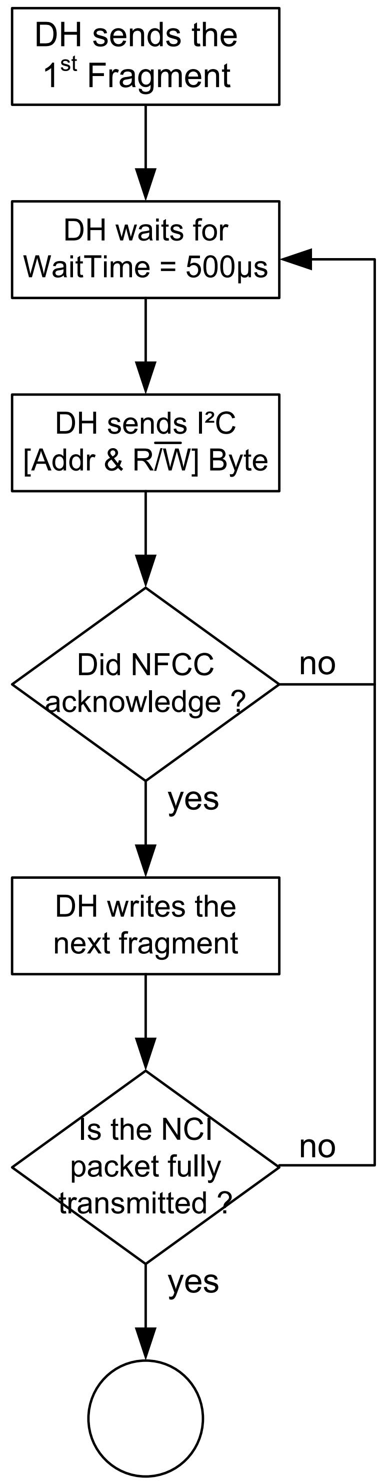

Fig 17. cI²C transport fragmentation algorithm, from DH point of view

3.6.2 Illustration of the I²C fragmentation:

The 2 next figures illustrate a transfer of an NCI message implying I²C fragmentation, with a fragment size of 36 Bytes maximum, when:

- The NCI message fits over a single NCI packet
- The NCI message fits over multiple NCI packets (NCI segmentation is used on top of I²C fragmentation)

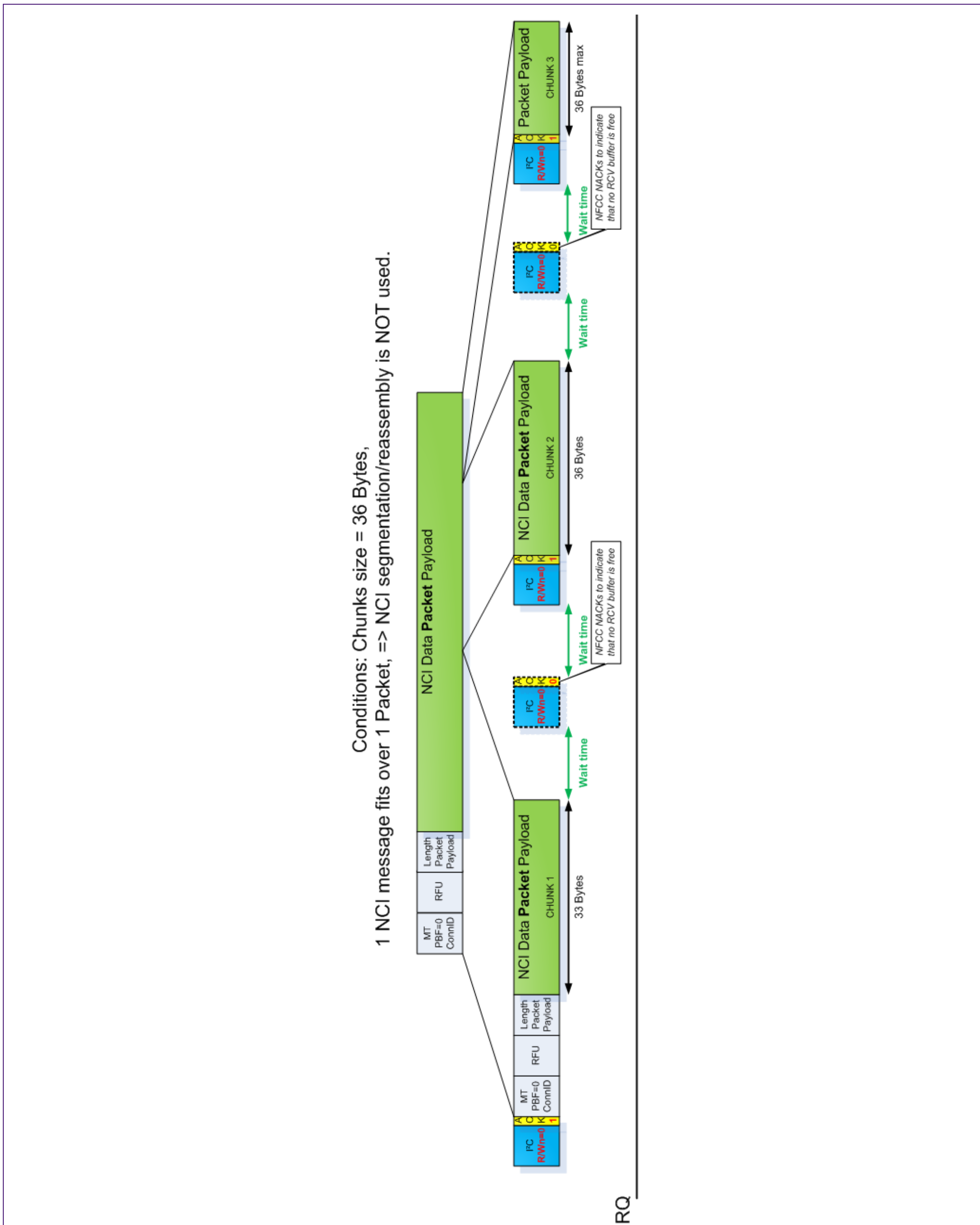

Fig 18. I²C Fragmentation when 1 NCI message = 1 NCI packet

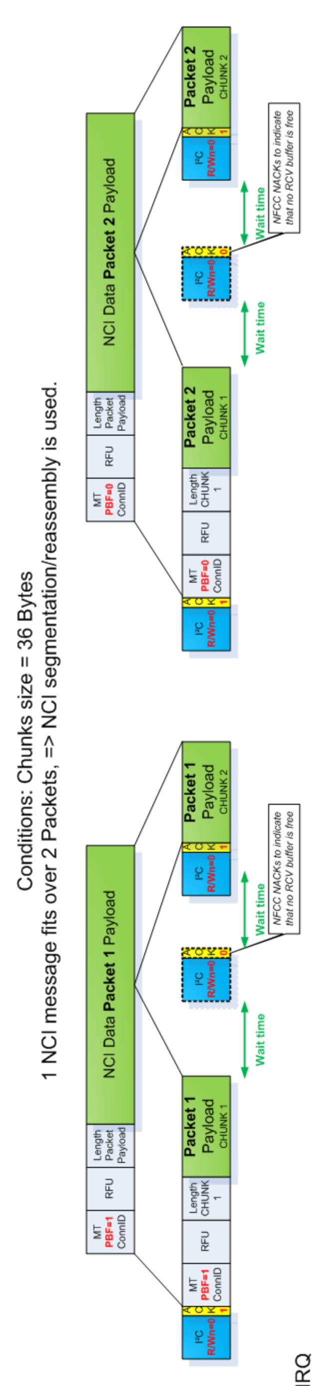

Fig 19. I²C Fragmentation when 1 NCI message is segmented into NCI packets

4. Compliance to [NCI] and PN7150 extensions

The PN7150 is a complex contactless System on Chip, which offers a lot of features. Unfortunately, [NCI] as defined by the NFC FORUM does not give full access to all these features. Therefore, NXP had to extend [NCI] with proprietary extensions, and the PN7150 DH interface which includes [NCI] plus the PN7150 extensions is referenced in the present document as [PN7150-NCI].

So, the following terms are used in the present user Manual with the detailed meaning hereafter:

[NCI]: NCI 1.0 as defined in *NFCForum-TS-NCI-1.0.pdf* available on the NFC FORUM web site.

[PN7150-NCI]: [NCI] + NXP proprietary extensions for the PN7150, in order to allow full access to all the features it offers. NXP tried to use [NCI] as much as possible and to limit the proprietary extensions.

4.1 Feature-based comparison of [NCI] and [PN7150-NCI]

The table below represents the features overview of the PN7150. It highlights the main differences between the NCI standard ([NCI]) and [PN7150-NCI]. The Chapter column contains shortcuts to the section in the document where the feature is described in details.

Table 4. Features overview

Chapter	Features	[NCI]	[PN7150-NCI]
→9	RF Discovery activity (NFC FORUM, EMVCo)	✓ (partially)	✓
→6	Reader/Writer ISO-DEP for NFC-A & NFC-B, T1T, T2T, T3T, T4T	✓	✓
→6	Reader/Writer MIFARE Classic, MIFARE Plus, ISO15693, Kovio, Tag-S	×	✓
→6.3.3	Presence check in Poll mode NFC-A & B	×	✓
→6.3.6	RF bit rates for Poll mode in techno NFC-A & NFC-B: 212kbps, 424kbps & 848kbps	×	✓
→7	Card Emulation ISO-DEP for NFC-A & NFC-B	✓	✓
→7	Card Emulation T3T for NFC-F	✓	✓
→8.1	P2P passive (Initiator & Target)	✓	✓
→8.2	P2P active (Initiator & Target)	×	✓
→10	Configuration: Power management, RF Settings, Clocking schemes	×	✓
→11	Testing: Antenna self-test, PRBS test	×	✓

✓ Partially Covered ✓Covered × Not Covered

4.2 [NCI] Implementation in the PN7150

[NCI] defines several features which are optional or configurable. For instance, data exchange can use an optional flow control, for which the number of credits is defined by the NFCC. So the intent of this section is to describe those features in [NCI] which are optional or configured by the NFCC, to highlight how they are implemented in the PN7150.

4.2.1 Logical connections & credits

Here is a simplified overview of an NFC Device as defined in the NFC FORUM:

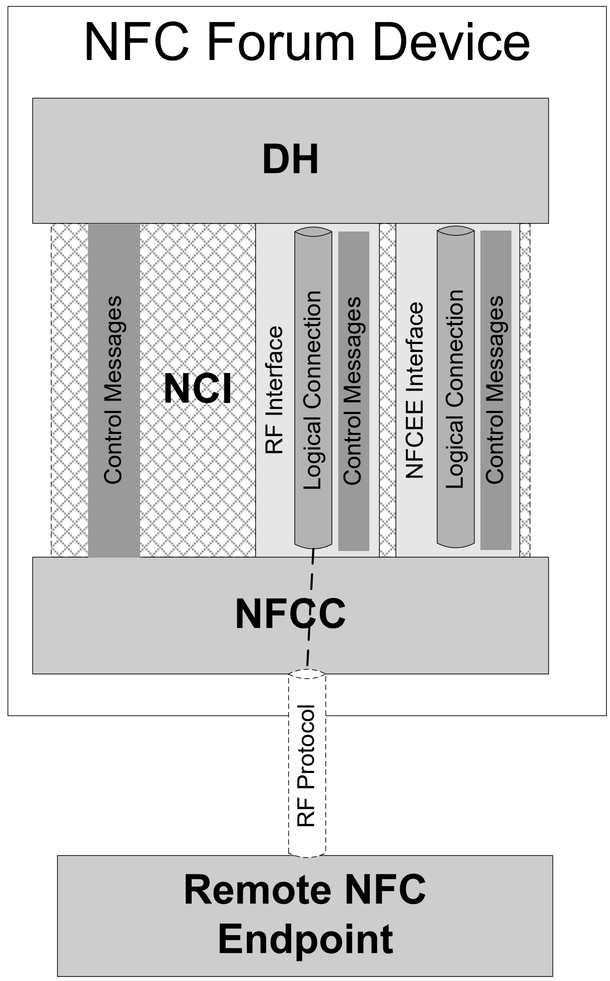

Fig 20. NFC FORUM Device architecture

Logical connections are used to transport data between the DH and the NFCC. Although optional in [NCI], [PN7150-NCI] implements data flow control based on credits management. In order to minimize the required buffer/memory size, the **number of credits is limited to 1** on each logical connection.

The “Max Logical Connections” parameter reported in ***CORE_INIT_RSP*** **equals 0x01 for** [PN7150-NCI]**.** That means that when the DH needs to create a new logical connection, it has first to close the currently opened one, if any.

Here is an overview of the logical connections & credits available in the PN7150:

Table 5. Logical Connections/Credits configuration

Logical connection	Number of connections	Number of credits	Max. Data Packet payload Size
Static RF connection	1	1	[32;255]

4.2.2 Compliance to [NCI] control messages

Here is a detailed status, for the current version PN7150:

Table 6. Status on the compliance to [NCI] control messages

Group	Control messages	Status
CORE	CORE_RESET_CMD / RSP[1] / NTF[2]	Partial Support
	CORE_INIT_CMD / RSP[5]	Full Support
	CORE_SET_CONFIG_CMD / RSP	Full Support
	CORE_GET_CONFIG_CMD / RSP	Full Support
	CORE_CONN_CREATE_CMD / RSP	Partial Support[3]
	CORE_CONN_CLOSE_CMD / RSP	Full Support
	CORE_CONN_CREDITS_NTF	Full Support
	CORE_GENERIC_ERROR_NTF	Full Support
	CORE_INTERFACE_ERROR_NTF	Full Support
RF	RF_DISCOVER_MAP_CMD / RSP	Full Support
	RF_SET_LISTEN_MODE_ROUTING_CMD / RSP	Not supported
	RF_GET_LISTEN_MODE_ROUTING_CMD / RSP / NTF	Not supported
	RF_DISCOVER_CMD / RSP / NTF	Partial Support[4]
	RF_DISCOVER_SELECT_CMD / RSP	Full Support
	RF_INTF_ACTIVATED_NTF	Full Support
	RF_DEACTIVATE_CMD / RSP / NTF	Full Support
	RF_FIELD_INFO_NTF	Full Support
	RF_T3T_POLLING_CMD / RSP / NTF	Full Support
	RF_NFCEE_ACTION_NTF	Full Support
	RF_NFCEE_DISCOVERY_REQ_NTF	Full Support
	RF_PARAMETER_UPDATE_CMD / RSP	Full Support
NFCEE	NFCEE_DISCOVER_CMD / RSP / NTF	Full Support
	NFCEE_MODE_SET_CMD / RSP	Full Support

[1] *CORE_RESET_RSP* will report NCI 1.1, however this is a known limitation and NCI 1.1 is NOT supported.

[2] *CORE_RESET_NTF* has sometimes an additional field, not compliant to [NCI]. See →5.1

[3] The number of Destination Specific parameters is limited to 1

[4] The Discovery Frequency parameter in *RF_DISCOVER_CMD* has no effect in PN7150; whatever the value written by the DH, PN7150 will behave as if it is set to 0x01.

[5] PN7150 wrongly declares in the "NFCC features" field of *CORE_INIT_RSP* that it supports the Discovery Frequency Configuration, although it does not.

> **!** PN7150 comes with a Maximum Control Packet Payload Size of 255 Bytes, as reported in the *CORE_INIT_RSP*. Since [NCI] defines that the maximum size of a Control Message is also 255 Bytes and that the DH has to completely fill a Control Packet when sending a long Control Message, Segmentation and Reassembly cannot be used by the DH with PN7150.

4.2.3 Compliance to [NCI] RF Interfaces

Here is a drawing of the RF interfaces available in [NCI]:

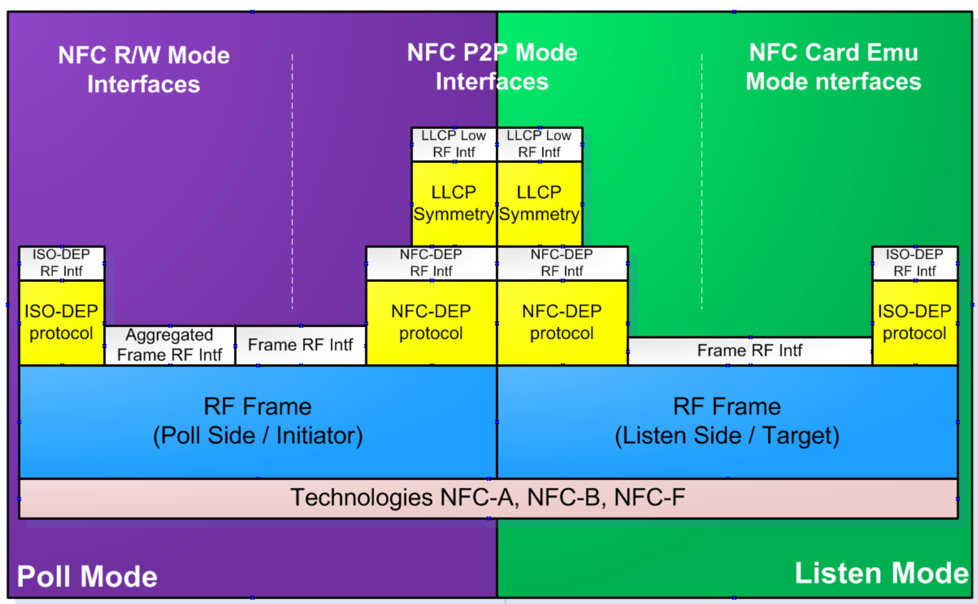

Fig 21. [NCI] RF Interface Architecture

This section details the status on the different RF interfaces supported by the PN7150.

Table 7. NCI Interface limitations

RF Interface present in [NCI]	Status
Poll side & Listen side Frame RF interface	Partial Support[1]
Poll side & Listen side ISO-DEP interface	Full support
Poll side & Listen side NFC-DEP interface	Full support

[1] Frame RF Interface is not supported for P2P Passive & Active modes

4.2.4 Compliance to [NCI] RF Discovery

[NCI] relies on the [ACTIVITY] specification defined by the NFC FORUM.

Since the P2P ACTIVE is not yet included in [ACTIVITY], the corresponding configuration parameters are mentioned as “RFU” in [NCI]. Since the PN7150 supports the P2P ACTIVE mode for both Initiator and Target roles, these parameters are actually used in [PN7150-NCI].

4.2.5 Compliance to [NCI] configuration parameters

[NCI] defines a set of configuration parameters, in [NCI_Table8] (see chapter →15). Most of them are supported by PN7150; however, a subset of these parameters is not supported.

Here is a status for all these parameters, together with their default value in PN7150:

Table 8. Compliance to [NCI] configuration parameters

Config parameters	Status	Coming from	Default value	Behavior if partial/no support
TOTAL DURATION	Partial Support	[NCI]	0x03E8 (1s)	Even if set for more, the total duration is limited to 2.57s
CON_DEVICE_LIMIT	Partial Support	[ACTIVITY]	0x03	Parameter is Read Only, Value is set to 3, except for ISO15693 where it is limited to 2 VICCs
PA_BAIL_OUT	No Support	[ACTIVITY]	-	Bail Out is always activated in Poll/NFC-A
PB_AFI	Full support	[DIGITAL]	0x00	
PB_BAIL_OUT	No Support	[ACTIVITY]	-	Bail Out is always activated in Poll/NFC-B
PB_ATTRIB_PARAM1	Full support	[DIGITAL]	0x00	
PB_SENSB_REQ_ PARAM	No Support	[DIGITAL]	-	No support of advanced features in NFC-B, no support of the extended SENSB_RES.
PF_BIT_RATE	Full support	[DIGITAL]	0x01 (212kbps)	
PF_RC_CODE	Full support	[DIGITAL]	0x00	!! the NCI mechanism to force the parameter to come back to its default value (*CORE_SET_CONFIG* with empty value) does not work for *PF_RC_CODE* !!
PB_H_INFO	Full support	[DIGITAL]	empty	
PI_BIT_RATE	Full support	[DIGITAL]	0x00 (106kbps)	
PN_NFC_DEP_SPEED	Full support	[DIGITAL]	0x00 (106kbps)	
PN_ATR_REQ_GEN_BYTES	Full support	[DIGITAL]	empty	
PN_ATR_REQ_CONFIG	Full support	[DIGITAL]	0x30	
LA_BIT_FRAME_SDD	Full support	[DIGITAL]	0x01	
LA_PLATFORM_CONFIG	Full support	[DIGITAL]	0x00	
LA_SEL_INFO	Full support	[DIGITAL]	0x00	Warning! This value has to be changed to emulate a card in DH with ISO-DEP/NFC-A
LA_NFCID1	Full support	[DIGITAL]	0x08000000	
LB_SENSB_INFO	Full support	[DIGITAL]	0x81	
LB_NFCID0	Full support	[DIGITAL]	0x08000000	

Config parameters	Status	Coming from	Default value	Behavior if partial/no support
LB_APPLICATION_DATA	Full support	[DIGITAL]	Empty	
LB_SFGI	Full support	[DIGITAL]	0x00	
LB_ADC_FO	Full support	[DIGITAL]	0x05	
LF_T3T_ IDENTIFIERS_1..4	Full support	[DIGITAL]	0xFFFF02FE000 000000000FFFFF FFFFFFFFFFF	By default, *LF_T3T_PMM_DEFAULT* is used
LF_T3T_ IDENTIFIERS_5..16	No Support	[DIGITAL]	-	
LF_PROTOCOL_TYPE	Full support	[DIGITAL]	0x02	
LF_T3T_PMM_DEFAULT	Full support	[DIGITAL]	0xFFFFFFFFFFF FFFFF	
LF_T3T_MAX	Full support	[NCI]	0x04	
LF_T3T_FLAGS	Full support	[NCI]	0x0000	
LF_CON_BITR_F	No Support	[DIGITAL]	-	Always both 212 & 424 kbps
LF_ADV_FEAT	No Support	[DIGITAL]	-	No advanced features supported in NFC-F
LI_FWI	Full support	[DIGITAL]	0x04	
LA_HIST_BY	Full support	[DIGITAL]	empty	
LB_H_INFO_RESP	No Support	[DIGITAL]	-	Consequence: the "Higher Layer Response" field in the ATTRIB Response is left empty
LI_BIT_RATE	Full support	[DIGITAL]	0x00 (106kbps)	
LN_WT	Full support	[DIGITAL]	0x08	
LN_ATR_RES_GEN_BYTES	Full support	[DIGITAL]	Empty	
LN_ATR_RES_CONFIG	Full support	[DIGITAL]	0x30	
RF_FIELD_INFO	Full support	[NCI]	0x00	
RF_NFCEE_ACTION	Full support	[NCI]	0x01	
NFCDEP_OP	Full support	[NCI]	0x0F	
NFCC_CONFIG_CONTROL	Full support	[NCI]	0x00	

4.2.6 Compliance to [NCI] data messages

PN7150 is fully compliant to the [NCI] data messages.

4.3 Extensions to [NCI] allowing full control of PN7150

The [PN7150-NCI] extensions section gives a quick overview of the numerous extensions required to [NCI] to give full access to all the features available in the PN7150.

4.3.1 [PN7150-NCI] ext. to [NCI] RF Protocols

PN7150 supports much more protocols than handled today by [NCI].

It is required to extend the [NCI_Table5] defined in [NCI] (see chapter →15) such that these protocols can be configured in various commands/notifications:

Table 9. Proprietary RF protocols

Chapter	Value	Description
→6.4	0x06	PROTOCOL_**15693**
→6.1	0x80	PROTOCOL_**MIFARE_CLASSIC**
→6.5	0x8A	PROTOCOL_**KOVIO**
	0x81-0x89, 0x8B-0x9F, 0xA0-0xFD	Reserved for Proprietary protocols

4.3.2 [PN7150-NCI] ext. to [NCI] Bit Rates in ISO15693 and NFC-F

PN7150 supports the Poll Mode for technology ISO15693. Unfortunately, [NCI] does not define an appropriate bit rate (26kbps) the NFCC has to report to the DH in the *RF_INTF_ACTIVATED_NTF*. NXP has defined a proprietary value for this bit rate.

PN7150 offers the possibility to poll for NFC-F @ 212 kbps and NFC-F @ 424 kbps. Unfortunately, [NCI] only allows configuring one of these 2 bit rates, but not both in the same discovery sequence.

The [NCI] parameter used to configure the bit rate in NFC-F is *PF_BIT_RATE*. By setting *PF_BIT_RATE* to the value of 0x81 "*NFC_BIT_RATE_212 AND NFC_BIT_RATE_424*", polling is done for both 212 and 424k in the same discovery sequence.

Table 10. Proprietary Bit rates

Chapter	Value	Description
→6.4	0x80	NFC_BIT_RATE_26
→6.2	0x81	NFC_BIT_RATE_212 AND NFC_BIT_RATE_424

4.3.3 [PN7150-NCI] ext. to [NCI] RF Interfaces

PN7150 offers some features which are not accessible using the currently defined RF interfaces in [NCI].

So the [NCI_Table6] (see chapter →15) needs to be extended with some proprietary RF interfaces, as described in the table below:

Table 11. RF Interfaces extension

Chapter	New RF Interface	Value	Brief description
→6.1.2	TAG-CMD	0x80	This new interface adds a header to the data payload, in order to encode commands such as: - T2T/MFUL sector select command - MIFARE Classic Authenticate command
		0x81-0xFE	Reserved for proprietary RF Interfaces

4.3.4 [PN7150-NCI] ext. to [NCI] Control messages

This section contains all the additional commands/notifications in [PN7150-NCI].

Table 12. [PN7150-NCI] additional commands/notifications

Chapter	PN7150-NCI Control message	Brief description	Support on PN7150
→5.4	NCI_PROPRIETARY_ACT_CMD/RSP	Command used by the DH to activate the proprietary functions inside the NFCC	Full Support
→6.3.3	RF_ PRES-CHECK_CMD/RSP/NTF	Command used to check if a T4T or an ISO-DEP tag is still in the field.	Full Support
→6.3.4	RF_T4T_SBLOCK_CMD/RSP/NTF	Command used to send S-Block to T4T or ISO-DEP tags	Full Support
→9.4.3	RF_TAG_DETECTOR_TRACE_NTF	Notification to collect the measurements performed by the Tag Detector	Full Support
→9.6.1	CORE_SET_POWER_MODE_CMD/RSP	Command allowing the DH to configure the power mode (standby or idle mode).	Full Support
→10.3	RF_GET_TRANSITION_CMD/RSP	To read out an RF register setting for a given RF Transition	Full Support
→11.2	TEST_PRBS_CMD/RSP	Command allowing the DH to send data over RF at different baud rates in order to verify the contactless part without any interaction with the NCI RF Discovery.	Full Support
→11.3	TEST_ANTENNA_CMD/RSP	Command allowing the DH to check the presence of the antenna components on the PCB.	Full Support
→11.4	TEST_GET_REGISTER_CMD/RSP	Command to receive the Value of the AGC_VALUE_REGISTER	Full Support

[NCI] defines some rules which constraint the use of the control messages. That means that depending on the state the NCI RF State Machine is in, depending on the RF Interface used, depending on some parameters, the control messages are valid or incorrect, and sometimes they trigger state transitions.

NXP has extended these rules for the [PN7150-NCI] extensions.

The following table gives an overview of these rules:

			CURRENT STATE														
			RFST_IDLE		RFST_DISCOVERY		RFST_W4_ALL_DISC.		RFST_W4_HOST_SELECT		RFST_POLL_ACTIVE		RFST_LISTEN_ACTIVE		RFST_LISTEN_SLEEP		
Source	Control Message	parameter / RF Interface	Next state	Response STATUS	Next state	Response STATUS	Next state	Response STATUS	Next state	Response STATUS	Next state	Response STATUS	Next state	Response STATUS	Next state	Response STATUS	According to
NCI 1.0	CORE_RESET_CMD/RSP			Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	NCI 1.0
NCI 1.0	CORE_INIT_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	CORE_SET_CONFIG_CMD/RSP	parameters impacting the RF discovery		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
		other parameters		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NCI 1.0	CORE_GET_CONFIG_CMD/RSP	parameters impacting the RF discovery		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
		other parameters		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NCI 1.0	CORE_CONN_CREATE_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NCI 1.0	CORE_CONN_CLOSE_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NCI 1.0	RF_DISCOVER_MAP_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	RF_SET_LISTEN_MODE_ROUTING_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	RF_GET_LISTEN_MODE_ROUTING_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	RF_DISCOVER_CMD/RSP		DISCOVERY	Status_OK		ALREADY_STARTED		ALREADY_STARTED		ALREADY_STARTED		ALREADY_STARTED		ALREADY_STARTED		ALREADY_STARTED	NCI 1.0
NCI 1.0	RF_DISCOVER_SELECT_CMD/RSP			SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	RF_DEACTIVATE_CMD/RSP/NTF*	Idle_Mode		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	NCI 1.0
		Sleep_Mode / Frame RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	W4_HOST_SELECT	Status_OK	LISTEN_SLEEP	Status_OK		REJECTED	NCI 1.0
		Sleep_Mode / ISO-DEP RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	W4_HOST_SELECT	Status_OK		REJECTED		REJECTED	NCI 1.0
		Sleep_Mode / other RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK		REJECTED		REJECTED		REJECTED	NCI 1.0
		SleepAF_Mode / Frame RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	W4_HOST_SELECT	Status_OK	LISTEN_SLEEP	Status_OK		REJECTED	NCI 1.0
	*: in states POLL_ACTIVE & LISTEN_ACTIVE	SleepAF_Mode / NFC-DEP RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	W4_HOST_SELECT	Status_OK		REJECTED		REJECTED	NCI 1.0
		SleepAF_Mode / other RF Interface		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK		REJECTED		REJECTED		REJECTED	NCI 1.0
		Discovery		SEEMANTIC	IDLE	Status_OK	IDLE	Status_OK	IDLE	Status_OK	DISCOVERY	Status_OK	DISCOVERY	Status_OK		REJECTED ?	NCI 1.0
NCI 1.0	RF_T3T_POLLING_CMD/RSP	Frame RF Interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC	NCI 1.0
		other RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	RF_PARAMETER_UPDATE_CMD/RSP	Frame RF Interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		Status_OK		SEEMANTIC	NCI 1.0
		other RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NCI 1.0
NCI 1.0	NFCEE_DISCOVER_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	?
NCI 1.0	NFCEE_MODE_SET_CMD/RSP			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	?
NXP	*CORE_SET_POWER_MODE_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*CORE_GET_POWER_MODE_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*NCI_PROPRIETARY_ACT_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*RF_T4T_SBLOCK_PARAM_CMD/RSP*	ISO-DEP RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC	NXP
		other RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*RF_PRES-CHECK_CMD/RSP*	ISO-DEP & NFC-DEP RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC	NXP
		other RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*RF_MF_PROX-CHECK_CMD/RSP*	ISO-DEP RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		Status_OK		SEEMANTIC		SEEMANTIC	NXP
		other RF interface		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*TEST_RF_FIELD_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*TEST_SWP_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*TEST_ANTENNA_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP
NXP	*TEST_PRBS_CMD/RSP*			Status_OK		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC		SEEMANTIC	NXP

Fig 22. CMDs/RSPs versus the current state of the NCI RF State Machine

			CURRENT STATE														
			RFST_IDLE		RFST_DISCOVERY		RFST_W4_ALL_DISC.		RFST_W4_HOST_SELECT		RFST_POLL_ACTIVE		RFST_LISTEN_ACTIVE		RFST_LISTEN_SLEEP		
Source	**Control Message**	**parameter**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**Next state**	**NTF expected in this state ?**	**According to**
NCI 1.0	CORE_RESET_NTF			Yes	IDLE	Yes	IDLE	Yes	IDLE	Yes	IDLE	Yes	IDLE	Yes	IDLE	Yes	NCI 1.0
NCI 1.0	RF_DISCOVER_NTF	Notif. Type = 2		No	W4_ALL_DISC	Yes		Yes		No		No		No		No	NCI 1.0
		Notif. Type = 0/1		No		No	W4_HOST_SELECT	Yes		No		No		No		No	NCI 1.0
NCI 1.0	RF_DEACTIVATE_NTF	Idle_Mode		No		No		No		No	IDLE	Yes	IDLE	Yes		No	NCI 1.0
		Sleep_Mode		No		No		No		No	W4_HOST_SELECT	Yes	LISTEN_SLEEP	Yes		No	NCI 1.0
		SleepAF_Mode		No		No		No		No	W4_HOST_SELECT	Yes	LISTEN_SLEEP	Yes		No	NCI 1.0
		Discovery		No		No		No		No	DISCOVERY	Yes	DISCOVERY	Yes	DISCOVERY	Yes	NCI 1.0
NCI 1.0	RF_INTF_ACTIVATED_NTF	Poll Mode		No	POLL_ACTIVE	Yes		No	POLL_ACTIVE	Yes		No		No		No	NCI 1.0
		Listen Mode		No	LISTEN_ACTIVE	Yes		No		No		No		No	LISTEN_ACTIVE	Yes	NCI 1.0
NCI 1.0	NFCEE_DISCOVER_NTF			Yes		Yes		Yes		Yes		Yes		Yes		Yes	NCI 1.0
NCI 1.0	CORE_INTERFACE_ERROR_NTF	Loop-Back logical connection		Yes		No		No		No		No		No		No	NXP
		RF static logical connection		No		No		No		No		Yes		Yes		No	NCI 1.0
		NFCEE logical connection		Yes		Yes		Yes		Yes		Yes		Yes		Yes	NCI 1.0
NCI 1.0	RF_NFCEE_ACTION_NTF			No		No		No		No		No		Yes		No	NCI 1.0
NCI 1.0	CORE_CONN_CREDITS_NTF	Loop-Back logical connection		Yes		No		No		No		No		No		No	NXP
NCI 1.0		RF static logical connection		No		No		No		No		Yes		Yes		No	NCI 1.0
NCI 1.0		NFCEE logical connection		Yes		Yes		Yes		Yes		Yes		Yes		Yes	NCI 1.0
NCI 1.0	CORE_GENERIC_ERROR_NTF	NCI 1.0 Status		No		Yes		No		Yes		No		No		No	NCI 1.0
NXP		***NXP Status***		Yes		Yes		Yes		Yes		Yes		Yes		Yes	NCI 1.0
NXP	***RF_T4T_SBLOCK_PARAM_NTF***			No		No		No		No		Yes		No		No	NXP
NXP	***RF_PRES-CHECK_NTF***			No		No		No		No		Yes		No		No	NXP
NXP	***RF_MF_PROX-CHECK_NTF***			No		No		No		No		Yes		No		No	NXP
NXP	***CLOCK_REQ_NTF***			No		Yes		No		No		No		No		No	NXP
NXP	***RF_COLLISION_NTF***	NFC FORUM profile		No		No		No		No		No		No		No	NXP
		EMVCo Profile		No		Yes		No		No		No		No		No	NXP
NXP	***TEST_SWP_NTF***			Yes		No		No		No		No		No		No	NXP

Fig 23. NTFs versus the current state of the NCI RF State Machine

PN7150 defines additional states to the RF state machine defined in [NCI_Chap2], to ensure a correct implementation of the “atomic behavior” of the pair of commands made by *CORE_RESET_CMD* & *CORE_INIT_CMD* and also to correctly handle wrong RF protocol to RF interface mapping through the *RF_DISCOVER_MAP_CMD*. The drawing below illustrates these additional states, linked to the [NCI]-defined RFST_IDLE:

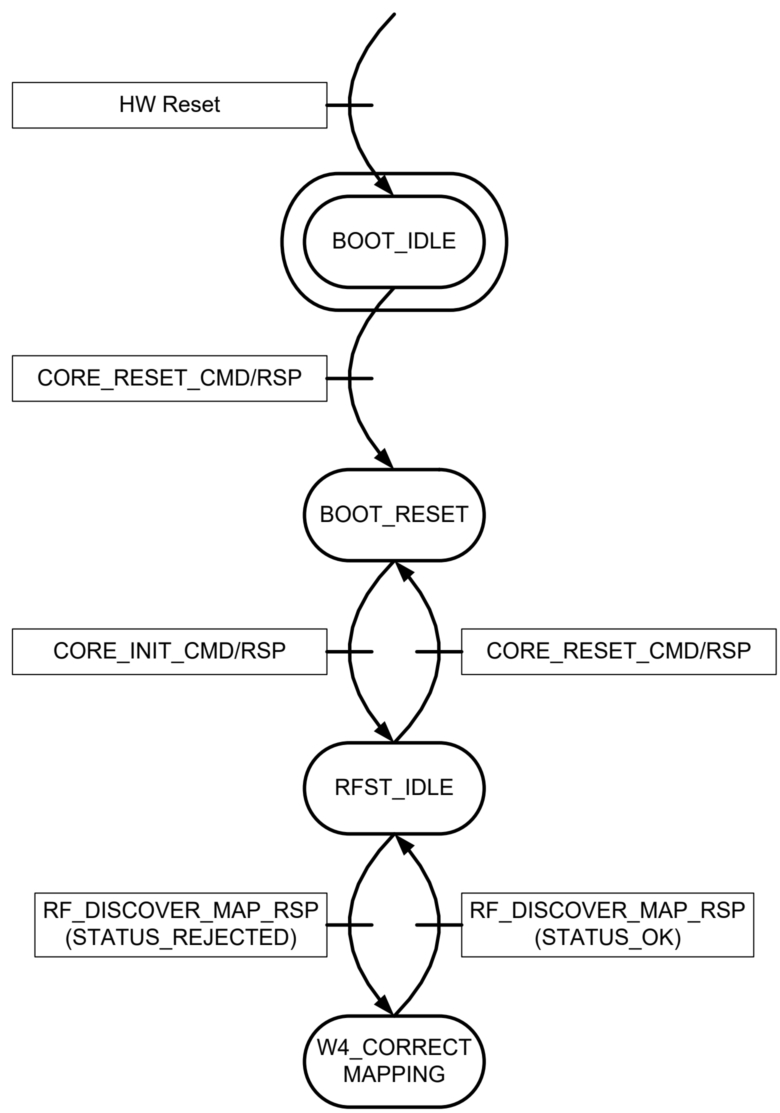

Fig 24. States added to the [NCI] State Machine.

4.3.5 [PN7150-NCI] ext. to [NCI] Configuration parameters

[NCI] lists a number of parameters, which are necessary to set up the RF discovery. But the PN7150 requires a lot more parameters, for instance to configure some RF protocols which are not supported by [NCI], to configure the power & clock management etc…

Here is a list of sets of parameters, sorted out by features to configure:

Table 13. Overview of additional Configuration parameters

Chapter	Feature to configure	Comment
→10.1	System	Parameters allowing the DH to configure the System: Clock management, IRQ and CLOCKREQ pins management, MIFARE Classic Keys handling…
→10.2	RF Discovery	Parameters allowing the DH to configure the Discovery activity (Tag Detector, Discovery profile between: NFC FORUM, NFC FORUM+ and EMVCo etc…).
→10.3	Contactless Front-End	Parameters allowing the DH to configure all internal HW settings in the Contactless InterFace (CIF).

4.3.6 [PN7150-NCI] ext. to [NCI] proprietary parameters space

[NCI] defines a parameter space with a size of 255 parameters, in which around 100 tags are allocated for proprietary parameters:

Table 14. Parameter space

Parameters space sub-sections	Tag
Assigned & reserved for NCI 1.0	0x00-0x9F
Reserved for Proprietary Use	0xA0-0xFE
RFU (Reserved for Extension)	0xFF

Regarding the PN7150 needs, this reserved area is not sufficient. To extend this space, the solution chosen is to define a space of Tags coded on 16 bits, instead of 8 bits.

These extended Tags will always start by the value 0xA0, which is the first value available in the Proprietary range.

This allows adding 256 new parameters.

Remark: If this is not sufficient in the future, we might use 16-bit tag values starting by 0xA1, 0xA2 etc…

Table 15. Extended TLV for proprietary parameters

Payload	Field	Length	Description
m+3 Octets	Tag = 0xA0XX	2 Octet	Extended tag identifier.
	Len	1 Octet	The length of Val (m).
	Val	m Octets	Value of the configuration parameter.

This is illustrated by the following picture:

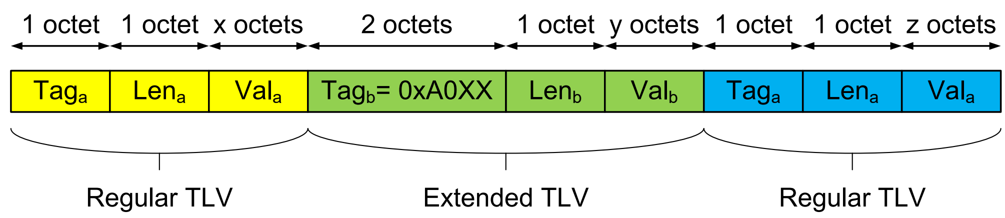

Fig 25. Regular & Extended TLVs comparison

4.3.7 [PN7150-NCI] ext. to [NCI] Status Codes

[NCI] defines a set of standard Status Codes in [NCI_Table1] (see chapter →15).

NXP has extended this set of status codes with the following values:

Table 16. Proprietary Status Codes

Status code	Description	Used in
0xE0	STATUS_DO_NOT_REPLY	CORE_GENERIC_ERROR_NTF
0xE1	STATUS_BOOT_TRIM_CORRUPTED	CORE_GENERIC_ERROR_NTF
0xE2	STATUS_PMU_DCDC_OVERLOAD	CORE_GENERIC_ERROR_NTF
0xE3	STATUS_PMU_TXLDO_OVERCURRENT	CORE_GENERIC_ERROR_NTF
0xE4	STATUS_EMVCO_PCD_COLLISION	CORE_GENERIC_ERROR_NTF

4.3.8 [PN7150-NCI] ext. to [NCI] Reason Code in CORE_RESET_NTF

[NCI] defines a set of standard Reason Codes in the CORE_RESET_NTF. Please refer to [NCI_Table9] (see chapter →15).

NXP has extended this set of reason codes with the following value:

Table 17. Proprietary Reason Codes in CORE_RESET_NTF

Reason Code	Description
0xA0	An assert has triggered PN7150 reset/reboot
0xA1	An over temperature has triggered the reset of PN7150
0xE6	An anti-tearing recovery mechanism succeed during boot

4.3.8.1 Internal Assert

In case of internal assert during PN7150 internal MCU execution, PN7150 autonomously reboots and issues an CORE_RESET_NTF notification with reason code ‘0xA0’, out of [NCI] compliance. Indeed, PN7150 appends one parameter at the end of the frame, providing information for debug purposes. The CORE_RESET_NTF frame format then becomes:

Table 18. CORE_RESET_NTF when reason code ‘0xA0’ is used

Payload Field(s)	Length	Description	Default
Reason Code	1 Octet	0xA0: NXP proprietary	0xA0
Config. Status	1 Octet	See [NCI]	
dwAssertionProgramCounter	4 Octets	Program counter for assertion	

4.3.8.2 Over temperature protection

PN7150 implements a monitoring of the internal temperature to prevent IC being damaged. When the temperature reach the limit PN7150 autonomously reboots and issues an CORE_RESET_NTF notification with reason code ‘0xA1’.

4.3.8.3 Anti-tearing recovery mechanism

PN715 implements a recovery mechanism in case of internal EEPROM memory corruption. Indeed, in case EEPROM memory gets corrupted (refer to chapter →10.3 for detailed information), at next boot PN7150 issues an CORE_RESET_NTF notification with reason code ‘0xE6’ after sending CORE_RESET_RSP (as response to first CORE_RESET_CMD sent by the DH).

> **!** This recovery mechanism induces all RF settings being reset to their default values, thus DH must handle this notification to trigger re-applying RF settings expected values during the initialization sequence (see chapter →10.3).

4.3.9 [PN7150-NCI] ext. to [NCI] RF Technology & Mode

PN7150 supports more RF Technology & Mode parameters than handled today by [NCI].

It is required to extend the [NCI_Table3] defined in [NCI] (see chapter →15) such that these RF Technology & Mode parameters can be used in RF_DISCOVER_CMD:

Table 19. Proprietary RF Technology & Mode parameters

Chapter	Value	Description
→6.5	0x70	NFCA_KOVIO_POLL_MODE
	0x71-0x76 0x78-0x7F	Reserved for Proprietary Technologies in Poll Mode

5. Initialization & Operation configuration

5.1 Reset / Initialization

[NCI] defines a Reset/Initialization sequence, which is based on two different commands:

- ⇨ CORE_RESET_CMD
- ⇨ CORE_INIT_CMD

These two commands have to be called by the DH in an "atomic" way: there cannot be any other command in-between and the PN7150 operation cannot start any operation (Reader/Writer, Card Emulation, P2P, Combined modes etc…) if it does not first receive these 2 commands.

[NCI] defines 2 modes for the Reset command: Keep Configuration & Reset Configuration. Here is the detail of the difference between the 2 reset modes:

Table 20. Comparison of the 2 Reset Modes

Features	Reset Configuration	Keep Configuration
CPU reboot	Yes	Yes
NCI Configuration parameters	Back to default	Kept
Proprietary Configuration parameters	Kept	Kept
Interface Mapping Table	Lost	Kept
Discovery activity	Lost	Lost

! PN7150 may delay the CORE_RESET_RSP

If the DH sends a CORE_RESET_CMD while PN7150 has already indicated that it has some data available to be read by the DH (IRQ pin activated), the DH has first to read the data available from PN7150 before it can get the CORE_RESET_RSP. The reason is that the NCI output buffer in PN7150 needs to be flushed before PN7150 can apply a Reset and then send the CORE_RESET_RSP.

5.2 Manufacturer Specific Information in [NCI] CORE_INIT_RSP

The NCI command CORE_INIT_RSP contains a field "Manufacturer Specific Information" with 4 bytes.

Here are the values of these 4 Bytes:

Table 21. Manufacturer specific information in CORE_INIT_RSP

Byte	Meaning	Condition to increment
0	Hardware Version number	New silicon
1	ROM Code Version number	New ROM Code
2	Firmware Major version	New Firmware, adding features
3	Firmware Minor version	New Firmware, solving bugs on existing features.

! PN7150B0HN/C11006 exposes FW version "01.AE", while previous IC versions exposes FW version '01.A0".

5.3 Whole sequence to prepare the PN7150 operation

After the Reset/Initialization sequence is passed, the PN7150 requires several other steps before it is ready to start operating as a Reader/Writer, Card Emulator etc…

The simplest case is when the DH issues a CORE_RESET_CMD with Reset Type = Keep Configuration.

On this figure,

⇨ Green background means mandatory exchange

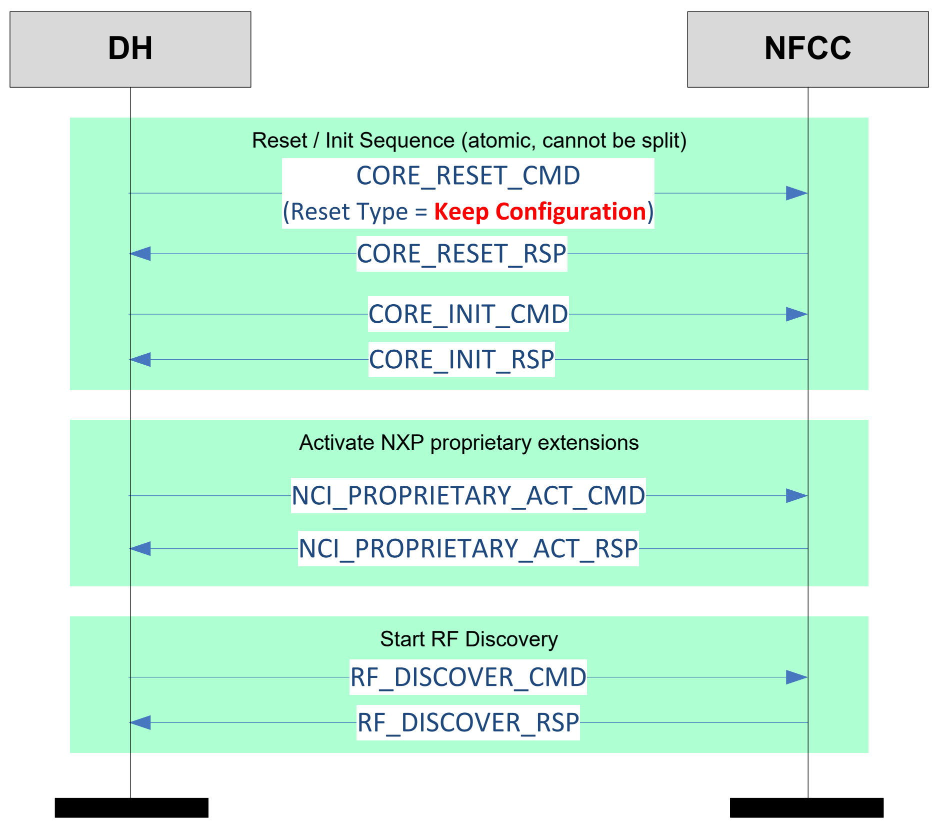

Fig 26. Initialization sequence to prepare the PN7150 operation (Keep Configuration)

Now, here is the figure which lists the complete sequence, starting by a Reset Command based on Reset Type = Reset Configuration. Since the entire configuration is lost, the PN7150 needs to be reconfigured and various optional steps are added, which might be needed or not, depending on the use case.

On this figure,

⇨ Green background means mandatory exchange

⇨ Blue background means optional exchange, depending on the use case.

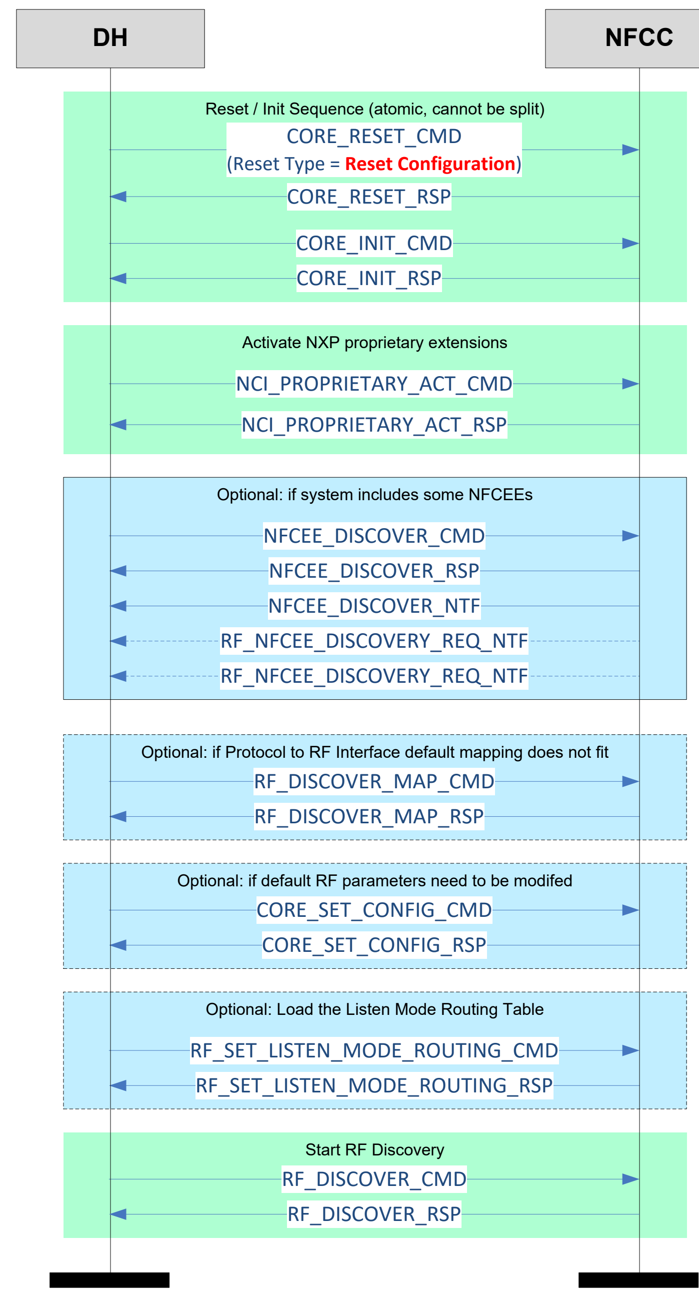

Fig 27. Initialization sequence to prepare the PN7150 operation (Reset configuration)

5.4 Proprietary command to enable proprietary extensions

It is visible on the previous flow chart that NXP has introduced a proprietary command sent by the DH to enable the proprietary extensions to [NCI], which are available in the PN7150. So, when the PN7150 receives this command NCI_PROPRIETARY_ACT_CMD, it knows that the DH is aware of the proprietary extensions and may therefore send proprietary notifications (see the list in Table 12). If the PN7150 does not receive this proprietary command, it knows that the DH do not understand proprietary extensions and will therefore not send any proprietary notifications.

Here is the description of this command:

Table 22. NCI_PROPRIETARY_ACT_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x02	0	DH informs the PN7150 that it knows the proprietary extensions.

Table 23. NCI_PROPRIETARY_ACT_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x02	2	PN7150 indicates that it understood the command.

Table 24. NCI_PROPRIETARY_ACT_RSP parameters

Payload Field(s)	Length	Value/Description	
STATUS	1 Octet	One of the following Status codes, as defined in [NCI_Table1]	
		0x00	STATUS_OK
		0x03	STATUS_FAILED
		Others	Forbidden
FW_Build_Number	4 Octets	NXP internal firmware build number	

5.5 Configuration template

In order to help the user of the PN7150 to issue the right configuration sequence for a given mode of operation, the present document will detail a typical configuration sequence, based on the following template:

Table 25. Template for a typical configuration sequence

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	…
	Mode	…
	RF Interface	…
CORE_SET_CONFIG_CMD	*Depends on technology & mode*	…
RF_DISCOVER_CMD	RF Technology & Mode	…

5.6 PLL input Clock Management

The PN7150 is flexible in terms of clock sources. It can be either:

- a 27.12MHz quartz
- or a clean clock signal available on the platform on which PN7150 is connected. A PLL inside PN7150 will convert this input clock signal into an internal 27.12MHz used to generate the RF carrier. The input clock frequency has to be one of the predefined set of input frequencies: 13MHz, 19.2MHz, 24MHz, 26MHz, 38.4MHz and 52MHz.

The DH has to configure the parameter CLOCK_SEL_CFG (see chapter →10.1) to configure what is the clock source as used in the current application.

Table 26. Clock sources supported

Name	Description
XTAL	To be selected when a 27.12MHz quartz is used as a clock source
PLL	To be selected when an input clock is provided to PN7150, with a frequency equal to either 13MHz, 19.2MHz, 24MHz, 26MHz, 38.4MHz or 52MHz

The same parameter (CLOCK_SEL_CFG) is used to configure which clock frequency is used as an input to the PLL when this is the clock source in use.

In order to optimize system power consumption, it may be required to switch OFF the PLL input clock when the PN7150 does not have to generate the 13.56MHz RF carrier.

A dedicated pin (CLKREQ) is used to inform the DH or a clock generating chip that the PN7150 requires to get the PLL input clock, such that it can generate the 13.56MHz RF carrier. PN7150 assumes that the PLL input clock is on and stable after a programmable time-out, which is configured thanks to the parameter CLOCK_TO_CFG (see chapter →10.1).

5.7 Transmitter voltage Configurations

The PN7150 supports 2 different configurations, called CFG1 and CFG2.

5.7.1 CFG1: Transmitter supply voltage from battery supply

In CFG 1 VBAT1 and VBAT2 are connected to the Battery and between 2.3V and 5.5V.

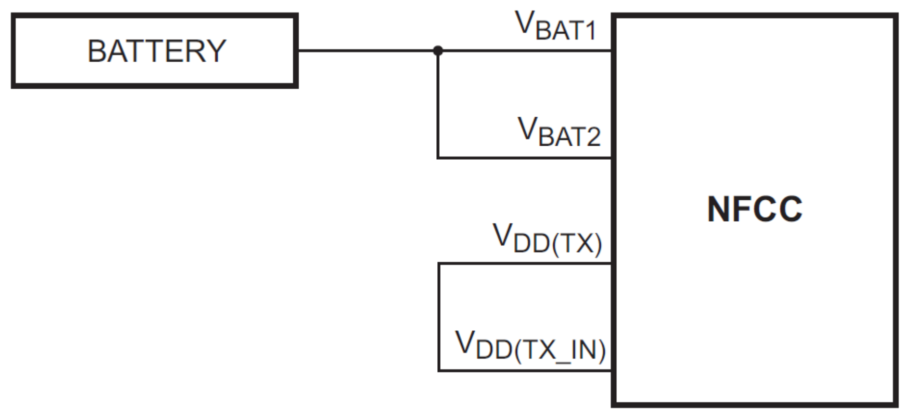

Fig 28. CFG1: VBAT1 = VBAT2 = 2.3V to 5.5V

This configuration is enabled by appropriate setting of *PMU_CFG* parameter. In addition *TVDDReqTime* parameter shall be set to 0x00 (see configuration chapter →10.1).

5.7.2 CFG2: Transmitter supply voltage from external 5V supply

In CFG 2 VBAT1 is connected to 5V while VBAT2 is connected to the battery (delivering between 2.3V and 5.5V). The internal TXLDO is used to generate a transmitter supply voltage of 4.7V from the external 5V.

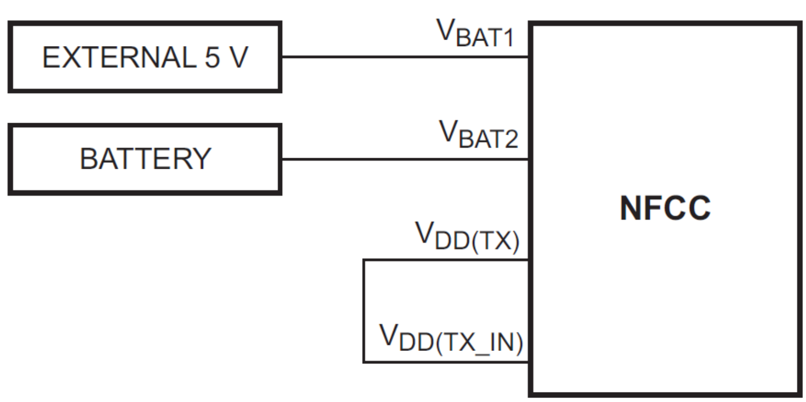

Fig 29. CFG2: VBAT1 = 5V, VBAT2 = 2.3V to 5.5V

This configuration is enabled by appropriate setting of *PMU_CFG* parameter (see configuration chapter →10.1).

6. Reader/Writer Mode

6.1 T1T, T2T, MIFARE Ultralight, MIFARE Classic and MIFARE Plus tags

Note: all the Tags/Cards in this category are based on NFC-A technology, but they do not support the ISO-DEP Protocol.

MIFARE Plus cards support the ISO-DEP protocol, but only when they are configured in Security Level3, which is out of scope for this section.

6.1.1 Access through the [NCI] Frame RF Interface

[NCI] allows the data exchange with tags T1T, T2T using the Frame RF Interface.

Most of the commands of the MIFARE Classic and MIFARE Plus can also be mapped on the Frame RF Interface, but NXP decided to use a separate RF interface (TAG-CMD, see →6.1.2) because some MIFARE Classic commands are split in 2 steps (e.g. Authenticate command) and have a tight response timeout (about 1ms) which can hardly be monitored by the DH through the NFCC.

Here is a summary of the Tags/Card based on technology NFC-A which can be accessed through the Frame RF interface

Table 27. Tag/Cards accessible over the [NCI] Frame RF Interface

Tag/Card	Access through the Frame RF Interface
T1T	✓
T2T	✓
MIFARE Ultralight, Ultralight C	✓
MIFARE Classic	×
MIFARE Plus for Security levels 1 & 2	×

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for T1T & T2T through the Frame RF Interface:

Table 28. Config. seq. for R/W of T1T or T2T through the Frame RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD*	RF Protocol (choose between the 2 possible protocols)	PROTOCOL_T1T
		PROTOCOL_T2T
	Mode	Poll
	RF Interface	Frame RF Interface
CORE_SET_CONFIG_CMD	PA_BAIL_OUT*	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_POLL_MODE

* Note: RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

* this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

6.1.2 [PN7150-NCI] extension: TAG-CMD Interface

In addition to the incompatibility of the Frame RF Interface with the MIFARE Classic Authenticate command described in the previous chapter, the intention when introducing the TAG-CMD interface was to add some commands such as ReadN/WriteN which would allow to read/write multiple bytes, and would rely on the NFCC to call several times the basic read/write commands defined in the T1T, T2T or MIFARE Classic protocols. Unfortunately, we had to withdraw this concept and the TAG-CMD as implemented in PN7150 is limited to MIFARE Classic operation in Reader/Writer and T2T operation in Reader/Writer when the Sector Select command is required.

The figure bellow represents the location of the TAG-CMD RF Interface:

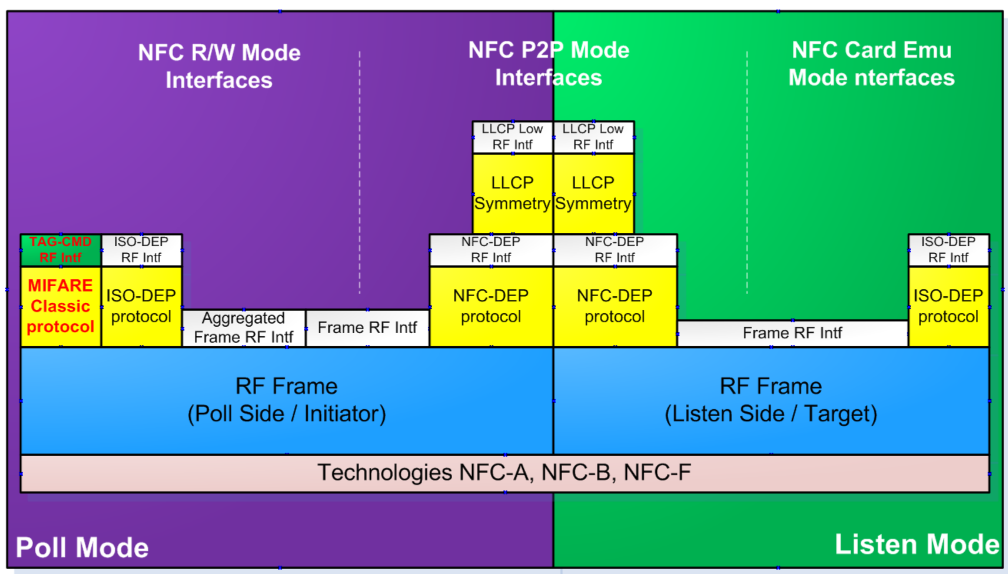

Fig 30. TAG-CMD RF Interface

6.1.3 [PN7150-NCI] extension: Payload structure of the TAG-CMD RF Interface

The TAG-CMD RF Interface is using the same data mapping as the one defined for the [NCI] Frame RF Interface (see section 8.2.1 in [NCI]). However, for the TAG-CMD RF Interface, the Payload is defined differently.

Two different structures are defined:

1. REQ (requests) : these are commands from the DH to the NFCC
2. RSP (responses): these are responses from the NFCC to the DH.

The diagram below details how the Payload is modified to insert a header, which carries the REQ ID or the RSP ID and some parameters, if required.

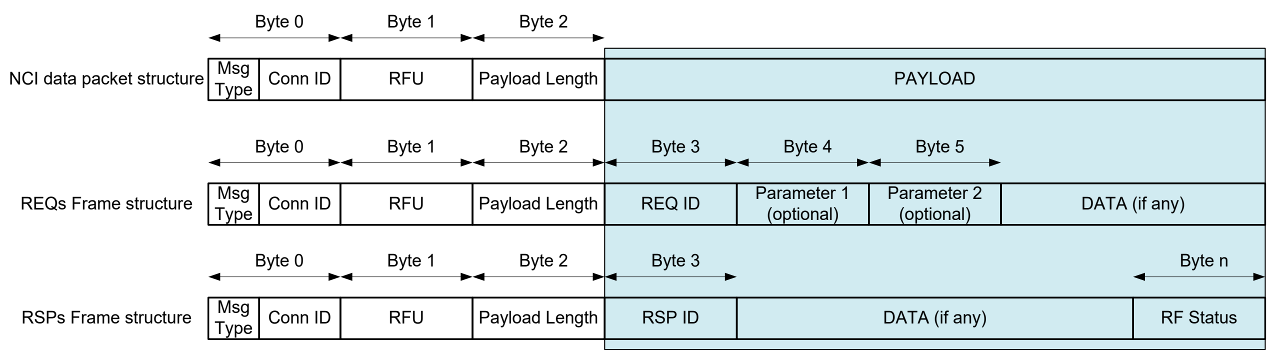

Fig 31. Data message payload for the TAG-CMD Interface

Note: REQs and RSPs don't share exactly the same structure:

REQs: Although illustrated with 2 parameters on the figure above, REQs may have no parameters or only one. Some REQuests might also need parameters bigger than 1 Byte. Parsing The REQ ID is the way to know how many parameters follow and how long they are.

RSPs: there are no parameters in ReSPonses. A Byte is added at the end of the payload (after the DATA field) to inform the DH on the RF status (to report RF errors if they were some). The Status codes used are the following:

Table 29. TAG-CMD RF Status code

Value	Description
0x00	STATUS_OK
0x03	STATUS_FAILED
0xB0	RF_TRANSMISSION_ERROR
0xB1	RF_PROTOCOL_ERROR
0xB2	RF_TIMEOUT_ERROR
Others	Forbidden

6.1.4 [PN7150-NCI] extension: REQs & RSPs rules

A REQ command is always going from DH to RF, through the NFCC.

A RSP response is always going from the RF to the DH, through the NFCC

The DH SHALL wait until it has received a RSP associated to a REQ before it can send a new REQ.

6.1.5 [PN7150-NCI] extension: List of REQs & RSPs

In this section, the following acronyms are used:

Table 30. Acronyms definition

Acronym	Description
T1T	NFC FORUM Type 1 Tag (based on Topaz/Jewel)
MF	MIFARE family, not ISO-DEP compliant, including T2T, MIFARE Ultralight (std or C), MIFARE Classic and MIFARE Plus for Security Level 1 & 2.
MFC	MIFARE Classic and MIFARE Plus for Security Level 1 & 2.

The added REQuests/ReSPonses pairs are listed in the following table:

Table 31. List of REQuests & ReSPonses

REQ/RSP Name	ID	Param 1	Param 2	Param 3	Data	Description
XCHG_DATA_REQ	0x10	None	None	None	Yes	MFC: DH sends Raw data to the NFCC, which encrypts them before sending them to MFC. T1T/T2T: DH sends Raw data to the NFCC, which forwards them in plain to the Tag.
XCHG_DATA_RSP	0x10	N/A	N/A	N/A	Yes	MFC: DH gets Raw data once RF data from MFC are decrypted by the NFCC, if successful. T1T/T2T: DH gets Raw plain data once the NFCC receives RF data from the Tag, if successful.
MF_SectorSel_REQ	0x32	Sector Address	None	None	No	T2T & MFU only: DH Sends the address of the Block to select.
MF_SectorSel_RSP	0x32	N/A	N/A	N/A	No	T2T & MFU only: DH gets the “Sector Select” response status
MFC_Authenticate_REQ	0x40	Sector Address	Key Selector	Key (optional)	No	DH asks NFCC to perform MFC Authenticate command.
MFC_Authenticate_RSP	0x40	N/A	N/A		No	DH gets the MFC Authenticate command status

All these REQs & RSPs are detailed in the next sections.

6.1.6 [PN7150-NCI] extension: raw data exchange REQs & RSPs

Table 32. XCHG_DATA_REQ

REQ_ID	REQ Name	Number of parameter(s)	Presence of data	Description
0x10	XCHG_DATA_REQ	0	Yes	MFC: DH sends Raw data to the NFCC, which encrypts them before sending them to MFC. T1T/T2T: DH sends Raw data to the NFCC, which forwards them in plain to the Tag.

Table 33. XCHG_DATA_RSP

RSP_ID	RSP Name	Presence of Data	Description
0x10	XCHG_DATA_RSP	Yes	MFC: DH gets Raw data once RF data from MFC are decrypted by the NFCC, if successful. T1T/T2T: DH gets Raw plain data once the NFCC receives RF data from the Tag, if successful. If the response from the MF tag in the field is an ACK or a NACK, the ACK/NACK is also sent back to the DH inside the Data field. Since ACK & NACK are 4-bit commands, they are transported on the 4 LSBs of the data Byte; the 4MSBs of that Byte are forced to the logical '0' value.

6.1.7 [PN7150-NCI] extension: T2T & MFU REQs & RSPs

All the REQs & RSPs described in this section can be used whatever the tag between:

- T2T
- MIFARE Ultralight (std or C)

Table 34. MF_SectorSel_REQ

REQ_ID	REQ Name	Number of parameter(s)	Presence of data	Description
0x32	MF_SectorSel_REQ	1	No	DH Sends the address of the Sector to select.

Table 35. MF_SectorSel_REQ parameter

	Parameter	Length (Byte)	Value	Description
1	Sector Address	1	?	Defines the address of the sector which has to be selected. The address can be any block address in this sector.

Table 36. MF_SectorSel_RSP

RSP_ID	RSP Name	Presence of Data	Description
0x32	MF_SectorSel_RSP	No	DH gets sector select status

6.1.8 [PN7150-NCI] extension: MIFARE Classic REQs & RSPs

Table 37. MFC_Authenticate_REQ

REQ_ID	REQ Name	Number of parameter(s)	Presence of data	Description
0x40	MFC_Authenticate_REQ	3	No	DH asks NFCC to perform MFC authenticate.

Table 38. MFC_Authenticate_REQ parameters

<table>
<tr><th></th><th>Parameter</th><th>Length (Byte)</th><th>Value</th><th>Description</th></tr>
<tr><td>1</td><td>Sector Address</td><td>1</td><td></td><td>Address of the sector to authenticate</td></tr>
<tr><td>2</td><td>Key Selector</td><td>1</td><td>N/A</td><td>
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Key A (‘0’) or Key B (‘1’)</td></tr>
<tr><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td>0 => use pre-loaded key
1 => use Key in param Nbr 3</td></tr>
<tr><td></td><td></td><td></td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>Pre-loaded key number (0 to 15)</td></tr>
<tr><td></td><td>0</td><td>0</td><td></td><td></td><td></td><td></td><td></td><td>RFU</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>Embedded Key (optional)</td><td>6</td><td>N/A</td><td>This parameter is present in the MFC_Authenticate_CMD only if bit b4 is set to logical '1' in Key Selector parameter. If present, this parameter defines the value of the Key used for the Authentication.</td></tr>
</table>

Table 39. MFC_Authenticate_RSP

RSP_ID	RSP Name	Presence of Data	Description
0x40	MFC_Authenticate_RSP	No	DH gets the “authenticate” cmd status

Table 40. TAG-CMD RF Status code, in the special case of MFC_Authenticate_CMD

Value	Description	Reason
0x00	STATUS_OK	Authentication was successful
0x03	STATUS_FAILED	Authentication failed (wrong key, time-out triggered during authentication etc…)
0xB0	RF_TRANSMISSION_ERROR	Not used
0xB1	RF_PROTOCOL_ERROR	Not used
0xB2	RF_TIMEOUT_ERROR	Not used
Others	Forbidden	

Once a sector is authenticated, PN7150 will automatically encrypt any data sent by the DH to be transferred over RF, thanks to the XCHG_DATA_REQ command.

The key used is the one used for the sector currently authenticated. In a symmetrical way, PN7150 will automatically decrypt the data received from RF before it forwards to the DH thanks to the *XCHG_DATA_RSP* response, again using the key of the sector currently authenticated.

Fig 32 illustrates the use of the *MFC_Authenticate_REQ* & *XCHG_DATA_REQ* in a typical NFC reader sequence for MIFARE Classic.

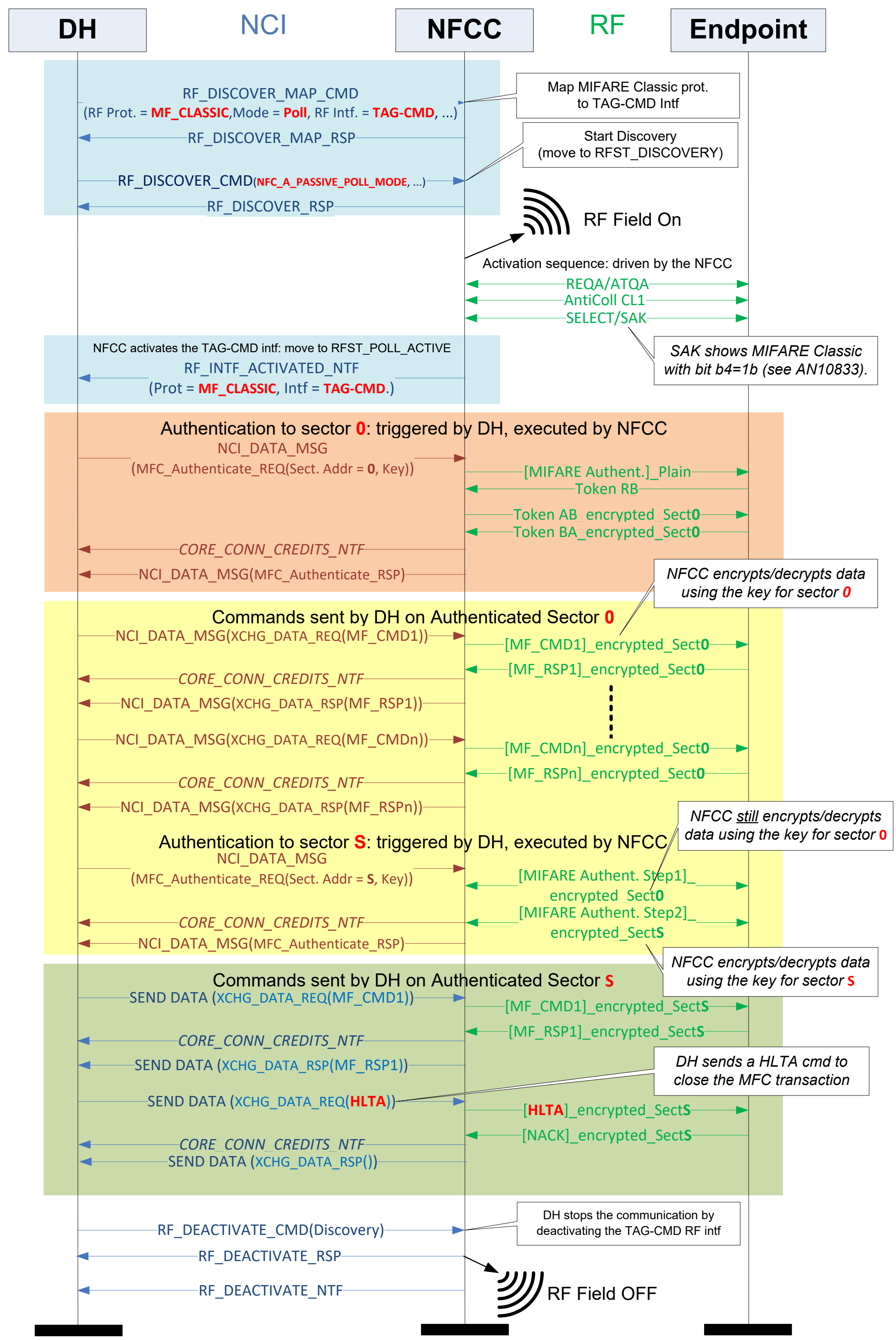

Fig 32. Reader Sequence for MIFARE Classic

6.1.9 Access through the TAG-CMD RF Interface

The TAG-CMD RF interface allows full access to all the Tags based on NFC-A technology and not supporting the ISO-DEP protocol, leaving up to the PN7150 to manage the low level TAG-CMD:

Table 41. Tag/Cards accessible over the TAG-CMD Interface

Tag/Card	Access through the TAG-CMD Interface
T1T	✓
T2T	✓
MIFARE Ultralight, Ultralight C	✓
MIFARE Classic	✓
MIFARE Plus for Security levels 1 & 2	✓

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for T1T, T2T, and MIFARE Classic through the TAG-CMD Interface:

Table 42. Config. seq. for R/W of T1T, T2T & MFC through the TAG-CMD Interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol (choose between the 3 possible protocols)	PROTOCOL_T1T
		PROTOCOL_T2T
		PROTOCOL_MIFARE_CLASSIC
	Mode	Poll
	RF Interface	TAG-CMD
CORE_SET_CONFIG_CMD	PA_BAIL_OUT [1]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_POLL_MODE

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

6.2 T3T tag

[NCI] allows the data exchange with a tag T3T by using the Frame RF Interface, so there is no need to add proprietary extensions here.

6.2.1 Access through the Frame RF Interface

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for T3T Tags/Cards through the Frame RF Interface:

Table 43. Config. seq. for R/W of T3T through the Frame RF Interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_T3T
	Mode	Poll
	RF Interface	Frame
CORE_SET_CONFIG_CMD	PF_BIT_RATE	

Command	Main Parameters	Values
	PF_RC_CODE	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_F_PASSIVE_POLL_MODE

6.3 T4T & ISO-DEP Tags/Cards

[NCI] allows the data exchange with a T4T tag or an ISO-DEP tag by using the Frame RF Interface or the ISO-DEP RF Interface, so there is no need to define a proprietary RF interface here.

6.3.1 Access through the Frame RF Interface

The Frame RF interface allows full access to all the Tags based on NFC-A & NFC-B technology and supporting the ISO-DEP protocol, assuming that the ISO-DEP protocol is fully handled by the DH:

Table 44. Tag/Cards accessible over the Frame RF Interface

Tag/Card	Access through the Frame RF Interface
T4T	✓
MIFARE DESFire	✓
MIFARE Plus for Security levels 3	✓
JCOP-based smart cards	✓

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for ISO-DEP Tags/Cards through the Frame RF Interface for technology NFC-A:

Table 45. Config. seq. for R/W of NFC-A / ISO-DEP through the Frame RF interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD *	RF Protocol	PROTOCOL_ISO-DEP
	Mode	Poll
	RF Interface	Frame
CORE_SET_CONFIG_CMD	PA_BAIL_OUT [1]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_POLL_MODE

* Note: RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for ISO-DEP Tags/Cards through the Frame RF Interface for technology NFC-B:

Table 46. Config. seq. for R/W of NFC-B / ISO-DEP through the Frame RF interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD *	RF Protocol	PROTOCOL_ISO-DEP

Command	Main Parameters	Values
	Mode	Poll
	RF Interface	Frame
CORE_SET_CONFIG_CMD	PB_AFI	
	PB_BAIL_OUT [1]	
	PB_SENSB_REQ_PARAM [2]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_B_PASSIVE_POLL_MODE

* Note: RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

[2] this parameter is not supported in PN7150: STATUS_INVALID_PARAM will be returned to the DH if it attempts to write this parameter.

6.3.2 Access through the ISO-DEP RF Interface

The ISO-DEP RF interface allows full access to all the Tags based on NFC-A & NFC-B technology and supporting the ISO-DEP protocol, leaving up to the PN7150 to manage the ISO-DEP protocol:

Table 47. Tag/Cards accessible over the ISO-DEP RF Interface

Tag/Card	Access through the ISO-DEP RF Interface
T4T	✓
MIFARE DESFire	✓
MIFARE Plus for Security levels 3	✓
JCOP-based smart cards	✓

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for ISO-DEP through the ISO-DEP Interface for technology NFC-A:

Table 48. Config. seq. for R/W of NFC-A / ISO-DEP through the ISO-DEP interface

Command	Main Parameters	Values
	RF Protocol	PROTOCOL_ISO-DEP
RF_DISCOVER_MAP_CMD	Mode	Poll
	RF Interface	ISO-DEP
	PA_BAIL_OUT [1]	
CORE_SET_CONFIG_CMD	PI_BIT_RATE	
	PA_ADV_FEAT [3]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_POLL_MODE

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

[2] this parameter is not supported in PN7150: STATUS_INVALID_PARAM will be returned to the DH if it attempts to write this parameter.

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for ISO-DEP through the ISO-DEP Interface for technology NFC-B:

Table 49. Config. seq. for R/W of NFC-B / ISO-DEP through the ISO-DEP interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_ISO-DEP
	Mode	Poll
	RF Interface	ISO-DEP
CORE_SET_CONFIG_CMD	PB_AFI	
	PB_BAIL_OUT [1]	
	PB_H_INFO	
	PI_BIT_RATE	
	PB_SENSB_REQ_PARAM [3]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_B_PASSIVE_POLL_MODE

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

[2] this parameter is not supported in PN7150: STATUS_INVALID_PARAM will be returned to the DH if it attempts to write this parameter.

6.3.3 [PN7150-NCI] extension: Presence check Command/Response

When a Tag/Card has been activated in Poll Mode, the RF State Machine is then in state RFST_POLL_ACTIVE. It is useful for the DH to know if the card is still in the field or not, especially at the end of the transaction. For that purpose, NXP has added a proprietary command to check the Tag/Card presence.

All the rules defined for command/response in [NCI] (section 3.2) apply to the command defined here. Here are two additional rules:

- ⇨ The DH can use this command ONLY if the RF State Machine is in state RFST_POLL_ACTIVE. PN7150 will respond “STATUS_SEMANTIC_ERROR” in case this command is sent in any other state
- ⇨ The DH can use this command ONLY if the active protocol is either ISO-DEP or NFC-DEP

Table 50. RF_PRES-CHECK_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x11	0	The DH asks to know if the ISO-DEP Tag/Card is in the field or not.

Table 51. RF_PRES-CHECK_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x11	1	The NFCC acknowledges the command received from the DH.

Table 52. RF_PRES-CHECK_RSP parameters

Payload Field(s)	Length	Value/Description	
STATUS	1 Octet	One of the following Status codes, as defined in [NCI_Table1]	
		0x00	STATUS_OK
		0x01	STATUS_REJECTED
		0x06	STATUS_SEMANTIC_ERROR
		Others	Forbidden

Table 53. RF_PRES-CHECK_NTF

GID	OID	Numbers of parameter(s)	Description
1111b	0x11	1	NFCC indicates if the ISO-DEP Tag/Card is still in the field or not.

Table 54. RF_PRES-CHECK_NTF parameters

Payload Field(s)	Length	Value/Description	
Presence	1 Octet	0x00	Card no more in the field
		0x01	Card still in the field
		0x02-0xFF	RFU

6.3.4 [PN7150-NCI] extension: S-Block Command/Response

In some circumstances the DH may want to send specific S-Block to the remote card.

All the rules defined for command/response in [NCI] (section 2.2) apply to the commands defined here. Here are two additional rules:

- ⇨ The DH SHALL not issue these commands if the ISO-DEP RF Interface is not activated.
- ⇨ If the DH issues such a command although the ISO-DEP RF Interface is not activated, the NFCC SHALL send the corresponding response with STATUS set to STATUS_SEMANTIC_ERROR.

Table 55. RF_T4T_SBLOCK_PARAM_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x10	1	Command to allow the DH to send S-Block S(PARAMETERS) over RF.

Table 56. RF_T4T_SBLOCK_PARAM_CMD parameters

Payload Field(s)	Length	Value/Description
ABI	N* Octets	S-Block S(PARAMETERS) to send; the payload only has to be provided (i.e. PARAMETERS), NFCC will encapsulate it in an S-Block.

* PN7150 supports maximum 10 Bytes for ABI length

Table 57. RF_T4T_SBLOCK_PARAM_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x10	1	The NFCC acknowledges the command received from the DH.

Table 58. RF_T4T_SBLOCK_PARAM_RSP parameters

Payload Field(s)	Length	Value/Description	
STATUS	1 Octet	0x00	STATUS_OK
		0x01	STATUS_REJECTED
		0x06	STATUS_SEMANTIC_ERROR
		Others	Forbidden

Table 59. RF_T4T_SBLOCK_PARAM_NTF

GID	OID	Numbers of parameter(s)	Description
1111b	0x10	2	The NFCC sends the response S-Blocks S(PARAMETERS) to the DH.

Table 60. RF_T4T_SBLOCK_PARAM_NTF parameters

Payload Field(s)	Length	Value/Description	
ABT	N[1] Octets	Response received on RF to the S-Block sent. If there is no error on RF, the payload only is provided (i.e. PARAMETERS), NFCC will extract it from the received S-Block. If there is an RF error, this field is empty.	
STATUS		0x00	STATUS_OK
		0x02	STATUS_RF_FRAME_CORRUPTED
		0xB0	RF_TRANSMISSION_ERROR
		0xB1	RF_PROTOCOL_ERROR
		0xB2	RF_TIMEOUT_ERROR
		Others	Forbidden

[1] PN7150 supports maximum 10 Bytes for ABT length

6.3.5 [PN7150-NCI] extension: WTX notification

After data was sent to the card/tag, it can request an additional processing time before sending data response. This is done with WTX (Waiting Time Extension) request. If WTX REQ/RESP exchange phase continues a NCI system notification WTX is sent with a period configurable via READER_FWITOX_NTF_CFG.

Table 61. PH_NCI_OID_SYSTEM_WTX

GID	OID	Numbers of parameter(s)	Description
1111b	0x17	0	Notification indicating that RF communication is in phase of WTX(RTOX) REQ/RESP exchange for longer period of time.

6.3.6 [PN7150-NCI] extension: Higher bit rates in Poll NFC-A & NFC-B

[NCI] does not “officially” support the use of higher bit rates in technology NFC-A & NFC-B.

PN7150 offers 4 different bit rates for these technologies, which can be used either in Poll Mode (to read/write an external Card/Tag) or in Listen Mode (to emulate a card):

1. 106 kbps (default bit rate, always used during activation)
2. 212 kbps
3. 424 kbps
4. 848 kbps

Everything is prepared (see the RF configuration parameter PI_BIT_RATE), except for the ISO-DEP RF Interface activation.

As currently defined in [NCI], the ISO-DEP RF interface activation for technology NFC-A is incompatible with bit rates higher than 106kbps, since this requires to handle the PPS commands exchange, which is not addressed in [NCI].

So the PN7150 implements an ISO-DEP RF Interface activation which is different from the one described in [NCI_Chap1] (see chapter →15). Here is a copy of this chapter, where the modification as implemented in the PN7150 is highlighted in *red italic*:

Copied from [NCI]

8.3.2.2 Discovery and Interface Activation

To enable Poll Mode for ISO-DEP, the DH sends the RF_DISCOVER_CMD to the PN7150 containing configurations with RF Technology and Mode values of NFC_A_PASSIVE_POLL_MODE and/or NFC_B_PASSIVE_POLL_MODE.

When the PN7150 is ready to exchange data (that is, after receiving a response to the protocol activation command from the Remote NFC Endpoint), it sends the RF_INTF_ACTIVATED_NTF to the DH to indicate that this Interface has been activated to be used with the specified Remote NFC Endpoint.

Detailed ISO-DEP RF Interface activation handling in the NFCC:

For NFC-A:
Following the anticollision sequence, if the Remote NFC Endpoint supports ISO-DEP Protocol, the NFCC sends the RATS Command to the Remote NFC Endpoint. And after receiving the RATS response, *the PN7150 MAY send the PPS command if PI_BIT_RATE was set by the DH to an allowed value higher than 0x00.* It SHALL then send the RF_INTF_ACTIVATED_NTF to the DH to indicate a Remote NFC Endpoint based on ISO-DEP has been activated. The RF_INTF_ACTIVATED_NTF will inform the DH on the actual bit rate used on RF.

For NFC-A the RF_INTF_ACTIVATED_NTF SHALL include the Activation Parameters defined in Table 74 (see below).

Table 74: Activation Parameters for NFC-A/ISO-DEP Poll Mode

Parameter	Length	Description
RATS Response Length	1 Octet	Length of RATS Response Parameter (n)
RATS Response	n Octets	All Bytes of the RATS Response as defined in [DIGITAL] starting from and including Byte 2.

End of Copy from [NCI]

6.4 [PN7150-NCI] extension: 15693 & I-Code tags

The current version of the NCI standard allows the data exchange with a tag ISO15693 by using the RF Frame interface. No additional interface is needed for this protocol. However, the data mapping is not yet defined in [NCI], therefore, NXP has defined it for [PN7150-NCI].

6.4.1 Access through the Frame RF Interface

The Frame RF interface allows full access to all the Tags based on NFC-15693 technology. Here is a list of such tags from the NXP portfolio:

Table 62. NFC-15693 compliant Tag/Cards accessible over the Frame RF Interface

Tag/Card	Access through the Frame RF Interface
I-Code SLI	✓
I-Code SLI-L	✓
I-Code SLI-S	✓

Here are the commands and configuration parameters to prepare the Reader/Writer Mode for NFC-15693 Tags/Cards through the Frame RF Interface:

Table 63. Config. seq. for R/W of NFC-15693 through the Frame RF Interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD *	RF Protocol	PROTOCOL_15693
	Mode	Poll
	RF Interface	Frame RF
RF_DISCOVER_CMD	RF Technology & Mode	NFC_15693_PASSIVE_POLL_MODE

* Note: RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

6.4.2 [PN7150-NCI] extension: Specific parameters for NFC_15693 Poll Mode

Once PN7150 detects and activates a remote NFC Endpoint based on NFC-15693, PN7150 will activate the Frame RF Interface, providing the following activation parameters:

Table 64. Specific parameters for NFC_15693 Poll Mode

Parameter	Length	Description
FLAGS	1 Octet	1st Byte of the Inventory Response
DSFID	1 Octet	2nd Byte of the Inventory Response
UID	8 Octets	3rd Byte to last Byte of the Inventory Response

6.4.3 [PN7150-NCI] extension: Data Mapping between the DH and RF

Data from the DH to RF

The NCI Data Message corresponds to the Request Format defined in [ISO15693-3] Section 7.3.

After receiving a Data Message from the DH, the PN7150 appends the appropriate EoD, SOF and EOF and then sends the result in an RF Frame in NFC-15693 technology to the Remote NFC Endpoint.

The following figure illustrates the mapping between the NCI Data Message Format and the RF frame when sending the RF frame to the Remote NFC Endpoint. This figure shows the case where NCI Segmentation and Reassembly feature is not used.

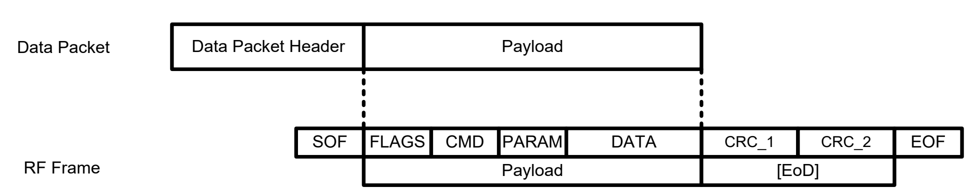

Fig 33. Format for Frame RF Interface (NFC-15693) for Transmission

Although the Frame RF interface is defined to be a transparent interface where the NFCC does not parse/modify the Bytes transmitted by the DH, the following exceptions occur:

! PN7150 is parsing the bit Option Flag (bit b7 in the request Flags Byte, as defined in ISO15693) to check if this bit is set by the DH or not. If set, this indicates that the tag is from TI, and PN7150 is sending commands over RF using a special mode, as defined for some commands in ISO15693.

Data from RF to the DH

The NCI Data Message corresponds to the Payload of the Response Format defined in [ISO15693-3] Section 7.4, followed by a Status field of 1 octet.

After receiving an RF frame, the PN7150 checks and removes the EoD, the SOF & EOF and sends the result in a Data Message to the DH.

In case of an error the Data Message may consist of only a part of the Payload of the received RF frame but it will always include the trailing Status field. So the PN7150 may send a Data Message consisting of only the Status field if the whole RF frame is corrupted.

If the RF frame was received correctly, the PN7150 sets the Status field of Data Message to a value of STATUS_OK. If the PN7150 detected an error when receiving the RF frame, it sets the Status field of the Data Message to a value of STATUS_RF_FRAME_CORRUPTED.

The following figure illustrates the mapping of the RF frame received from the Remote NFC Endpoint in technology NFC-15693 to the Data Message format to be sent to the DH. This figure shows the case where NCI Segmentation and Reassembly feature is not used.

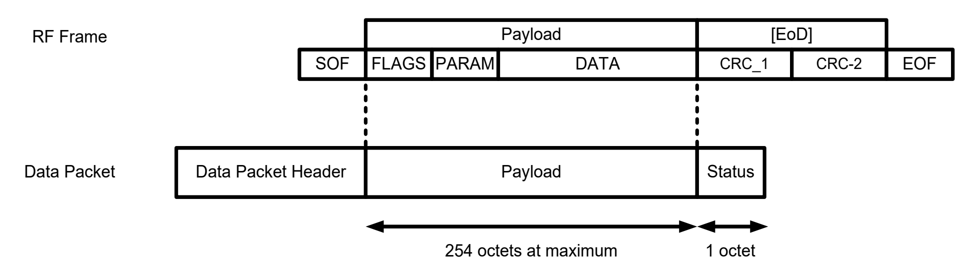

Fig 34. Format for Frame RF Interface (NFC-15693) for Reception

6.4.4 PN7150 behavior with multiple VICCs

PN7150 supports collision resolution (using the Inventory command), so it can detect multiple VICCs (2 maximum, as defined for CON_DEVICE_LIMIT in →4.2.5).

Here is the behavior when two VICCs are detected and then, one of them is removed from the Field before the DH wants to select it:

- PN7150 is in state RFST_DISCOVERY; it detects 2 VICCs. It sends an RF_DISCOVER_NTF to the DH for VICC1 and moves to RFST_W4_ALL_DISCOVERIES.

- PN7150 is in state RFST_W4_ALL_DISCOVERIES, it sends an RF_DISCOVER_NTF to the DH for VICC2 and moves to RFST_W4_HOST_SELECT.
- PN7150 is in state RFST_W4_ALL_DISCOVERIES and waits for the DH to select one of the 2 VICCs. Once it receives the RF_DISCOVER_SELECT_CMD from the DH, PN7150 immediately activates the Frame RF Interface and does not check if the selected VICC is still in the field. That means that PN7150 will not send a CORE_GENERIC_ERROR_NTF (Discovery_Target_Activation_Failed) to the DH if the selected VICC is not in the field anymore. The state is now changed to RFST_POLL_ACTIVE.
- PN7150 is in state RFST_POLL_ACTIVE; it waits for the DH to send some data to transfer over RF. Once it gets this data, PN7150 forwards it over RF. If the selected VICC is not in the field anymore, PN7150 will stay mute and will not send any data back to the DH. The DH has to implement a time-out function, to detect that the VICC is not in the field anymore. Once this timeout is triggered, the DH can de-activate the Frame RF Interface by sending the RF_DEACTIVATE_CMD.

6.5 [PN7150-NCI] extension: KOVIO tags

Kovio tags are very particular tags which use a sub-set of NFC-A technology.

The basic concept is that the tag is powered from RF Field generated by PN7150, and it will spontaneously generate a 16-Byte ID using NFC-A load modulation, although it did not receive any command from PN7150. Once PN7150 has detected a Kovio tag by capturing its ID, PN7150 will send a RF_INTF_ACTIVATED_NTF, transporting the tag ID as RF parameter.

Table 65. Kovio specific RF parameters inside the RF_INTF_ACTIVATED_NF

Payload Field(s)	Length	Value/Description
…		
Length of RF Technology Specific Parameters	1 Octet	16
RF Technology Specific Parameters	16 Octets	Kovio ID
…		

It is then up to the DH to decide when to leave the RFST_POLLING_ACTIVE state, and also to decide if it directly comes back to RFST_DISCOVERY, where the same Kovio Tag may be discovered again, or if it comes back to RFST_IDLE first, in order to wait without any RF activity or re-configuring the RF Discovery so that PN7150 does not poll for a Kovio tag again.

Kovio tags are accessed through the [NCI] Frame RF Interface.

Due to the very particular behavior of the Kovio tags, it is necessary to configure the RF Discovery specifically for these tags, using the NFC-A_KOVIO_POLL_MODE parameter for the RF_DISCOVER_CMD as highlighted in the table below:

Table 66. Config. seq. for R/W of Kovio tags through the Frame RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD*	RF Protocol	PROTOCOL_KOVIO
	Mode	Poll
	RF Interface	Frame RF Interface
CORE_SET_CONFIG_CMD	PA_BAIL_OUT[1]	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_KOVIO_POLL_MODE

* Note: RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with Bail Out in NFC-A, whatever the value written by the DH to that parameter.

7. Card Emulation Mode

The PN7150 supports Card Emulation hosted by the DH based on either technology NFC-A, NFC-B or NFC-F.

7.1 ISO-DEP card emulation through NFC-A & NFC-B

[NCI] defines all the mechanisms necessary to implement this feature. Two options are possible:

1. The DH wants to manage by itself the ISO-DEP protocol; it SHALL then map the ISO-DEP protocol on the Frame RF Interface.

   ! Not supported in PN7150

2. The DH leaves the ISO-DEP protocol management to the NFCC: it SHALL then map the ISO-DEP protocol on the ISO-DEP interface.

Here are the commands and configuration parameters to prepare the ISO-DEP Card Emulation for technology NFC-A in the DH through the ISO-DEP RF Interface:

Table 67. Config. seq. for CE of ISO-DEP/NFC-A

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_ISO-DEP
	Mode	Listen
	RF Interface	ISO-DEP
CORE_SET_CONFIG_CMD	LA_BIT_FRAME_SDD	
	LA_PLATFORM_CONFIG	
	LA_SEL_INFO	
	LA_NFCID1	
	LI_FWI	
	LA_HIST_BY	
	LI_BIT_RATE	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_LISTEN_MODE

Here are the commands and configuration parameters to prepare the ISO-DEP Card Emulation for technology NFC-B in the DH through the Frame RF Interface:

Table 68. Config. seq. for CE of ISO-DEP/NFC-B

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_ISO-DEP
	Mode	Listen
	RF Interface	ISO-DEP
CORE_SET_CONFIG_CMD	LB_SENSB_INFO	
	LB_NFCID0	

Command	Main Parameters	Values
	LB_APPLICATION_DATA	
	LB_SFGI	
	LB_ADC_FO	
	LI_FWI	
	LB_H_INFO_RESP [1]	
	LI_BIT_RATE	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_B_PASSIVE_LISTEN_MODE

[1] this parameter is not active in PN7150: it can be read/written, but PN7150 will always behave with empty Higher Layer – Response field in the ATTRIB response, whatever the value written by the DH to that parameter.

7.2 T3T card emulation through NFC-F

7.2.1 Configuring the T3T card emulation

As described in the NFC specification, several Listen F parameters exist to set up T3T with NCI commands.

Table 69. Values to configure the T3T on DH

ID	Length	Values and description
LF_T3T_MAX	1 byte	0 – 16, defines the maximum amount of LF_T3T_IDENTIFIERS supported by the NFCC. PN7150 supports four maximum.
LF_T3T_IDENTIFIERS_1 - 4	10 bytes	Bytes 0 and 1 define the SC to be used by the T3T. Bytes 2 – 10 define the NFCID2 value to be used.

7.2.2 Access through the Frame RF Interface

The Frame RF interface allows emulating a T3T card, assuming that the DH is able to manage the T3T protocol on its own.

Here are the commands and configuration parameters to prepare the T3T Card Emulation for technology NFC-F through the Frame RF Interface:

Table 70. Configuration seq. for ISO-DEP/NFC-A Card Emulation in the DH over Frame RF Interface

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD *	RF Protocol	PROTOCOL_T3T
	Mode	Listen
	RF Interface	Frame
CORE_SET_CONFIG_CMD	LF_T3T_MAX	See above, used to set SC, NFCID2
	LF_T3T_IDENTIFIERS_X	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_F_PASSIVE_LISTEN_MODE

* Note : RF_DISCOVER_MAP_CMD is optional since the mapping to Frame RF Intf. is done by default

8. P2P Initiator & Target Mode

8.1 P2P Passive mode

[NCI] defines all the mechanisms necessary to implement this feature. Two options are possible:

1. The DH wants to manage by itself the NFC-DEP protocol; it SHALL then map the NFC-DEP protocol on the Frame RF Interface.

   ! Not supported in PN7150

2. The DH leaves the NFC-DEP protocol management to the NFCC: it SHALL then map the NFC-DEP protocol on the NFC-DEP interface.

The NFC-DEP RF interface allows the DH to emulate an NFC-DEP Target or Initiator in P2P Passive, leaving up to the PN7150 to manage the NFC-DEP protocol.

Here are the commands and configuration parameters to prepare the NFC-DEP Target in P2P Passive hosted by the DH, for technologies NFC-A and NFC-F, through the NFC-DEP RF Interface:

Table 71. Config. seq. of NFC-DEP/NFC-A&F Passive Target over NFC-DEP RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_NFC-DEP
	Mode	Listen
	RF Interface	NFC-DEP
CORE_SET_CONFIG_CMD	LA_BIT_FRAME_SDD	
	LA_PLATFORM_CONFIG	
	LA_SEL_INFO	
	LA_NFCID1	
	LF_CON_BITR_F	
	LF_PROTOCOL_TYPE	
	LN_WT	
	LF_ADV_FEAT[1]	
	LN_ATR_RES_GEN_BYTES	
	LN_ATR_RES_CONFIG	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_LISTEN_MODE
	RF Technology & Mode	NFC_F_PASSIVE_LISTEN_MODE

[1] this parameter is not supported in PN7150

Here are the commands and configuration parameters to prepare the NFC-DEP Initiator for technologies NFC-A and NFC-F in the DH through the Frame RF Interface:

Table 72. Config. seq. of NFC-DEP/NFC-A&F Passive Initiator over NFC-DEP RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_NFC-DEP
	Mode	Poll
	RF Interface	NFC-DEP
CORE_SET_CONFIG_CMD	PA_BAIL_OUT	
	PF_BIT_RATE	
	PF_RC_CODE	
	PN_NFC_DEP_SPEED	
	PN_ATR_REQ_GEN_BYTES	
	PN_ATR_REQ_CONFIG	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_PASSIVE_POLL_MODE
	RF Technology & Mode	NFC_F_PASSIVE_POLL_MODE

8.2 P2P Active mode

All P2P active modes are supported (Initiator for NFC-A & NFC-F and Target for NFC-A & NFC-F).

As for the P2P Passive mode, the PN7150 allow access to P2P Active mode through the NFC-DEP RF Interface, the Frame RF Interface implemented in PN7150 not supporting the NFC-DEP protocol.

The NFC-DEP RF interface allows the DH to emulate an NFC-DEP Target or Initiator in P2P Active, leaving up to the NFCC to manage the NFC-DEP protocol.

Here are the commands and configuration parameters to prepare the NFC-DEP Target in P2P Active hosted by the DH, for technologies NFC-A and NFC-F, through the NFC-DEP RF Interface:

Table 73. Config. seq. of NFC-DEP/NFC-A&F Active Target over NFC-DEP RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_NFC-DEP
	Mode	Listen
	RF Interface	NFC-DEP
CORE_SET_CONFIG_CMD	LA_BIT_FRAME_SDD	
	LA_PLATFORM_CONFIG	
	LA_SEL_INFO	
	LA_NFCID1	
	LF_CON_BITR_F	
	LF_PROTOCOL_TYPE	
	LN_WT	
	LN_ATR_RES_GEN_BYTES	
	LN_ATR_RES_CONFIG	

Command	Main Parameters	Values
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_ACTIVE_LISTEN_MODE
	RF Technology & Mode	NFC_F_ACTIVE_LISTEN_MODE

Here are the commands and configuration parameters to prepare the NFC-DEP Initiator for technologies NFC-A and NFC-F in the DH through the Frame RF Interface:

Table 74. Config. seq. of NFC-DEP/NFC-A&F Active Initiator over NFC-DEP RF Intf

Command	Main Parameters	Values
RF_DISCOVER_MAP_CMD	RF Protocol	PROTOCOL_NFC-DEP
	Mode	Poll
	RF Interface	NFC-DEP
CORE_SET_CONFIG_CMD	PA_BAIL_OUT	
	PF_BIT_RATE	
	PN_NFC_DEP_SPEED	
	PN_ATR_REQ_GEN_BYTES	
	PN_ATR_REQ_CONFIG	
RF_DISCOVER_CMD	RF Technology & Mode	NFC_A_ACTIVE_POLL_MODE
	RF Technology & Mode	NFC_F_ACTIVE_POLL_MODE

8.3 Presence check command

As already described in →6.3.3, the PN7150 comes with a proprietary function to allow the DH knowing if the Tag/Card is still present or not. The command description in →6.3.3 also applies in Initiator mode (Active or Passive).

8.4 WTX notification

As already described in →6.3.5, the PN7150 comes with a proprietary notification WTX which indicates that peers are in phase of exchanging RTOX REQ/RESP (NFC DEP equivalent of WTX in ISO DEP) for the configured period of time. The notification description in →6.3.5 also applies in Initiator mode (Active or Passive).

9. RF Discovery Management

9.1 RF Discovery functionalities

This contains the overall RF Discovery concepts applied in PN7150. [NCI] defines the general RF state machine allowing the NFC controller to discover either cards or readers or peers. This RF state machine contains a state called RFST_DISCOVERY where the RF Discovery profile is applied.

In order to ensure standard compliance, the PN7150 supports 2 different RF discovery profiles:

- NFC FORUM profile: implementation of the NFC FORUM polling activity,
  - Either limited to the current technologies defined in this standardization body (NFC-A, NFC-B, NFC-F and P2P passive).
  - Or extended with the additional technologies supported by PN7150, i.e. P2P Active and ISO15693. PN7150 also offers the possibility to extend this profile by polling for both NFC-F 424 and NFC-F 212.
- EMVCo profile: mode allowing the PN7150 to be compliant to the EMVCo polling activity.

In addition to these RF profiles, the PN7150 offers a way to limit the power consumption by applying a tag detector concept. The tag detector can be seen as a precondition to enable a dedicated profile. It means that if the tag detector is triggered, the default profile is automatically started.

Note that [NCI] defines the TOTAL_DURATION of the discovery period independently of the reader phases applied. To simplify the implementation, for the PN7150 it has been decided to apply a timer only during the Listen/pause phase. So depending on the polling phase configuration (1 technology or more), the total duration will vary a bit. This is considered as acceptable and agreed by the NCI task Force in the NFC FORUM.

The following drawing shows the [PN7150-NCI] RF state machine. It differs from [NCI] only by the additions in red.

Here are these additions:

✓ A loop-back transition on state *RFST_POLL_ACTIVE*, corresponding to the *RF_PRES_CHECK_CMD* which can be sent by the DH to know if the Card/PICC is still in the field. See the command description in chapter →6.3.3.

✓ A new status code used on the *CORE_GENERIC_ERROR_NTF* loop-back transition on state *RFST_DISCOVERY*: this new status code is used when PN7150 is configured to behave as an EMVCo PCD, and it detects collision. See →9.5.1.2 for more details.

✓ A new transition from *RFST_POLL_ACTIVE* to *RFST_DISCOVERY*: this transition is triggered by PN7150, when it is configured to behave as an EMVCo PCD and it detects that the RF communication with the PICC is broken. See →9.5.1.2

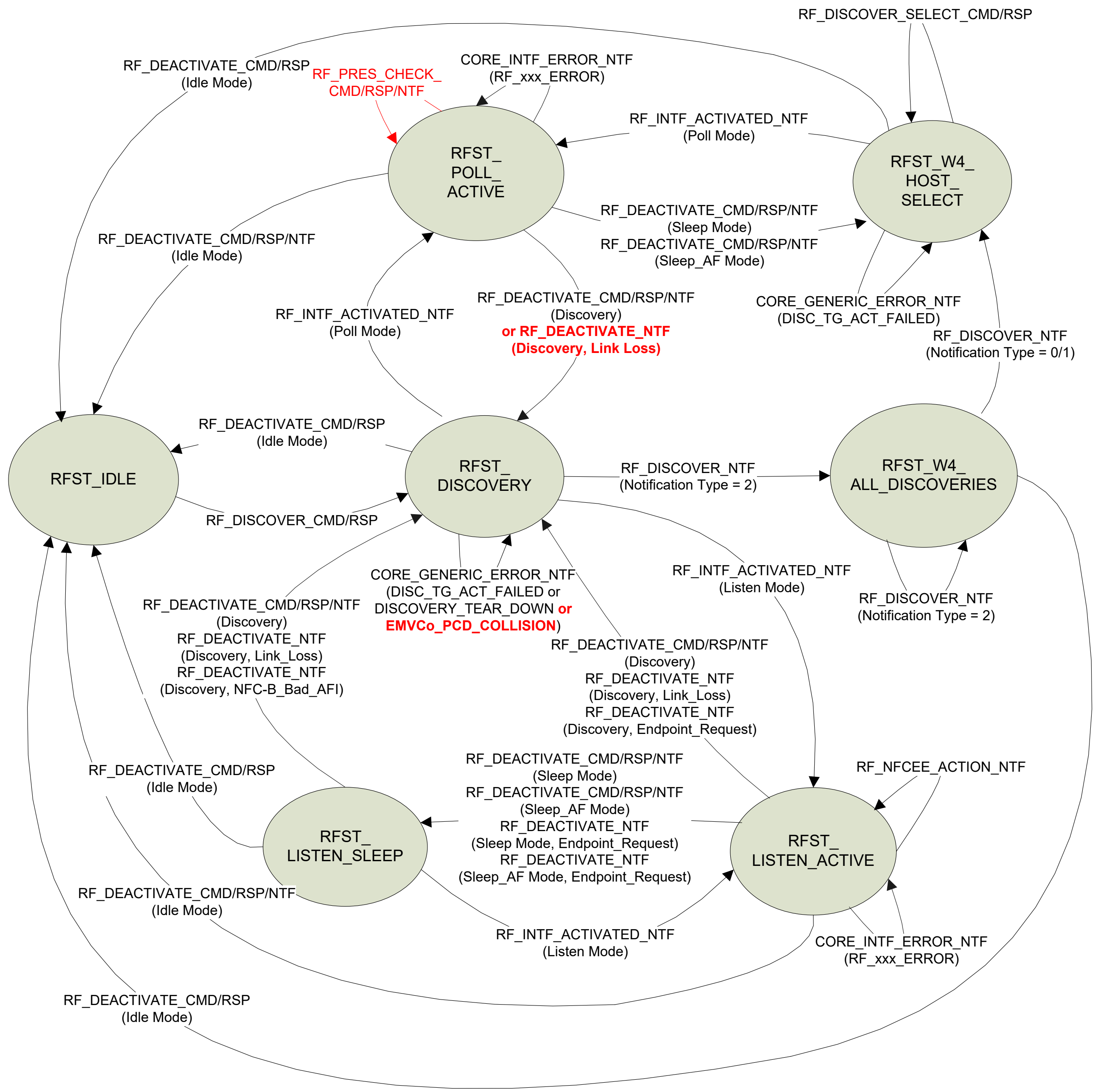

Fig 35. NXP RF State machine

Since the [NCI] RF State Machine is quite complex, it is presented slightly differently in Annex A of the present document: the State Machine is drawn depending on the RF interface to be used. See chapter →13 for further details.

> ! Since PN7150 does not support Listen Mode using the Frame RF Interface, it does not accept the RF_DEACTIVATE_CMD(Sleep Mode) or

RF_DEACTIVATE_CMD(Discovery) in RFST_LISTEN_ACTIVE or RFST_LISTEN_SLEEP.

9.2 NFC FORUM Profile as defined in [NCI]

The NFC FORUM profile is the implementation of the RF discovery activity as defined in the NFC FORUM (see [ACTIVITY] specification). [NCI] only covers technologies NFC-A, NFC-B & NFC-F. So the basic NFC FORUM profile will poll for these technologies only. Furthermore, for NFC-F, only one bit rate is used during the polling phase. This is configured thanks to the "Poll F parameter" *PF_BIT_RATE* as defined in [NCI], section →6.1.4. So the DH configures if NFC-F is polled at 212kbps or at 424kbps, before it activates the discovery by sending the *RF_DISCOVER_CMD* command.

The figure bellow represents the profile defined by the NFC FORUM, assuming that the DH has enabled the 3 technologies currently supported by the NFC FORUM (NFC-A, NFC-B, NFC-F) in Poll mode & Listen mode. To do so, it has to send the following command:

```
RF_DISCOVER_CMD(    6,
                    [NFC_A_PASSIVE_POLL_MODE,1],
                    [NFC_B_PASSIVE_POLL_MODE,1],
                    [NFC_F_PASSIVE_POLL_MODE,1],
                    [NFC_A_PASSIVE_LISTEN_MODE,1],
                    [NFC_B_PASSIVE_LISTEN_MODE,1],
                    [NFC_F_PASSIVE_LISTEN_MODE,1] )
```

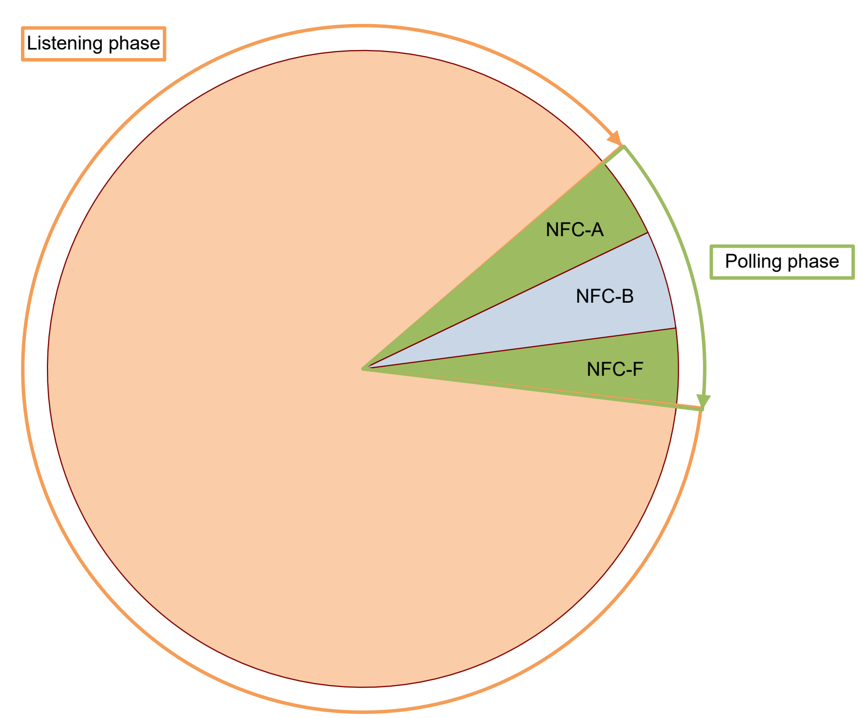

Fig 36. RF Discovery sequence in case of NFC FORUM profile

9.3 [PN7150-NCI] extension: additional technologies not yet supported by the NFC FORUM

PN7150 supports more technologies than currently supported by the NFC FORUM specifications: P2P Active, ISO15693 VCD and KOVIO Reader.

Furthermore, PN7150 offers an additional proprietary value for the configuration parameter PF_BIT_RATE, which allows configuring for both 212 kbps & 424 kbps to be polled in NFC-F in Passive Mode.

Thanks to the *RF_DISCOVER_CMD* and the *PF_BIT_RATE*, the DH has full flexibility to extend the default RF Discovery profile as currently defined in the [NCI] specification. Here is an example how the DH can enable all technologies available in PN7150, for both Poll & Listen Mode:

1. The DH sets PF_BIT_RATE to 0x80, such that the PN7150 polls for 212 & 424 kbps in technology F PASSIVE.

```
CORE_SET_CONFIG_CMD(     NbrParam = 0x01,
                         ID = 0x18,
                         Length = 0x01,
                         Val = 0x80 )
```

2. The DH enables all technologies & modes available in PN7150:

```
RF_DISCOVER_CMD(     11,
                     [NFC_A_PASSIVE_POLL_MODE,1],
                     [NFC_B_PASSIVE_POLL_MODE,1],
                     [NFC_F_PASSIVE_POLL_MODE,1],
                     [NFC_15693_PASSIVE_POLL_MODE,1],
                     [NFC_KOVIO_POLL_MODE,1],
                     [NFC_A_ACTIVE_POLL_MODE*,1],
                     [NFC_A_PASSIVE_LISTEN_MODE,1],
                     [NFC_B_PASSIVE_LISTEN_MODE,1],
                     [NFC_F_PASSIVE_LISTEN_MODE,1],
                     [NFC_A_ACTIVE_LISTEN_MODE,1],
                     [NFC_F_ACTIVE_LISTEN_MODE,1]
                )
```

** NCI_DISCOVERY_TYPE_POLL_F_ACTIVE is not allowed, see →4.2.4.*

The resulting RF discovery is drawn below (note that KOVIO does not have a specific Poll Phase, since it is based on a Response only, as described in →6.5):

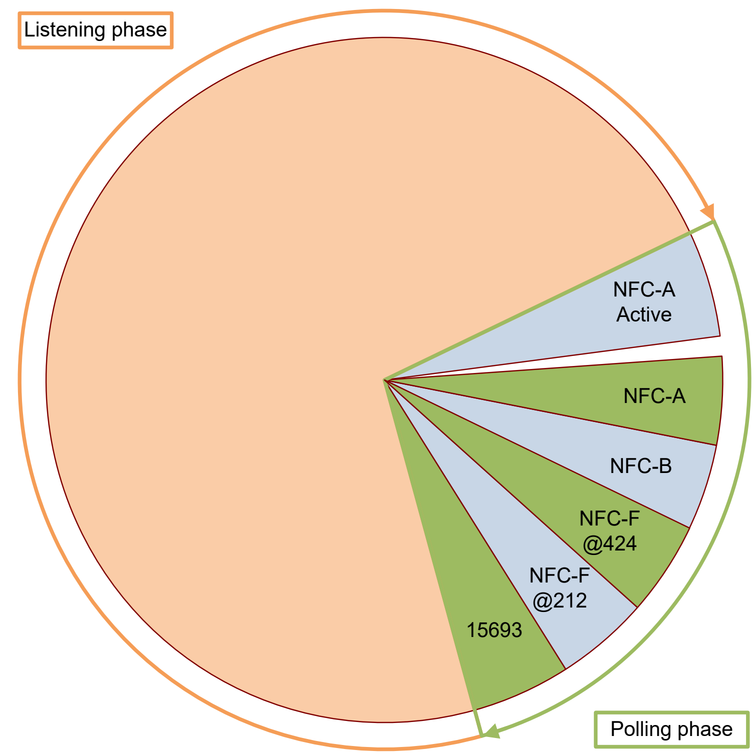

Fig 37. RF Discovery sequence in case of NFC FORUM+ profile

Note: the transition from the Poll NFC-A Active phase to the Poll NFC-A (passive) is done through an RF field off/on sequence.

For more details concerning the different phases duration, guard time, Bailout, please refer to the configuration section (chapter →10.2) where all these parameters are defined.

9.4 [PN7150-NCI] extension: Low Power Card Detector (LPCD) Mode

9.4.1 Description

The Low Power Card Detector is an NXP proprietary extension, which can be used by the DH to reduce the power consumption.

The concept is to avoid using the Technology Detection Activity as defined in [ACTIVITY], which implies to generate an RF Field for several tens of milliseconds and to send technology specific request commands to see if there is a Card/Tag in the field to respond. The more technologies the PN7150 is configured to detect, the longer the RF Field is generated and the higher the current consumption.

The LPCD is based on another concept, which only relies on the antenna characteristics, not on valid responses from a Card/Tag. Indeed, the antenna impedance is influenced by the Card/tag which may enter into its proximity, due to the magnetic coupling between the

2 antennas. The LPCD is therefore monitoring the antenna impedance, to see if there is a significant variation which is interpreted as being caused by a Card/Tag being in proximity.

To achieve that, the LPCD periodically generates very short pulses of RF Field, without any modulation, and measures some antenna characteristics during this pulse. The time between these RF pulses is defined by the *TOTAL_DURATION* parameter, as specified for the RF Discovery in [NCI].

When a Card/Tag enters the field, there is an antenna impedance variation. If this variation is higher than a pre-defined threshold, the NFC FORUM polling loop profile is automatically started (the LPCD is not supported when using EMVCo polling loop profile). The PN7150 is then sending technology specific request commands, expecting a response since the LPCD detected a change on the antenna impedance.

Note: the LPCD may also be triggered by a metal object, which can influence the Antenna impedance in a similar way as a Card/Tag. The PN7150 will anyhow detect that this object is not a contactless device since it immediately starts sending contactless commands to check if a Card/Tag can respond.

The Low Power Card Detector is configured and enabled/disabled thanks to a specific configuration parameter *TAG_DETECTOR_CFG* described in →10.2.1.

The threshold is also defined by an additional configuration parameter *TAG_DETECTOR_THRESHOLD_CFG* described in the same section.

The figure below describes the RF Discovery when the LPCD is enabled:

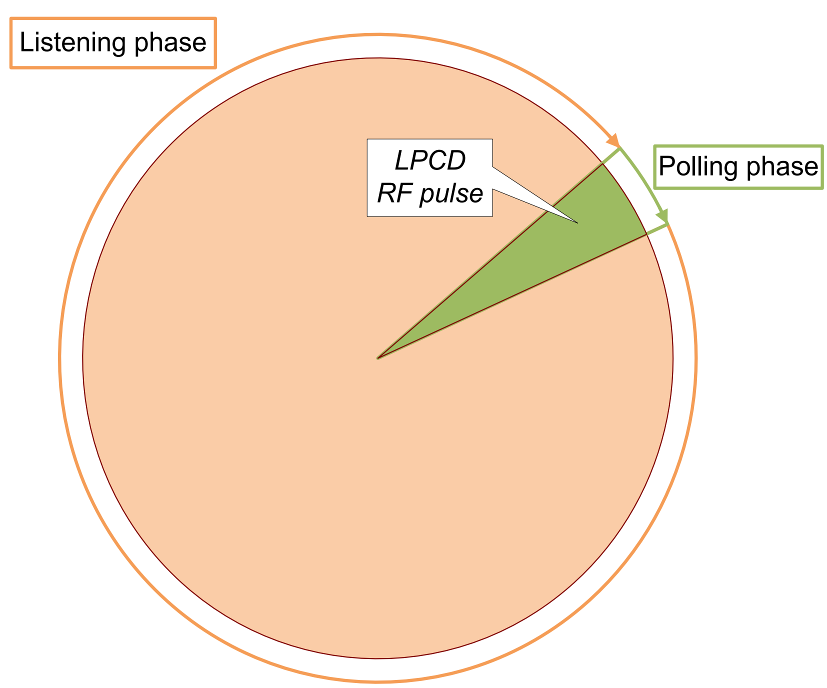

Fig 38. RF Discovery sequence in case of Low Power Card Detector mode

The figure below compares the RF Discovery with the LPCD disabled to the RF Discovery with the LPCD enabled and highlights the impact on the average current consumption (the assumption being here that *TOTAL_DURATION* ~ 300ms):

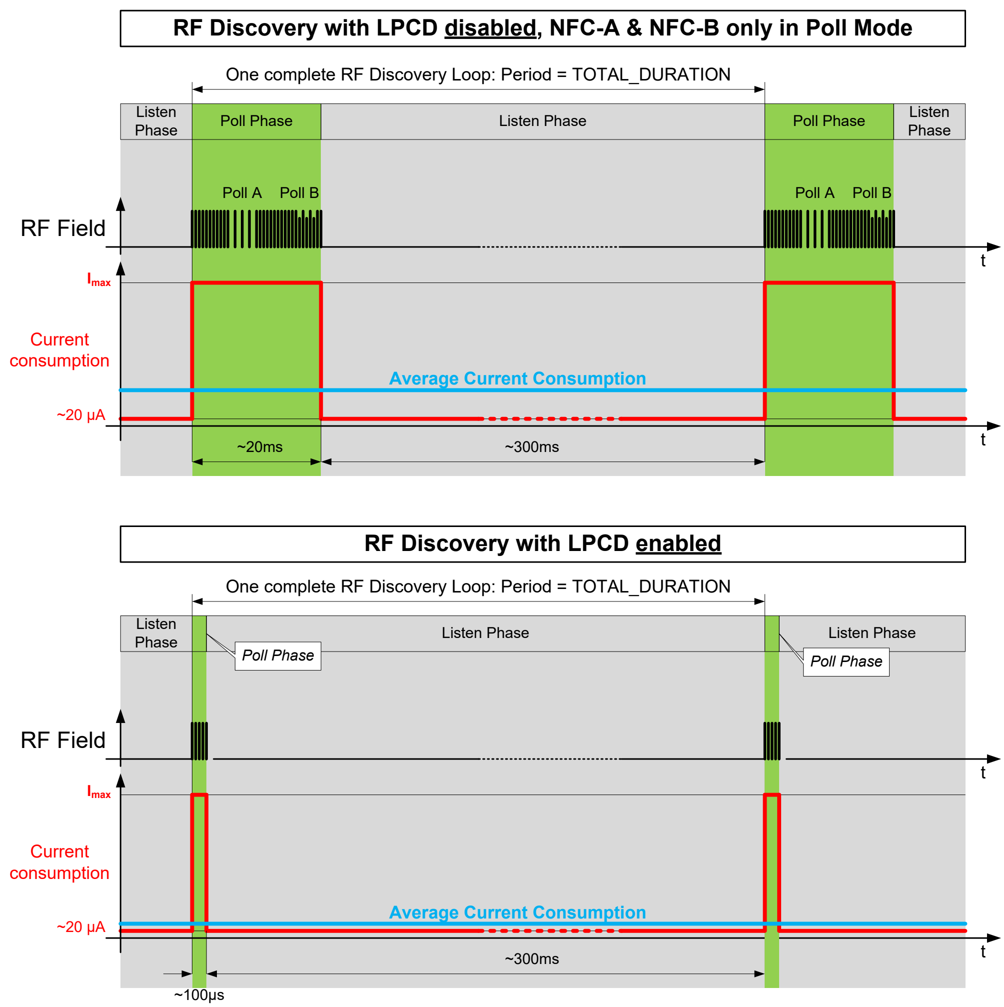

Fig 39. Comparison of the RF Discovery with the LPCD disabled or enabled

A specific application note explains how to properly configure and optimize this LPCD in a given application. See [AN 11757].

9.4.2 Configuration of the Technology Detection Activity when the LPCD has detected an "object"

As described in the previous chapter, once the PN7150 detects a change in the antenna impedance, it performs a Technology Detection as defined in [ACTIVITY] which tries to activate the “object” by sending Request Commands from the different technologies configured for the RF Discovery.

In order to improve the likelihood to catch such a Card/Tag, the PN7150 comes with a retry mechanism which performs several Technology Detection polling cycles before it switches back to LPCD.

During this retry mechanism, a temporary period is used, called *TechDet_PERIOD*. This is specified in steps of 10ms. The number of the retry cycles can also be configured thanks to the *TechDet_NBR_RETRIES* parameter.

Table 75. Parameters used to configure the overall period of the RF Discovery:

LPCD Status	Period between 2 consecutive Technology Detections	Period between 2 consecutive LPCD RF pulses
Enabled	*TechDet_PERIOD*	*TOTAL_DURATION*
Disabled	*TOTAL_DURATION*	Not applicable

The next figure illustrates how these 3 parameters *TOTAL_DURATION*, *TechDet_PERIOD* and *TechDet_NBR_RETRIES* influence the Low Power Card Detector and the RF Discovery:

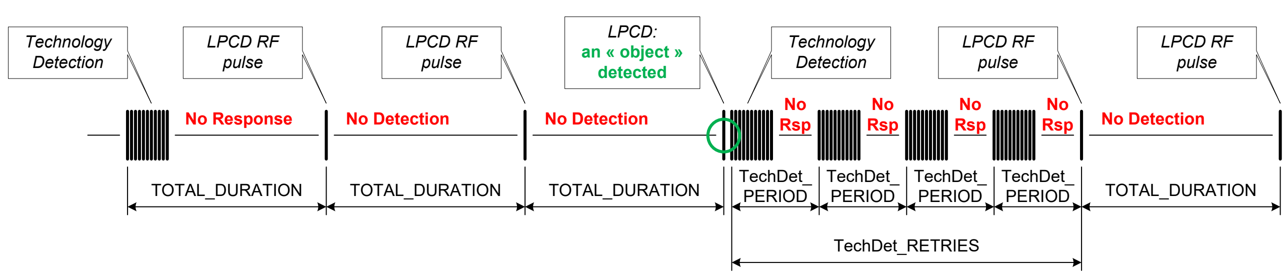

Fig 40. Illustration of the Low Power Card detector and the subsequent Technology Detection cycles

See →10.2.1 for the description of the configuration parameter *TechDet_AFTER_LPCD_CFG* containing the 2 parameters *TechDet_PERIOD* and *TechDet_NBR_RETRIES*.

9.4.3 Notification when the Trace Mode is enabled

The Low Power Card Detector needs to be tuned in each application; it is therefore useful to get some information from PN7150 so that the Low Power Card Detector can be appropriately configured.

The Low Power Card Detector can be configured to enable a Trace Mode, where the following Notification will be sent to the DH by PN7150:

Table 76. RF_LPCD_TRACE_NTF

GID	OID	Numbers of parameter(s)	Description
1111b	0x13	2	PN7150 sends the actual measurement + the threshold

Table 77. RF_LPCD_TRACE_NTF parameters

Payload Field(s)	Length	Value/Description
Reference Value	2 Octets	Reference Value used by Low Power Card Detector function to compare with the measurement value. Coding is little Endian.
Measurement Value	2 Octets	Value measured on the AGC. Coding is little Endian.

9.5 [PN7150-NCI] extension: EMVCo Profile in Poll & Listen Modes

The EMVCo profiles are introduced in PN7150 for EMVCo compliancy. Indeed there are incompatibilities between the RF Discovery activity as defined in the NFC FORUM and the RF discovery defined in EMVCo standard.

9.5.1 EMVCo profile in Poll Mode

9.5.1.1 Configuring PN7150 to implement the EMVCo polling loop profile

To be compliant to the EMVCo certification tests, the RF Discovery has to be configured so that only NFC-A and NFC-B are supported in Poll phase and so that there is no Listen phase. So the DH has to send the following command:

```
RF_DISCOVER_CMD(        2,
                        [NCI_DISCOVERY_TYPE_POLL_A_PASSIVE,1],
                        [NCI_DISCOVERY_TYPE_POLL_B_PASSIVE,1])
```

In addition, PN7150 needs to be aware of the fact that it has to behave according to the EMVCo RF discovery, not according to the NFC FORUM RF discovery based on [ACTIVITY]. A specific configuration parameter *POLL_PROFILE_SEL_CFG* (see 10.2.1) is defined for that purpose, allowing to select the active profile of the RF discovery in Poll Mode. When this parameter is set to 0x01, PN7150 implements a specific discovery algorithm, compliant to the EMVCo standard. The target is to ensure that there is one single card in the field. So PN7150 has to detect any collision inside 1 technology (NFC-A or NFC-B) or to detect if there are multiple cards based on different technologies (i.e. 1 card in NFC-A and 1 card in NFC-B).

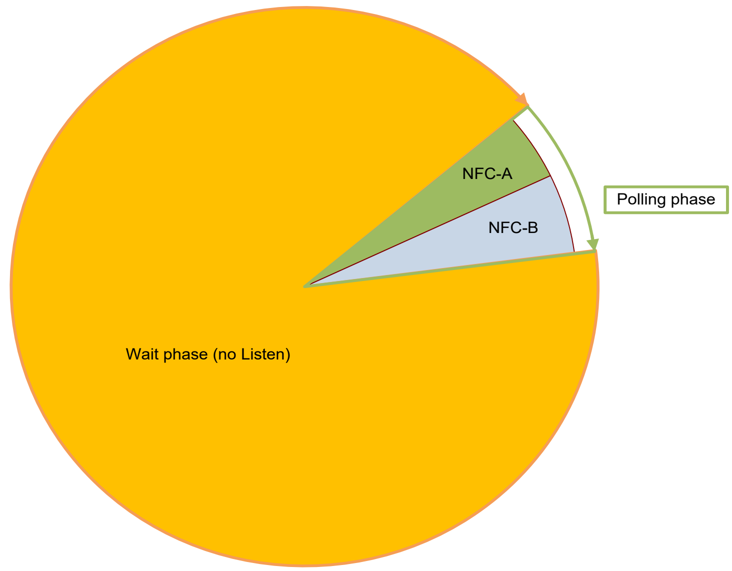

Fig 41. RF Discovery sequence in case of EMVCo profile

If there is a card detected in the field, then the polling sequence is modified by the PN7150, in order to look for another potential card in the field.
This is illustrated by the 2 figures below:

- On the 1st one, there is no card in the RF Field, so PN7150 keeps polling by alternating WUPA & WUPB commands.

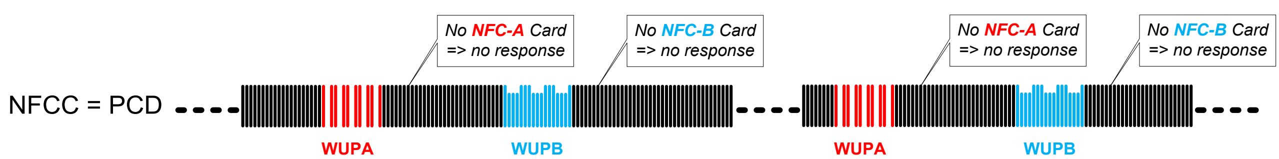

Fig 42. EMVCo polling without a card in the field

- On the 2nd one, an NFC-A card is placed in the RF Field. The PN7150 detects it, activates it and puts it in HALT state and then looks for a potential NFC-B card in the field. Since there is no NFC-B card in the field, the PN7150 activates the NFC-A card again, then the PN7150 activates the ISO-DEP interface and the DH can start to exchange data with the NFC-A card to proceed with the payment application.

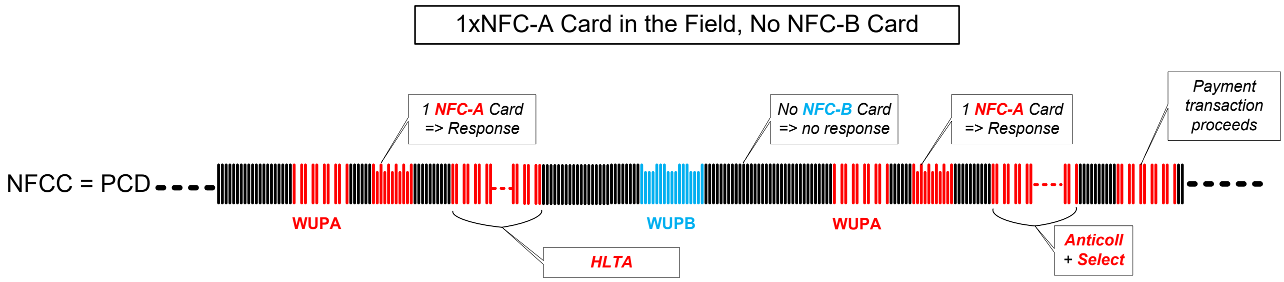

Fig 43. EMVCo polling with NFC-A card in the field

> ! In PN7150 the Low Power Card Detector is automatically disabled when the EMVCo profile is enabled, since these 2 features are conflicting if simultaneously enabled.

9.5.1.2 Notification for RF technology collision

When the EMVCo polling loop profile is activated, PN7150 will activate the ISO-DEP RF Interface through *RF_INTF_ACTIVATED_NTF* only when there is 1 single card in the field, whatever the technology (NFC-A or NFC-B).

When PN7150 detects a collision on RF (either in one technology or between technologies), it will report a special Status in the *CORE_GENERIC_ERROR_NTF*: *STATUS_EMVCo_PCD_COLLISION*. The current state will remain *RFST_DISCOVERY*, as graphically described in Fig 35. The identifier of this proprietary Status is defined in →0.Note that if the cards remain in the RF Field, PN7150 will keep sending the *CORE_GENERIC_ERROR_NTF* with status *STATUS_EMVCo_PCD_COLLISION* at each polling loop: this can be used as a presence check mechanism.

When the EMVCo profile for Poll Mode is activated and PN7150 has detected a single PICC (i.e. no collision) but it is unable to properly activate this PICC, then PN7150 will send a *CORE_GENERIC_ERROR_NTF* with status *DISCOVERY_TARGET_ACTIVATION_FAILED* as defined in [NCI].

9.5.1.3 Modification of the NCI RF State Machine in case of failure during data exchange

When the EMVCo profile for Poll Mode is activated, the PN7150 has to comply with tight timings verified during the EMVCo PCD certification. In case the RF link with the PICC is broken, the regular way to behave according to NCI is that the PN7150 will detect a time-out or an unrecoverable protocol error and send then a *CORE_INTERFACE_ERROR_NTF* with the appropriate status. It is then up to the DH to stop the RF Discovery with *RF_DEACTIVATE_CMD*(*IDLE*) and to restart the RF Discovery with *RF_DISCOVER_CMD*. Unfortunately, the time required to execute this sequence is highly dependent on the DH latency and it is often not possible to match the timings expected and checked by the EMVCo PCD certification.

To solve this issue, NXP has decided to add a transition from the *RFST_POLL_ACTIVE* to *RFST_DISCOVERY*, triggered by the sending of the

RF_DEACTIVATE_NTF(Discovery, Link Loss). In such a way, when PN7150 has detected a timeout or an unrecoverable protocol error during the RF communication with the PICC, it will autonomously come back to *RFST_DISCOVERY*, switching off the RF Field, as requested by EMVCo and then restarting the Polling phase in a timely manner, as requested by EVMCo.

This new transition is graphically described in Fig 35.

9.5.2 EMVCo profile in Listen Mode

To be compliant to the EMVCo certification tests emulating an EMVCo PICC, PN7150 has to behave as a single PICC based on either technology NFC-A or NFC-B.

In order to solve this issue, PN7150 comes with a specific configuration parameter: *LISTEN_PROFILE_SEL_CFG*, detailed in section →10.2.2.

Thanks to this parameter, a specific EMVCo PICC profile can be activated such that PN7150 will "hide" the non-yet-selected technology to the EMVCo PCD. Once this parameter is activated, the PICC selection sequence is as follows (assuming NFC-A is selected first):

- Once NFC-A has been selected by the PCD through the REQA command, PN7150 disables the NFC-B card emulation so that the REQB command sent later on by the EMVCo PCD gets no answer.
- The payment transaction can then successfully go through based on technology NFC-A.
- PN7150 waits then for an RF Field off/on sequence before enabling the non-selected technology (NFC-B) again.

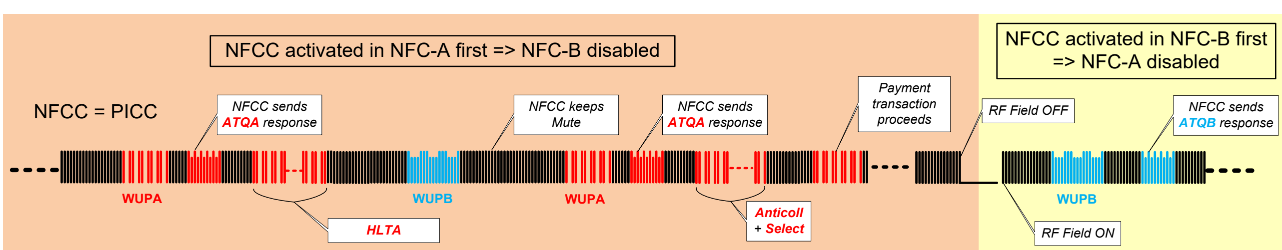

Fig 44. EMVCo Listen with first NFC-A activated by the PCD then NFC-B activated, after field off/on sequence

9.6 [PN7150-NCI] extension: Power optimization

PN7150 offers a standby mode, which can be activated together with the RF Discovery, such that the overall power consumption is significantly reduced.

One dedicated proprietary function is added to enable/disable this standby mode: *CORE_SET_POWER_MODE*.

9.6.1 CORE_SET_POWER_MODE Command/Response

> **!** The Standby Mode is enabled by default. Given the very strong impact on the power consumption, disabling the Standby Mode should be restricted to debug sessions.

Table 78. CORE_SET_POWER_MODE_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x00	1	Command to request the PN7150 to enable/disable the Standby Mode

Table 79. CORE_SET_POWER_MODE_CMD parameter

Payload Field(s)	Length	Value/Description	
Mode	1 Octet	0x00	Standby Mode disabled
		0x01	Standby Mode enabled
		0x03-0xFF	RFU

Table 80. CORE_SET_POWER_MODE_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x00	1	Response to inform the DH of the status of the CORE_SET_POWER_MODE_CMD.

Table 81. CORE_SET_POWER_MODE_RSP parameter

Payload Field(s)	Length	Value/Description	
Status	1 Octet	0x00	STATUS_OK
		0x06	STATUS_SEMANTIC_ERROR
		0x09	STATUS_INVALID_PARAM
		Others	Forbidden

9.6.2 Standby wake-up

The PN7150 wakes-up from standby when one of the following event occurs:

- Regular polling-loop starts. When the DH has served the PN7150 with a *NCI_RF_DISCOVER_CMD* command, the PN7150 enters into the standby mode and automatically leave the low power mode after the period defined by *TOTAL_DURATION*.
- RF level detector triggered. An external field has been introduced in the NFC volume during the standby period of the polling loop and at least one listen phase has been requested by the *NCI_DISCOVER_CMD*.
- Host interface activity detected. See →3.3 section.

10. Configurations

> **!** When the DH needs to update the value of the parameters described hereafter, it shall send a *CORE_RESET_CMD*/*CORE_INIT_CMD* sequence after the *CORE_SET_CONFIG_CMD*, to ensure that the new value is used for the parameters.
>
> If numerous parameters are updated thanks to multiple *CORE_SET_CONFIG_CMD* commands, a single *CORE_RESET_CMD*/ *CORE_INIT_CMD* sequence is enough after the last *CORE_SET_CONFIG_CMD*.

> **!** Any *CORE_SET_CONFIG_CMD* to one of the following parameters or to the [NCI] standard parameters will trigger an EEPROM write cycle. Since the PN7150 EEPROM has a limited number of Erase/Write cycles (300 000), it is highly recommended to only use the *CORE_SET_CONFIG_CMD* during the NCI initialization sequence.

10.1 [PN7150-NCI] extension: System configurations

PN7150 offers several parameters used to configure the system aspects.

Table 82. Core configuration parameters

Name & Rights	Description	Ext. Tag	Len.	Default Value
CLOCK_REQUEST_CFG ***RW in E²PROM***	Indicates how the clock is requested to the DH by the PN7150. 0x00: Clock Request is disabled 0x01: Hardware-based Clock Request is enabled: CLKREQ pin set to high when clock requested, otherwise it is set to hi-Z (High Impedance). 0x02-0xFF: RFU	0xA0 0x02	1	0x01

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>CLOCK_SEL_CFG
RW in E^{2}PROM</td><td>Input Clock selection & configuration for the internal 13.56MHz CLOCK
<table>
<tr><th>Bits [4:3]</th><th>Clk Source</th><th>Description</th></tr>
<tr><td>01b</td><td>XTAL</td><td>A 27.12MHz quartz has to be connected to PN7150</td></tr>
<tr><td>10b</td><td>PLL</td><td>A clean clock signal has to be directly provided on the Clock pad (bits [2:0] have to be configured in addition to specify the clock value, see the table below)</td></tr>
<tr><td>11b</td><td>RFU</td><td></td></tr>
<tr><td>00b</td><td>RFU</td><td></td></tr>
</table>
When the PLL is used, the bits [2:0] have to be configured according to the following table, depending on the clock provided to PN7150
<table>
<tr><th>Bits [2:0]</th><th>Clk In</th></tr>
<tr><td>000b</td><td>13.0 MHz</td></tr>
<tr><td>001b</td><td>19.2 MHz</td></tr>
<tr><td>010b</td><td>24 MHz</td></tr>
<tr><td>011b</td><td>26.0 MHz</td></tr>
<tr><td>100b</td><td>38.4 MHz</td></tr>
<tr><td>101b</td><td>52 MHz</td></tr>
<tr><td>110b-111b</td><td>RFU</td></tr>
</table>
</td><td>0xA0 0x03</td><td>1</td><td>0x08</td></tr>
<tr><td>CLOCK_TO_CFG
RW in E^{2}PROM</td><td>Indicates the timeout value to be used for clock request acknowledgment (from 1.53ms to 10 ms in steps of 330µs).

So the actual Time Out value (in µs) is given by the following formula:TimeOut (µs) = 1200 + (CLOCK_TO_CFG)*330

Minimum value is 01. Value 0x00 SHALL NOT be used, otherwise there is no timeout (no wait time). In this case the PLL is started immediately without waiting for the external sys_clock.

Maximum value to be used is 0x06, to ensure the NFCC is ready to reply 5ms after an external field on.</td><td>0xA0 0x04</td><td>1</td><td>0x01</td></tr>
</table>

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>IRQ_POLARITY_CFG
RW in E²PROM</td><td>Configuration of the IRQ pin polarity
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td>I²C transport fragmentation
'1' => enabled,
'0'=> disabled</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>IRQ PIN polarity config.</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td></td><td>0</td><td>0</td><td></td><td>0</td><td>All these bits SHALL be set to logical ‘0’ (RFU)</td></tr>
</table>
b1=’0’ => PN7150 requests to transmit when IRQ pin = ’1’.
b1=’1’ => PN7150 requests to transmit when IRQ pin = ‘0’.</td><td>0xA0 0x05</td><td>1</td><td>0x00</td></tr>
<tr><td>VBAT_MONITOR_EN_CFG
RW in E²PROM</td><td>To Enable/Disable the Battery monitor & configure the Threshold
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>Vbat Monitor Enable</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>Vbat Monitor Threshold</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td>RFU</td></tr>
</table>
b0: ‘1’ to Enable, ‘0’ to disable.
b1: ‘1’ to set the threshold to 2.3V and ‘0’ to set it to 2.75V.
Note: in *NCI_RFST_DISCOVERY* state, setting this parameter will be rejected by the NFCC with an INVALID PARAM status ‘0x09’ instead of SEMANTIC ERROR status ‘0x06’.</td><td>0xA0 0x06</td><td>1</td><td>0x00</td></tr>
<tr><td>VEN_CFG
RW in E²PROM</td><td>Configures the internal VEN signal, in case the VEN pin driver is NOT supplied from PVDD. In such a case, when PVDD is switched OFF, the VEN pin level in unknown, so the internal VEN signal is defined by one bit in an internal register (VEN_Value) while the VEN pin has to be pulled-down (to avoid leakages) thanks to a 2nd bit in the same register (VEN_Pulld) which has then to be set to '1' to activate the Pull Down. These 2 bits can be configured through NCI thanks to VEN_CFG LSbits, according to the following table:
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>VEN_Value</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>VEN_Pulld</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td>RFU</td></tr>
</table>
Note, in order to **force** a certain VEN value to be used internally (no matter which state the external VEN pin level is in) the VEN_Pulld value HAS to be set. Only if VEN_Pulld is set and PVDD is switched off the internal VEN state will be forced to what is specified in VEN_Value.</td><td>0xA0 0x07</td><td>1</td><td>0x03</td></tr>
</table>

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>TO_BEFORE_STDBY_CFG
RW in E^{2}PROM</td><td>Timeout used to wait after last DH-NFCEE communication before going into standby (from 0 to 65.536s in steps of 1ms).
Applies only when the discovery is stopped and standby mode is activated by *SET_PWR_MODE_CMD*.
Pay attention that the parameter value is defined in little endian (LSB first).</td><td>0xA0 0x09</td><td>2</td><td>0x03E8 (1s)</td></tr>
<tr><td>PAD_SLEW_RATE_CFG
RW in E^{2}PROM</td><td>Parameter used to configure the slew rate of the pads, on a per pad basis:
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td>CLK_REQ</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>IRQ</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td></td><td>0</td><td>RFU</td></tr>
</table>
For each of the pads, '1' => fast slew rate, '0' => slow slew rate.</td><td>0xA0 0x0A</td><td>1</td><td>0x00</td></tr>
<tr><td>RF_TRANSITION_CFG
RW in E^{2}PROM</td><td>TLV parameter to configure the RF transitions: see chapter →10.3</td><td>0xA0 0x0D</td><td></td><td></td></tr>
</table>

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>PMU_CFG
RW in E²PROM</td><td>Configuration of the Power Management Unit (PMU)
Byte 0:
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>VBAT1 connected to 5V
0 - CFG1, 1 - CFG2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>1</td><td>0</td><td>RFU</td></tr>
</table>
Byte 1:
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td>TVDD monitoring threshold:
0 - 3.6V (CFG1, CFG2)
1 - 5V (CFG2)</td></tr>
<tr><td></td><td></td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td>TxLDO Voltage in card mode communication:
000: 3V (CFG1, CFG2)
001: 3.3V (CFG1, CFG2)
010: 3.6V (CFG1, CFG2)
011: 4.5V (CFG2)
100: 4.7V (CFG2)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td>X</td><td>X</td><td>TxLDO Voltage in reader mode communication:
000: 3V (CFG1, CFG2)
001: 3.3V (CFG1, CFG2)
010: 3.6V (CFG1, CFG2)
011: 4.5V (CFG2)
100: 4.7V (CFG2)</td></tr>
<tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>RFU</td></tr>
</table>
Byte 2: RFU. Must be 0x00 for CFG1 and 0x01 in CFG2.</td><td>0xA0 0x0E</td><td>3</td><td>0x020900
(CFG1)</td></tr>
<tr><td>DH_EEPROM_AREA_2
RW in E²PROM</td><td>32-Byte EEPROM area dedicated to the DH to store/retrieve non-volatile data. The 32 Bytes have to be read (CORE_GET_CONFIG_CMD) or written (CORE_SET_CONFIG_CMD) is a row: it is not possible to access only a subset of these 32 Bytes.</td><td>0xA0 0x14</td><td>32</td><td></td></tr>
<tr><td>DYN_LMA_SETTINGS_CFG
RW in E²PROM</td><td>Parameter used to Read/write the Configuration as well as the Lookup table for the dynamic LMA feature</td><td>0xA0 0x92</td><td>68</td><td>See Table 84</td></tr>
</table>

Table 83. DYN_LMA_SETTINGS_CFG Description

Bytes	Description	Len.	Default Value
0 … 1	RFU	2	N/A
2	bLutSize: Size of LUT, DO NOT MODIFY this parameter	1	0x10
3	bNbLutEntries: Number of entries in DynLma look up table . bits 0:3 = Number of Entries for Type A/B (0 means LMA disabled for this Type) . bits 4:7 = Number of Entries for Type F (0 means LMA disabled for this Type) The number of entries for Type A/B + Type F shall not exceed the Total number of Entries. The Entries for TypeF follow the ones for Type A/B. This means if number of entries for Type A/B is 8 Entry 8 is the first for TypeF	1	0x00
4	dwLutEntry0: bits 20:18 = TXLDO output voltage: PMU_TXLDO_CONTROL_REG/TXLDO_SELECT bits 17:16 = CLIF_ANA_TX_AMPLITUDE_REG / TX_CW_AMPLITUDE_ALM_CM bit 15 = CLIF_TX_CONTROL_REG / TX_ALM_TYPE_SELECT bits 14:10 = CLIF_ANA_TX_AMPLITUDE_REG / TX_RESIDUAL_CARRIER bits 09:00 = AGC_VALUE	4	0x037C02
…	dwLutEntry…	4	N/A
64 ... 67	dwLutEntryF	4	0x000032

10.2 [PN7150-NCI] extension: RF Discovery configuration

10.2.1 Poll Mode

Several configuration parameters are required for the Poll Mode in RF discovery:

Table 84. Poll Mode configuration

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>TAG_DETECTOR_CFG
RW in E²PROM</td><td>Tag detector enabling/disabling as follows:
<table>
<tr><th colspan="8">Bit Mask</th><th>Description</th></tr>
<tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td>Detection based on the AGC</td></tr>
<tr><td>X</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Activation of the Trace mode</td></tr>
<tr><td></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td></td><td>RFU</td></tr>
</table>
'1' => Enabled; '0' => Disabled</td><td>0xA0 0x40</td><td>1</td><td>0x00</td></tr>
<tr><td>TAG_DETECTOR_
THRESHOLD_CFG
RW in E²PROM</td><td>Sets the detection level.</td><td>0xA0 0x41</td><td>1</td><td>0x04</td></tr>
<tr><td>TAG_DETECTOR_
PERIOD _CFG
RW in E²PROM</td><td>Time in steps of 8us to wait before reading the AGC value.</td><td>0xA0 0x42</td><td>1</td><td>0x0F</td></tr>
</table>

Name & Rights	Description	Ext. Tag	Len.	Default Value
TAG_DETECTOR_ FALLBACK_CNT_CFG ***RW in E^{2}PROM***	Parameter used to configure the "Hybrid" mode ton insert a regular Polling cycle every N pulses generated by the LPCD: 0x00: Hybrid mode disabled: LPCD only, no regular Polling cycle unless an "object" is detected by the LPCD. 0x02- 0xFF: Hybrid mode enabled, inserting a regular Polling cycle every 'N' pulses of LPDC. 'N' is coded by the value assigned to *TAG_DETECTOR_FALLBACK_CNT_CFG* in decimal.	0xA0 0x43	1	0x50
POLL_PROFILE_SEL_ CFG ***RW in E^{2}PROM***	Discovery profile selection in Poll Mode as follows: 0x00: NFC FORUM profile All static configurations (Bail-out) will be set to the [NCI] default value (disabled). 0x01: EMVCo profile 0x02- 0xFF: RFU	0xA0 0x44	1	0x00
GT_NFC-AA_CFG ***RW in E^{2}PROM***	Guard time (in steps of 0.59µs) used between the start of unmodulated RF field & 1st command for Poll NFC-A Active (min='0001', max='FFFF')	0xA0 0x46	2	0x21C4 (5.1ms)
GT_NFC-AP_CFG ***RW in E^{2}PROM***	Guard time (in steps of 0.59µs) used between the start of unmodulated RF field & 1st command for Poll NFC-A Passive (min='0001', max='FFFF')	0xA0 0x47	2	0x2219 (5.15ms)
GT_NFC-B_CFG ***RW in E^{2}PROM***	Guard time (in steps of 0.59µs) used between the start of unmodulated RF field & 1st command for Poll NFC-B Passive (min='0001', max='FFFF')	0xA0 0x48	2	0x2219 (5.15ms)
GT_NFC-F_CFG ***RW in E^{2}PROM***	Guard time (in steps of 0.59µs) used between the start of unmodulated RF field & 1st command for Poll NFC-F Passive (min='0001', max='FFFF') Note: If previous phase on polling loop is a FeliCa Poll that fail on Timeout, you will see an additional 5 ms delay due to the FeliCa timeout itself	0xA0 0x49	2	0x878D (20.47ms)
GT_15693_CFG ***RW in E^{2}PROM***	Guard time (in ms) used between the start of unmodulated RF field & 1st command for Poll 15693 Passive (min='0001', max='FFFF')	0xA0 0x4A	2	0x07B8 (1.17ms)
PF_SYS_CODE_CFG ***RW in E^{2}PROM***	Discovery configuration parameters for Poll F: system code	0xA0 0x4C	2	0xFFFF
MFC_KEY-0_CFG ***WO[1] in E^{2}PROM***	Key 0, used in MIFARE Classic Authentication command.	0xA0 0x4D	6	0xA0A1 A2A3 A4A5
MFC_KEY-1_CFG ***WO[1] in E^{2}PROM***	Key 1, used in MIFARE Classic Authentication command.	0xA0 0x4E	6	0xD3F7 D3F7 D3F7
MFC_KEY-2_CFG ***WO[1] in E^{2}PROM***	Key 2, used in MIFARE Classic Authentication command.	0xA0 0x4F	6	0xFFFF FFFF FFFF

Name & Rights	Description	Ext. Tag	Len.	Default Value
MFC_KEY-3_CFG ***WO*[1] *in E²PROM***	Key 3, used in MIFARE Classic Authentication command.	0xA0 0x50	6	0xFFFF FFFF FFFF
MFC_KEY-4_CFG ***WO*[1] *in E²PROM***	Key 4, used in MIFARE Classic Authentication command.	0xA0 0x51	6	0xFFFF FFFF FFFF
MFC_KEY-5_CFG ***WO*[1] *in E²PROM***	Key 5, used in MIFARE Classic Authentication command.	0xA0 0x52	6	0xFFFF FFFF FFFF
MFC_KEY-6_CFG ***WO*[1] *in E²PROM***	Key 6, used in MIFARE Classic Authentication command.	0xA0 0x53	6	0xFFFF FFFF FFFF
MFC_KEY-7_CFG ***WO*[1] *in E²PROM***	Key 7, used in MIFARE Classic Authentication command.	0xA0 0x54	6	0xFFFF FFFF FFFF
MFC_KEY-8_CFG ***WO*[1] *in E²PROM***	Key 8, used in MIFARE Classic Authentication command.	0xA0 0x55	6	0xFFFF FFFF FFFF
MFC_KEY-9_CFG ***WO*[1] *in E²PROM***	Key 9, used in MIFARE Classic Authentication command.	0xA0 0x56	6	0xFFFF FFFF FFFF
MFC_KEY-10_CFG ***WO*[1] *in E²PROM***	Key 10, used in MIFARE Classic Authentication command.	0xA0 0x57	6	0xFFFF FFFF FFFF
MFC_KEY-11_CFG ***WO*[1] *in E²PROM***	Key 11, used in MIFARE Classic Authentication command.	0xA0 0x58	6	0xFFFF FFFF FFFF
MFC_KEY-12_CFG ***WO*[1] *in E²PROM***	Key 12, used in MIFARE Classic Authentication command.	0xA0 0x59	6	0xFFFF FFFF FFFF
MFC_KEY-13_CFG ***WO*[1] *in E²PROM***	Key 13, used in MIFARE Classic Authentication command.	0xA0 0x5A	6	0xFFFF FFFF FFFF
MFC_KEY-14_CFG ***WO*[1] *in E²PROM***	Key 14, used in MIFARE Classic Authentication command.	0xA0 0x5B	6	0xFFFF FFFF FFFF
MFC_KEY-15_CFG ***WO*[1] *in E²PROM***	Key 15, used in MIFARE Classic Authentication command.	0xA0 0x5C	6	0xFFFF FFFF FFFF
FSDI_CFG ***RW in E²PROM***	Frame Size value for the PN7150 to display in RATS or ATTRIB.	0xA0 0x5D	1	0x08
JEWEL_RID_CFG ***RW in E²PROM***	Parameter used to configure if the RID is sent on RF to the T1T by PN7150 during the RF activation or not: 0x01 => The RID is sent on RF to the T1T 0x00 => The RID is NOT sent on RF to the T1T	0xA0 0x5E	1	0x00

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td></td><td>In both cases, the RF_INTF_ACTIVATED_NTF will NOT embed the RID response from the T1T, as defined in [NCI].</td><td></td><td></td><td></td></tr>
<tr><td>FELICA_TSN_CFG
RW in E²PROM</td><td>TSN value transported by the PN7150 in the SENSF_REQ command: the DH defines the number of time slots for collision resolution.
!! This value has to be set to 0x03 for NFC FORUM compliance (DTA/Digital protocol tests) !!</td><td>0xA0 0x5F</td><td>1</td><td>0x00</td></tr>
<tr><td>TechDet_AFTER_LPCD_CFG
RW in E²PROM</td><td>Parameter used to configure the RF Discovery taking place right after the Low Power Card Detector has triggered a detection:
<table><tr><td colspan="8">Bit Mask</td><td>Description</td></tr><tr><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td>TechDet_PERIOD
In steps of 10ms</td></tr><tr><td></td><td></td><td></td><td></td><td></td><td>X</td><td>X</td><td>X</td><td>TechDet_NBR_RETRIES</td></tr></table>See →9.4.2 for more details on the use of this parameter.</td><td>0xA0 0x61</td><td>1</td><td>0x00</td></tr>
</table>

[1] WO (Write Only) parameters can only be written, using *CORE_SET_CONFIG_CMD*. PN7150 will always return *CORE_GET_CONFIG_RSP*(*STATUS_INVALID_PARAM*) to any attempt to read the value of the WO parameter.

10.2.2 Listen Mode

Table 85. Listen Mode Configuration

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>TO_RF_OFF_CFG
RW in E²PROM</td><td>Specifies the time out (in ms) applied by PN7150 before it restarts a Polling sequence, after it has detected a Field OFF in Listen Mode</td><td>0xA0 0x80</td><td>2</td><td>0x012C
(300 ms)</td></tr>
<tr><td>LISTEN_PROFILE_SEL_CFG
RW in E²PROM</td><td>Discovery profile selection in Listen Mode, as follows:<table><tr><td>0x00</td><td>NFC FORUM profile</td></tr><tr><td>0x01</td><td>EMVCo</td></tr><tr><td>0x02- 0xFF</td><td>RFU</td></tr></table></td><td>0xA0 0x81</td><td>1</td><td>0x01</td></tr>
<tr><td>LISTEN_ISODEP_FSCI_CFG
RW in E²PROM</td><td>Parameter to define the FSC parameter (RF Frame Size for the PICC), as defined in [14443-4]:<table><tr><td>0x00</td><td>FSC = 16</td></tr><tr><td>0x01</td><td>FSC = 24</td></tr><tr><td>0x02</td><td>FSC = 32</td></tr><tr><td>0x03</td><td>FSC = 40</td></tr><tr><td>0x04</td><td>FSC = 48</td></tr><tr><td>0x05</td><td>FSC = 64</td></tr><tr><td>0x06</td><td>FSC = 96</td></tr><tr><td>0x07</td><td>FSC = 128</td></tr><tr><td>0x08</td><td>FSC = 256</td></tr><tr><td>0x09- 0xFF</td><td>RFU</td></tr></table></td><td>0xA0 0x83</td><td>1</td><td>0x08</td></tr>
</table>

10.3 [PN7150-NCI] extension: Contactless Interface configurations

PN7150 offers multiple configuration options for the Contactless Interface, to allow an optimum match between the antenna characteristics and the transmitter and receiver in PN7150. A generic TLV mechanism has been defined to write the Contactless Interface settings. It relies on the [NCI] *CORE_SET_CONFIG_CMD* and is described hereafter:

Table 86. Mechanism to configure the RF transitions:

<table>
<tr><th>Name & Rights</th><th>Description</th><th>Ext. Tag</th><th>Len.</th><th>Default Value</th></tr>
<tr><td>RF_TRANSITION_CFG
RW in E²PROM</td><td>Parameter to configure one RF transition.
One transition will be coded as:
<table><tr><td>Transition ID (TID)</td><td>CLIF register offset (RO)</td><td>Register Value (RV)</td></tr><tr><td rowspan="3">1 Byte</td><td rowspan="3">1 Byte</td><td>1 Byte</td></tr><tr><td>2 Bytes</td></tr><tr><td>4 Bytes</td></tr></table></td><td>0xA0 0D</td><td>3, 4 or 6</td><td>N/A</td></tr>
</table>

Name & Rights	Description	Ext. Tag	Len.	Default Value
	The list of transition IDs and the appropriate values for the Register offset & its value is available in [AN 11755], as referenced in →15			

! *CORE_SET_CONFIG_CMD* command to set RF Transitions triggers internal EEPROM memory page write operation. To prevent memory corruption, any interruption of this command (between *CORE_SET_CONFIG_CMD* and *CORE_SET_CONFIG_RSP*) by hardware reset or power off MUST be prevented.

Thus, it is recommended to:

- Prevent re-applying RF Transitions parameters when not required (those parameters been stored in non-volatile memory, there are persistent even in case NCI CORE_RESET_CMD with option "reset configuration" is applied).
- Split the RF Transition settings into several *CORE_SET_CONFIG_CMD* commands to limit the time for the command treatment inside PN7150 (*CORE_SET_CONFIG_CMD* with only one RF Transition takes 2.7ms, 5.4ms in the specific case where the RF parameter resides in 2 separate Flash memory blocks)
- Avoid mixing RF Transition parameters with other parameters (not starting with address 0xA00D) in a same *CORE_SET_CONFIG_CMD* command

! PN7150B0HN/C11006 version only allows recovering from such memory corruption. Refer to 4.3.8.3 for more details about this mechanism.

! PN7150 only supports *RF_TRANSITION_CFG* with command *CORE_SET_CONFIG_CMD*. *CORE_GET_CONFIG_CMD* is not supported.

To read out the values a specific command *RF_GET_TRANSITION_CMD* is to be used.

Table 87. RF_GET_TRANSITION_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x14	2	The DH asks to read out the value of an RF Transition

Table 88. RF_ GET_TRANSITION_CMD parameters

Payload Field(s)	Length	Value/Description
RF Transition ID	1 Octet	RF Transition Identifier

Payload Field(s)	Length	Value/Description
CLIF Register Offset	1 Octet	Offset of the register to read out from the CLIF

Table 89. RF_ GET_TRANSITION_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x14	2	The PN7150 acknowledges the command received from the DH and sends the RF Transition value to the DH.

Table 90. RF_ GET_TRANSITION_RSP parameters

Payload Field(s)	Length	Value/Description
STATUS	1 Octet	One of the following Status codes, as defined in [NCI_Table1] 0x00: STATUS_OK 0x01: STATUS_REJECTED 0x06: STATUS_SEMANTIC_ERROR Others: Forbidden
RF Transition Length	1 Octet	Length of the following parameter (RF Transition Value): 0x01: 1 Octet to follow 0x02: 2 Octets to follow 0x04: 4 Octets to follow Others: RFU
RF Transition Value	1, 2 or 4 Octets	RF Transition Value ! Value coded in Little Endian.

11. Test Mode

11.1 Test Session

The PN7150 has the ability to generate a continuous PRBS pattern on the RF interface.

Whatever the test command used by the DH, it is necessary to implement a "test session", which isolates the test mode from a regular "NCI session" of PN7150. This test session is defined thanks to the following sequence:

- Reset/Initialize the PN7150 using *CORE_RESET_CMD*/*CORE_INIT_CMD*
- Launch selected test function
- Get the response transporting executed test status
- Reset/ Initialize the PN7150 using *CORE_RESET_CMD*/*CORE_INIT_CMD* (except for *TEST_PRBS_CMD*, which requires a HW reset first to stop the pattern generation on RF).

11.2 TEST_PRBS_CMD/RSP

This command is used to start PRBS infinite stream generation:

Table 91. TEST_PRBS_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x30	6	Command to start PRBS generation

Table 92. TEST_PRBS_CMD parameters

Payload Field(s)	Length	Value/Description	
PRBS Mode	1 Octet	0x00	Firmware PRBS
		0x01	Hardware PRBS
PRBS type	1 Octet	0x00	PRBS9
		0x01	PRBS15
Technology to stream	1 Octet	0x00	Type A
		0x01	Type B
		0x02	Type F
Bitrate	1 Octet	0x00	106 kbps (Type A,B)
		0x01	212 kbps (Type A,B& F)
		0x02	424 kbps (Type A,B & F)
		0x03	848 kbps (Type A,B)
PRBS series length	2 Octets	A value between 0x0001 – 0x01FF	

Table 93. TEST_PRBS_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x30	1	PN7150 reports if the *TEST_PRBS_CMD* is successful or not.

Table 94. TEST_PRBS_RSP parameters

Payload Field(s)	Length	Value/Description	
STATUS	1 Octet	0x00	STATUS_OK
		0x06	STATUS_SYNTAX_ERROR
		0x09	STATUS_INVALID_PARAM
		Others	Forbidden

! The only way to stop the on-going PRBS pattern generation is to apply a HW reset (through the VEN pin).

11.3 TEST_ANTENNA_CMD/RSP

This command is used to execute the antenna self-test measurements, which allow to check that all the discrete components connected between PN7150 and the contactless antenna are properly soldered on the PCB.

Four different measurements are necessary to check the correct connection of all the discrete components, therefore a complete Antenna Self-Test requires to execute the *TEST_ANTENNA_CMD* 4 consecutive times, with a different set of parameters for each execution.

Table 95. TEST_ANTENNA_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x3D	2-4	Command to execute antenna self-test measurements.

Table 96. TEST_ANTENNA_CMD parameters

Payload Field(s)	Length	Value/Description	
Measurement ID	1 Octet	0x01	TxLDO current measurement
		0x02	AGC value reading
		0x04	AGC value reading with fixed NFCLD level
		0x20	Switch RF Field On/Off
		0x03, 0x05-0x1F, 0x21-0xFF	RFU
Parameters of individual test measurement	1-3 Octets	For individual test parameters please refer to →Table 98	

Table 97. Parameters to include in TEST_ANTENNA_CMD depending on the measurement to perform

Meas. ID	Measurement Description	Param. number	Parameter name	Length	Description	Typ. value
0x01	TxLDO current measurement	1	Wait_Time	1 Octet	Time to wait (in µs) before capturing the TX-LDO current	0x80
0x02	AGC value reading	1	Wait_Time	1 Octet	Time to wait (in µs) before capturing the AGC value	0xC8
		2	CLIF AGC input [7:0]	1 Octet	Value to write in CLIF AGC input register, bits [7:0]	0x60
		3	CLIF AGC input [9:8]	1 Octet	The 2 LSbits of parameter 3 are mapped on bits [9:8] of CLIF AGC input register. The 6 MSbits of parameter 3 have to be set to '0'.	0x03
0x04	AGC value reading with fixed NFCLD level	1	Wait_Time	1 Octet	Time to wait (in µs) before capturing the AGC value	0x20
		2	CLIF ANA NFCLD value [3:0]	1 Octet	The 4 LSbits of parameter 2 are mapped on bits [3:0] of CLIF ANA NFCLD input register. The 4 MSbits of parameter 2 have to be set to '0'	0x08
		3	Masked PMU TxLDO control bit [5]	1 Octet	bit [5] of parameter 3 is mapped to bit [5] in PMU TxLDO cntrl register. All other bits in parameter 3 ([7:6] & [4:0]) have to be set to '0'	0x20
0x20[1]	Switch RF Field On/Off	1	RF Field Generation	1 Octet	'1' => RF Field is generated '0' => RF Field is not generated	

! [1] Option 0x20 (Switch RF Field On/Off) absolutely requires to first disable the Standby mode, thanks to the *CORE_SET_POWER_MODE_CMD* (see →9.6.1).

Table 98. TEST_ANTENNA_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x3D	5	PN7150 returns individual measurement status code and the result of the measurement.

Table 99. TEST_ANTENNA_RSP parameters

Payload Field(s)	Length	Value/Description
STATUS	1 Octet	0x00: STATUS_OK 0x01: Test execution rejected (PN7150 in wrong state) 0x04: STATUS_TEST_EXEC_FAILED 0x09: STATUS_INVALID_PARAM Others: Forbidden
Result_Parameter_1	1 Octet	Value depending on the measurement performed : see →Table 101

Payload Field(s)	Length	Value/Description
Result_Parameter_2	1 Octet	Value depending on the measurement performed : see →Table 101
Result_Parameter_3	1 Octet	Value depending on the measurement performed : see →Table 101
Result_Parameter_4	1 Octet	Value depending on the measurement performed : see →Table 101

Table 100. Parameters provided in TEST_ANTENNA_RSP as a result of the measurement performed

Meas. ID	Measurement Description	Param. nbr	Parameter name	Length	Description
0x01	TxLDO current measurement	1	TxLDO output value	1 Octet	Raw value (RawVal) of TxLDO measurement (0x00-0x7F)
		2	Measured range	1 Octet	0x00: 50-100 mA Absolute value = 0.4 x RawVal + 50 [mA] 0x01: 20-70 mA Absolute value = 0.4 x RawVal + 20 [mA]
		3	RFU	1 Octet	
		4	RFU	1 Octet	
0x02	AGC value reading	1	AGC Value LSB	1 Octet	
		2	AGC Value MSB	1 Octet	
		3	RFU	1 Octet	
		4	RFU	1 Octet	
0x04	AGC value reading with fixed NFCLD level	1	AGC Value LSB	1 Octet	
		2	AGC Value MSB	1 Octet	
		3	RFU	1 Octet	
		4	RFU	1 Octet	
		4	AGC Value MSB	1 Octet	
0x20	Switch RF Field On/Off	1	RFU	1 Octet	
		2	RFU	1 Octet	
		3	RFU	1 Octet	
		4	RFU	1 Octet	

! RFU Bytes in *TEST_ANTENNA_RSP* can have any value from 0x00 to 0xFF.

11.4 TEST_GET_REGISTER_CMD/RSP

This command is used to retrieve the current Value of the AGC_VALUE_REGISTER.

Table 101. TEST_GET_REGISTER_CMD

GID	OID	Numbers of parameter(s)	Description
1111b	0x33	0	Command to retrieve the Value of the AGC_VALUE_REGISTER

Table 102. TEST_GET_REGISTER_CMD parameters

Payload Field(s)	Length	Value/Description
Fix parameters	4 Octet	The parameters have fixed values and shall be 0x40 0x00 0x40 0xD8.

Table 103. TEST_GET_REGISTER_RSP

GID	OID	Numbers of parameter(s)	Description
1111b	0x33	1	4 Bytes containing the current Value of AGC_VALUE_REG

12. PN7150 Practical approach

12.1 Basic examples for Reader/Writer Mode

12.1.1 R/W Mode with 1 NFC endpoint

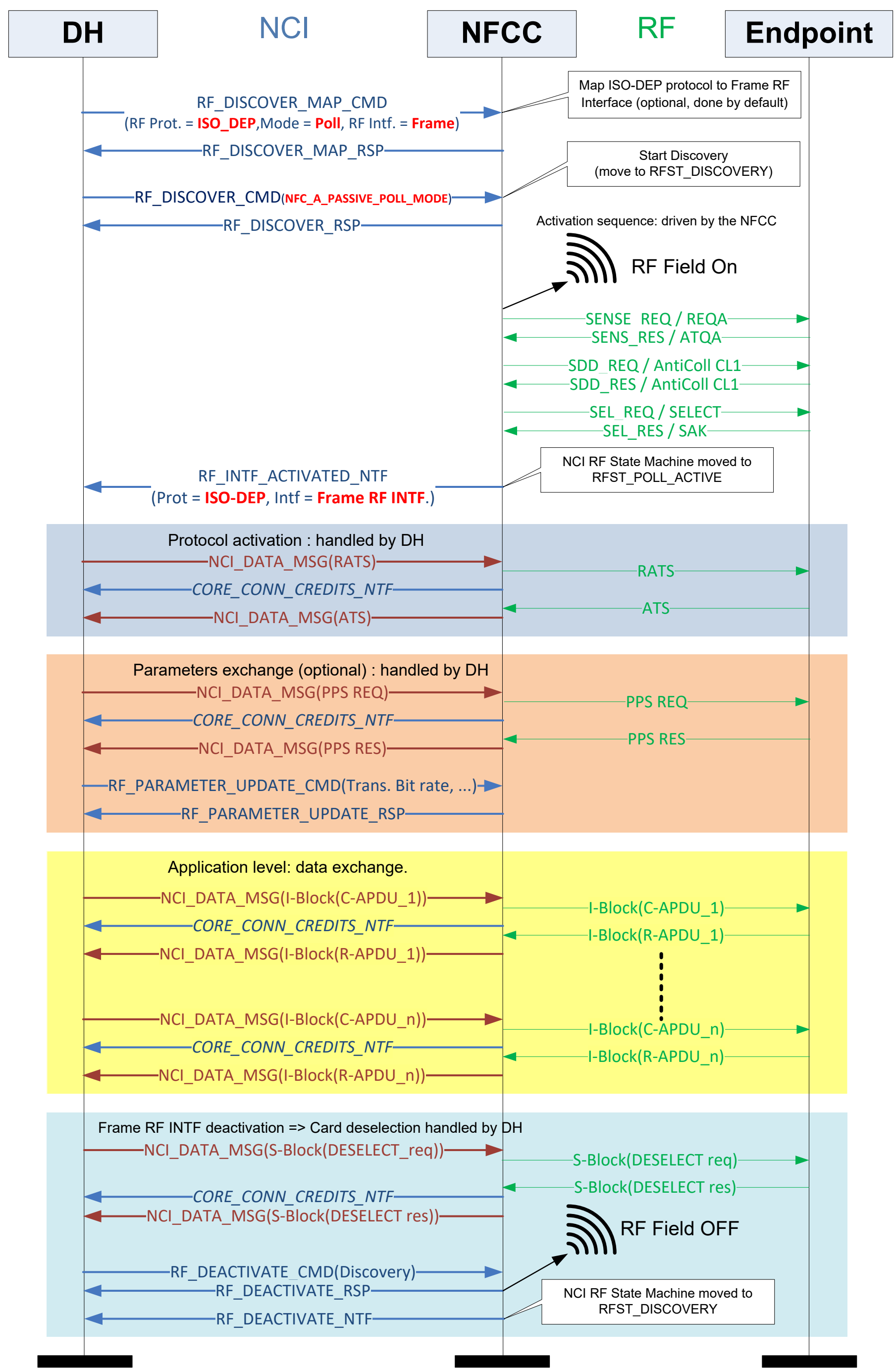

Fig 45. Poll Mode, NFC-A, ISO-DEP protocol / RF Frame Interface

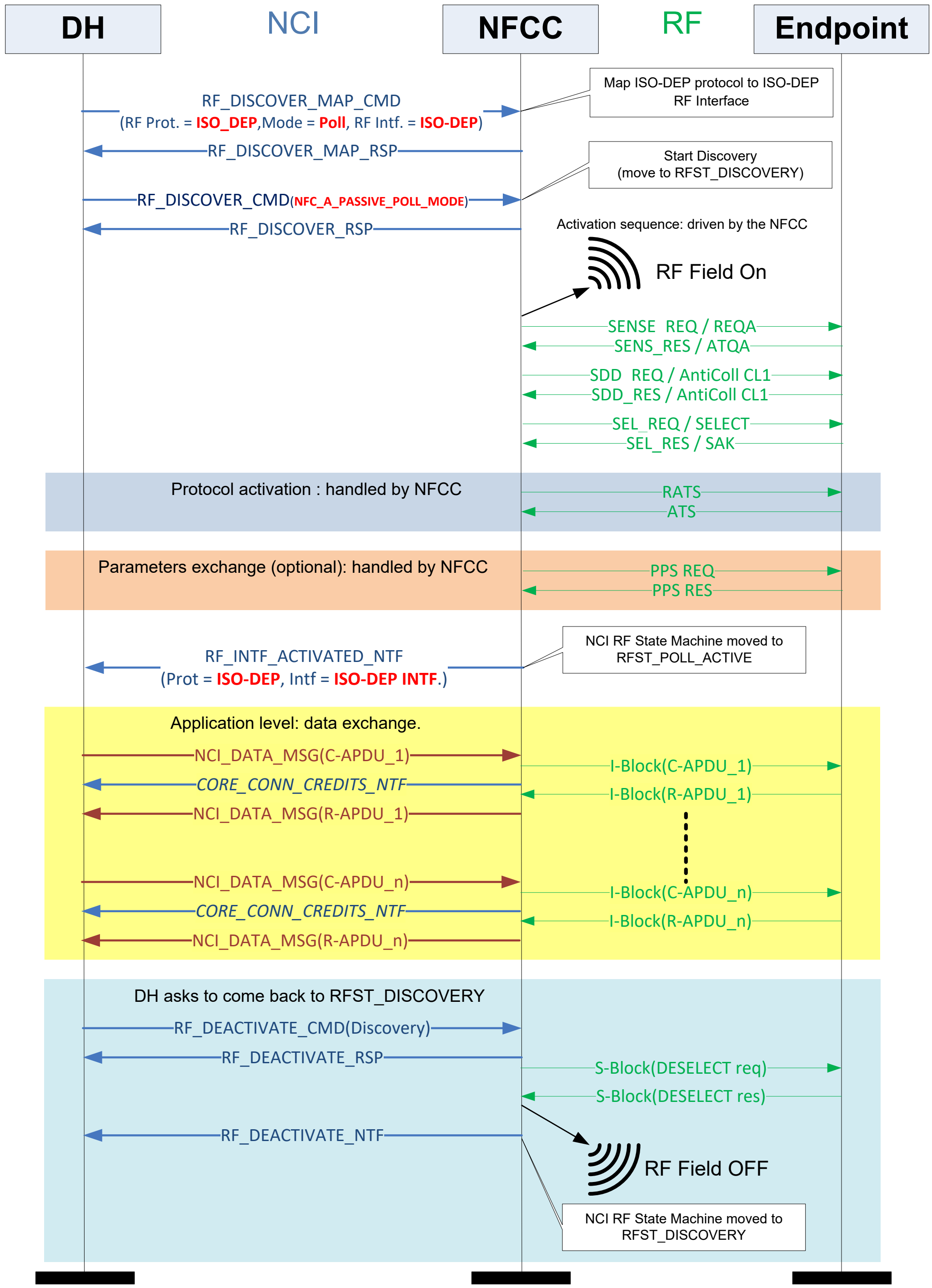

Fig 46. Poll Mode, NFC-A, ISO-DEP protocol / ISO-DEP Interface

12.1.2 R/W Mode with 2 NFC endpoints

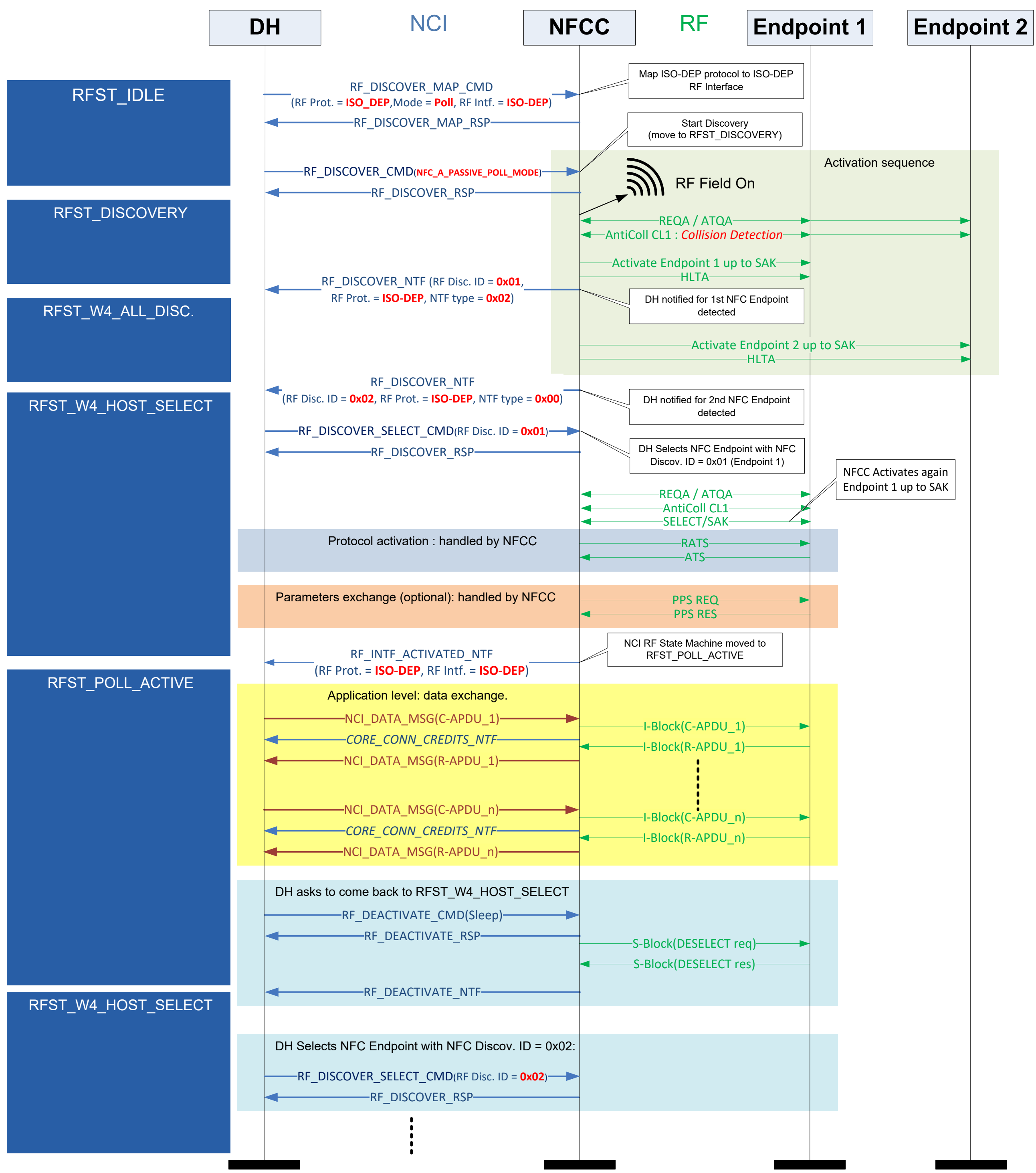

Fig 47. Poll Mode, 2 NFC-A Cards, ISO-DEP protocol / ISO-DEP Interface

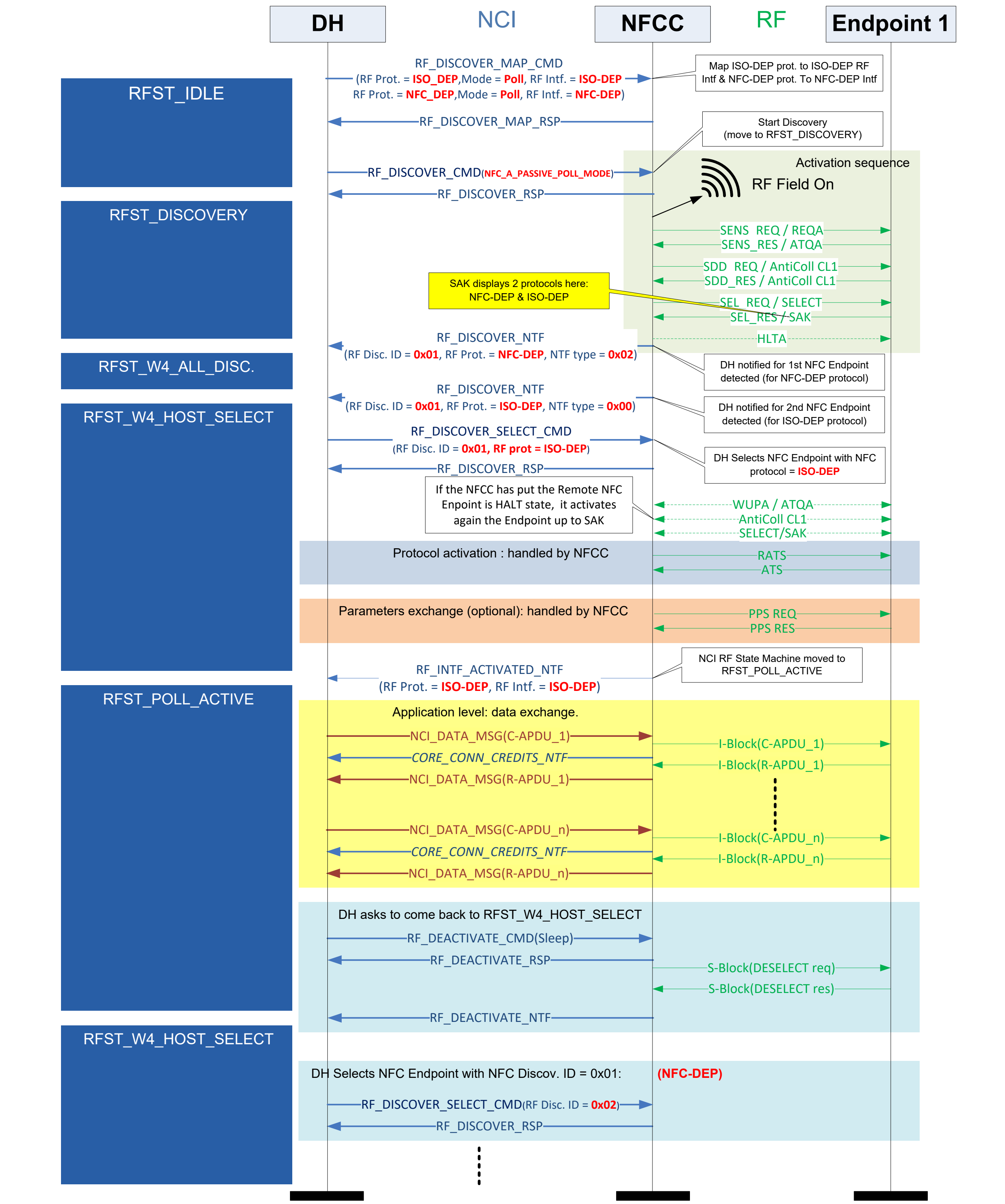

Fig 48. Poll Mode, 1 NFC-A Device, 2 RF protocols (merged SAK)

12.2 Basic examples for Card Emulation Mode

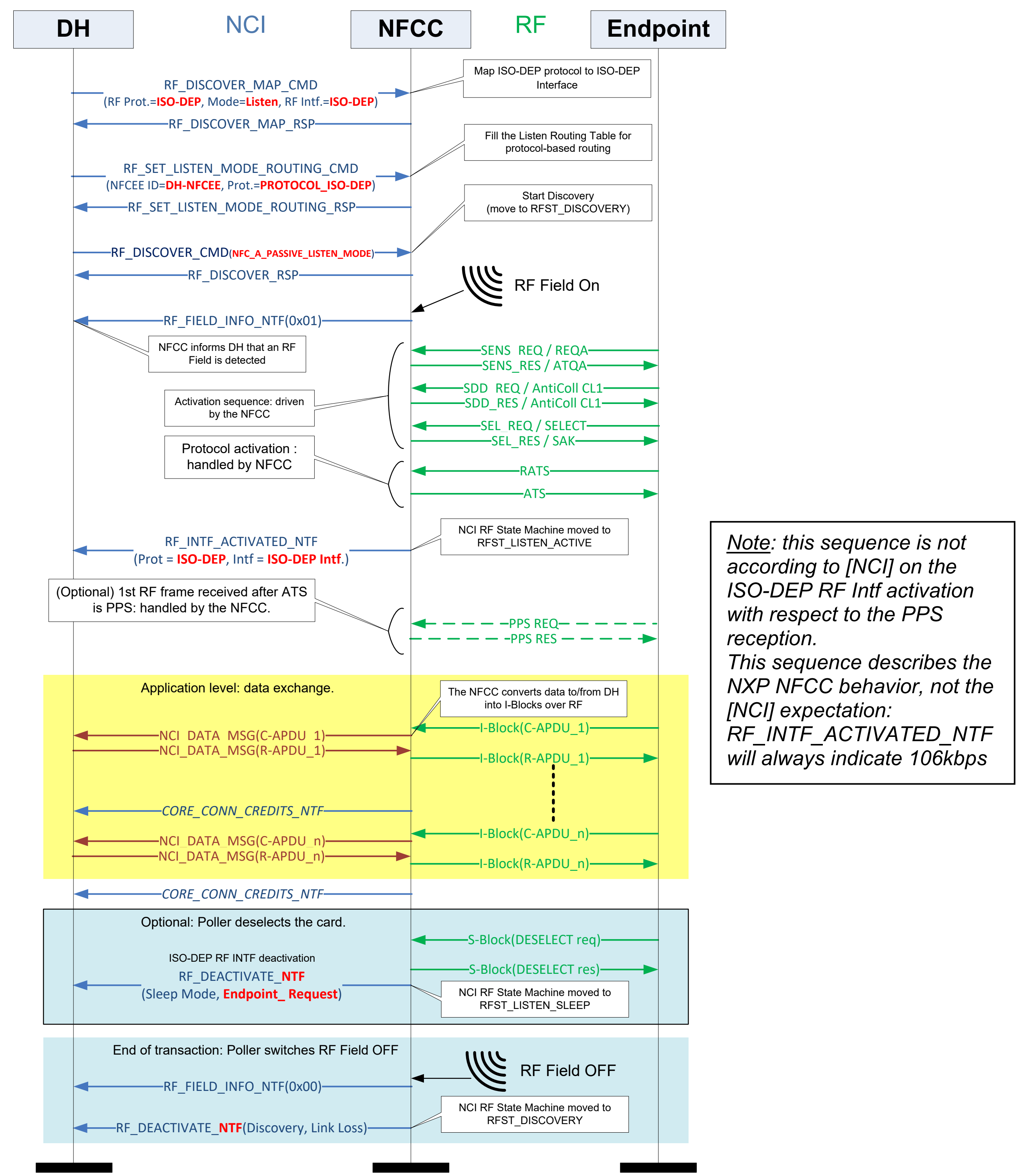

Fig 49. Card Emulation, NFC-A, ISO-DEP protocol / ISO-DEP Interface, optional PPS

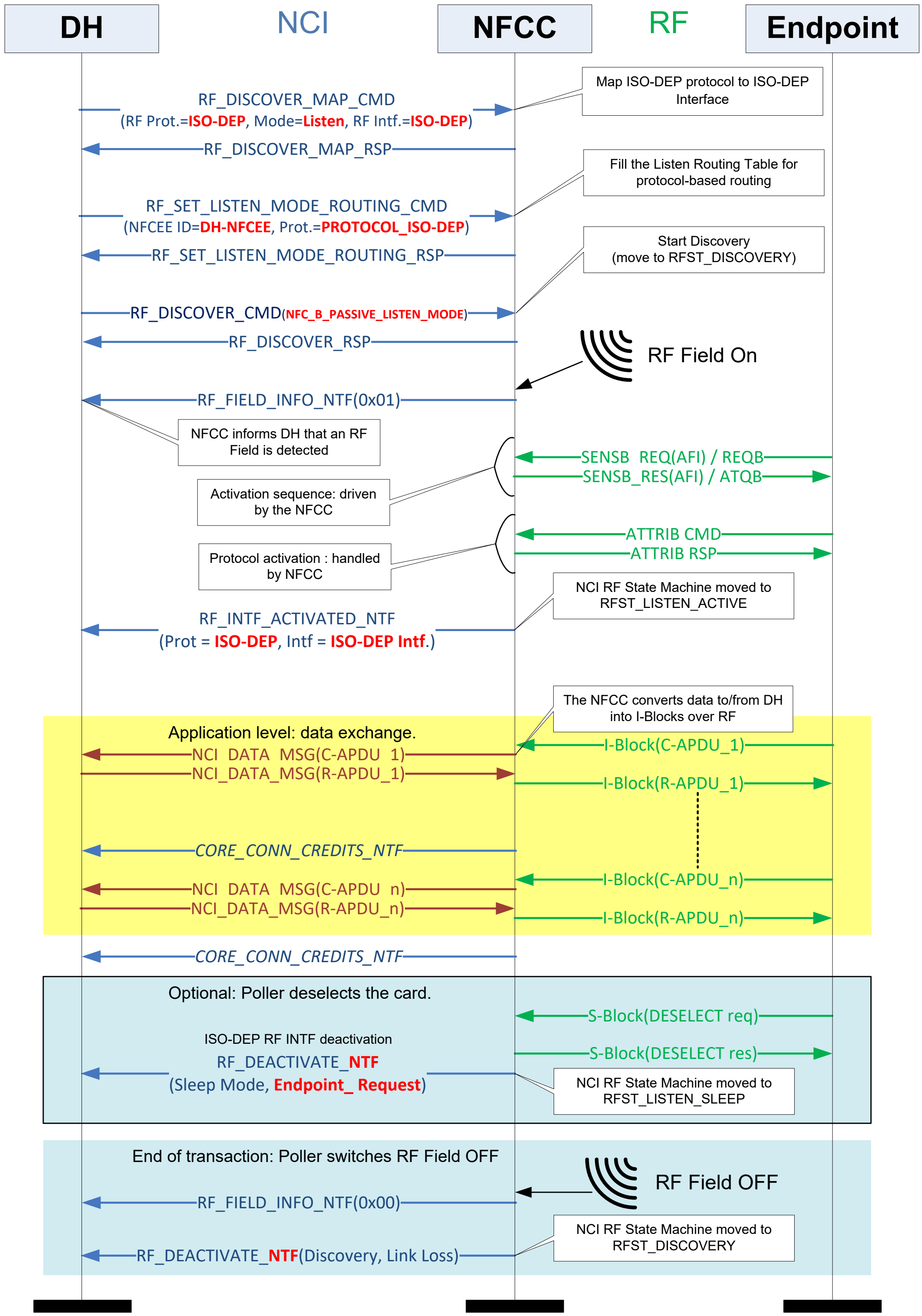

Fig 50. Card Emulation, NFC-B, ISO-DEP protocol / ISO-DEP Interface

12.3 Basic examples for P2P Passive Mode

12.3.1 Target in P2P Passive Mode / NFC-A @ 106kbps

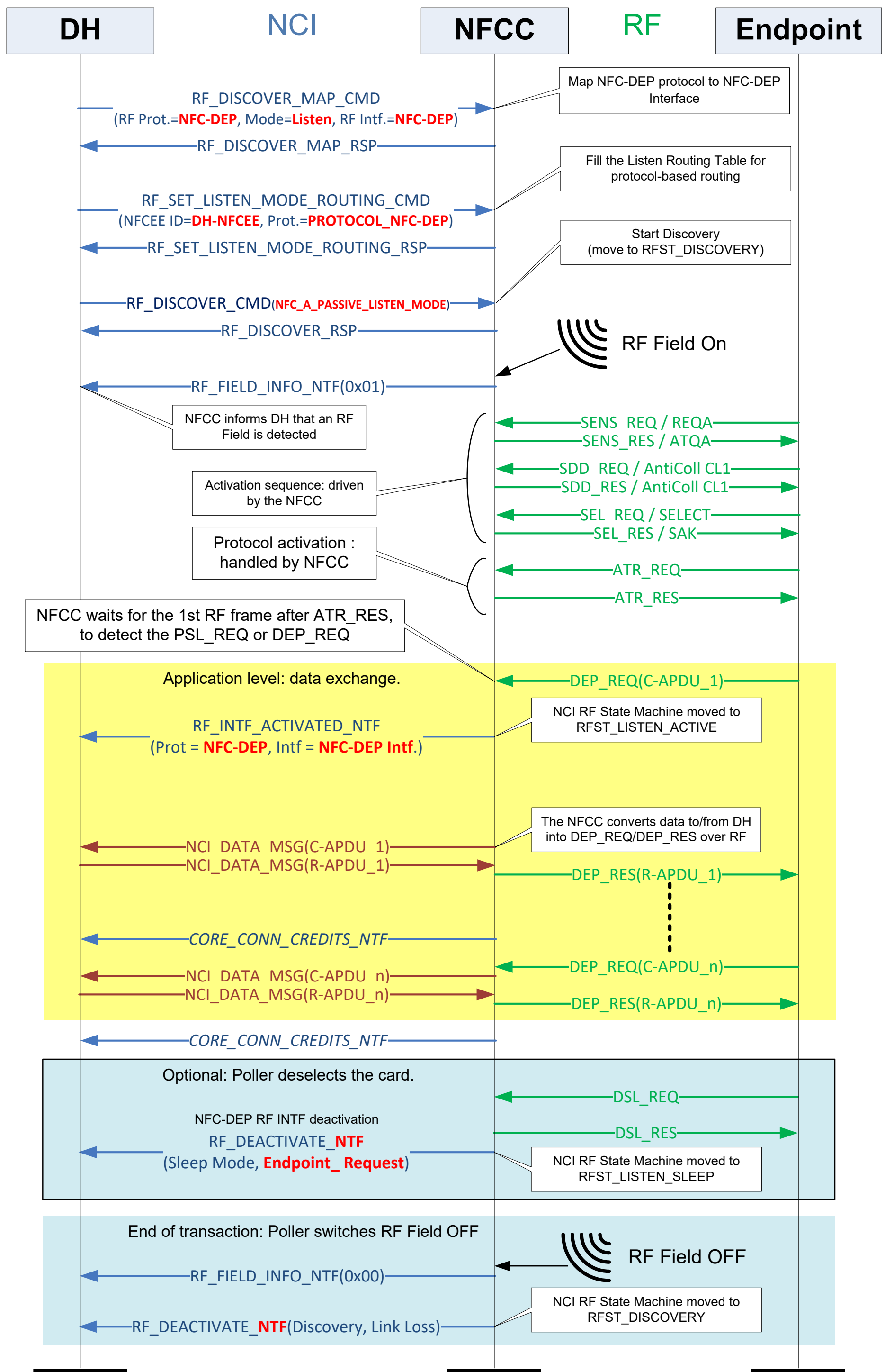

Fig 51. P2P/Target, NFC-A Passive, NFC-DEP protocol / NFC-DEP Interface, NO PSL

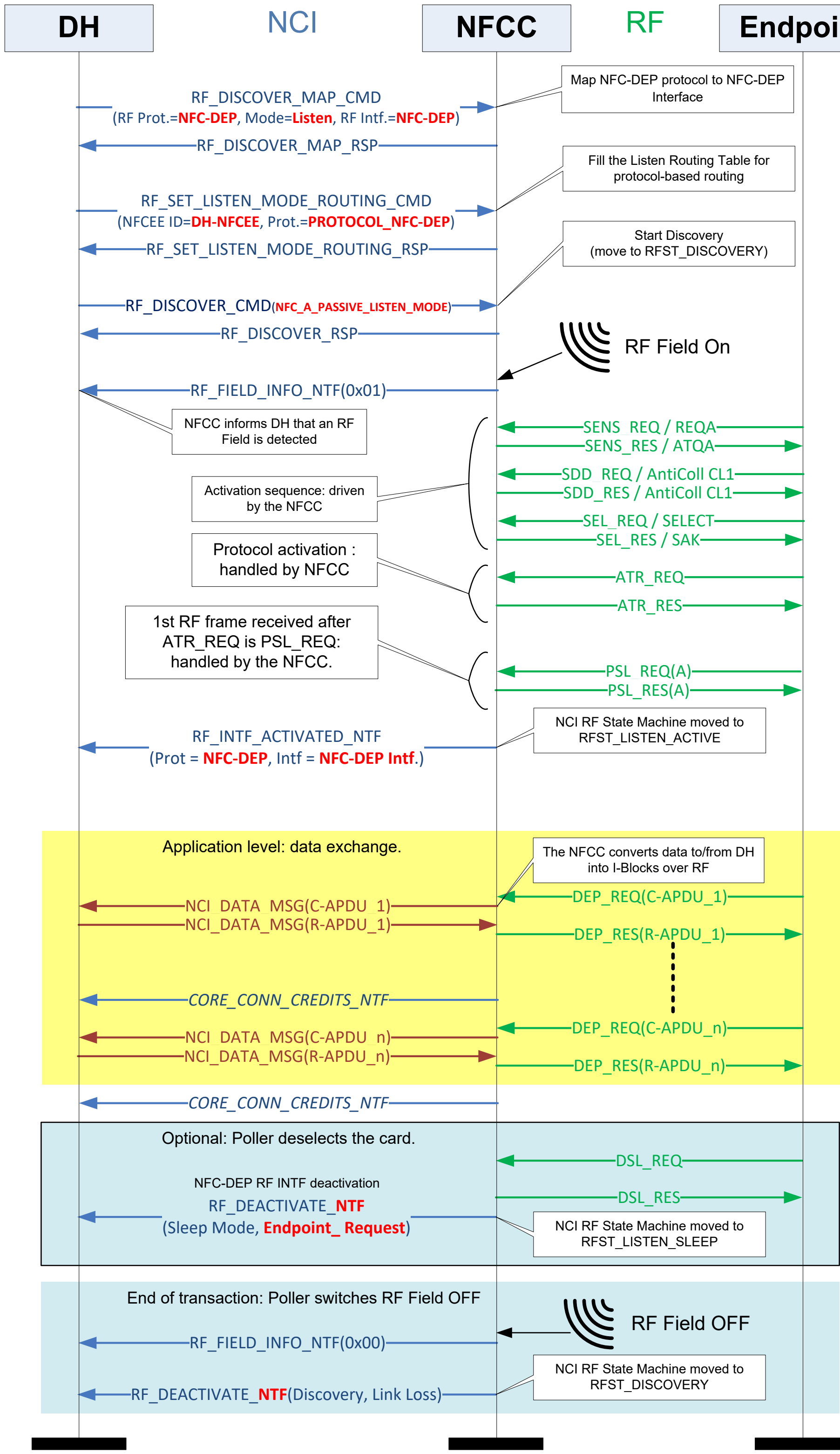

Fig 52. P2P/Target, NFC-A Passive, NFC-DEP protocol / NFC-DEP Interface + PSL

12.3.2 Initiator in P2P Passive Mode

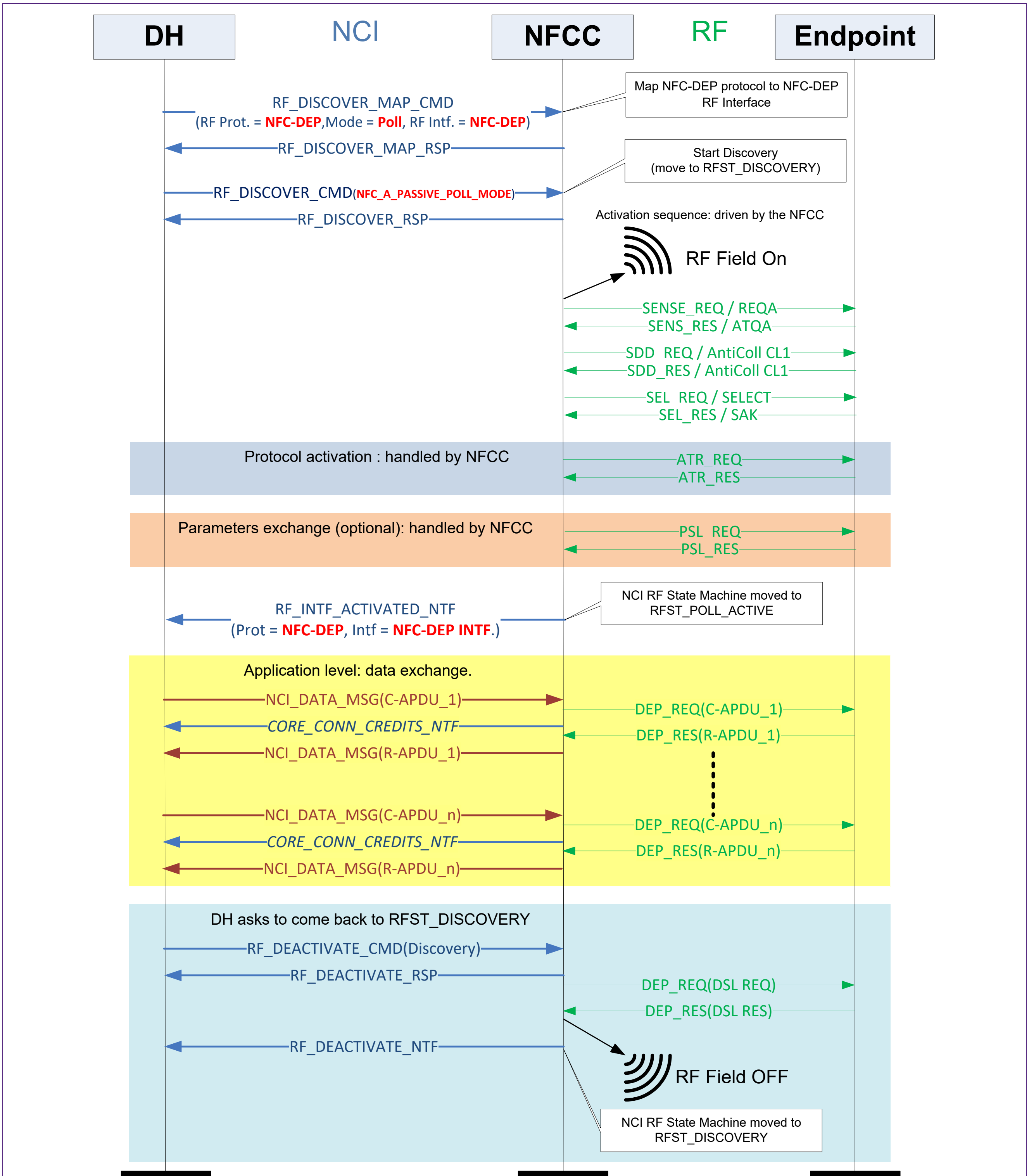

Fig 53. P2P/Initiator, NFC-A Passive, NFC-DEP protocol / NFC-DEP RF Interface

12.4 Basic examples for P2P Active Mode

12.4.1 Target in P2P Active Mode

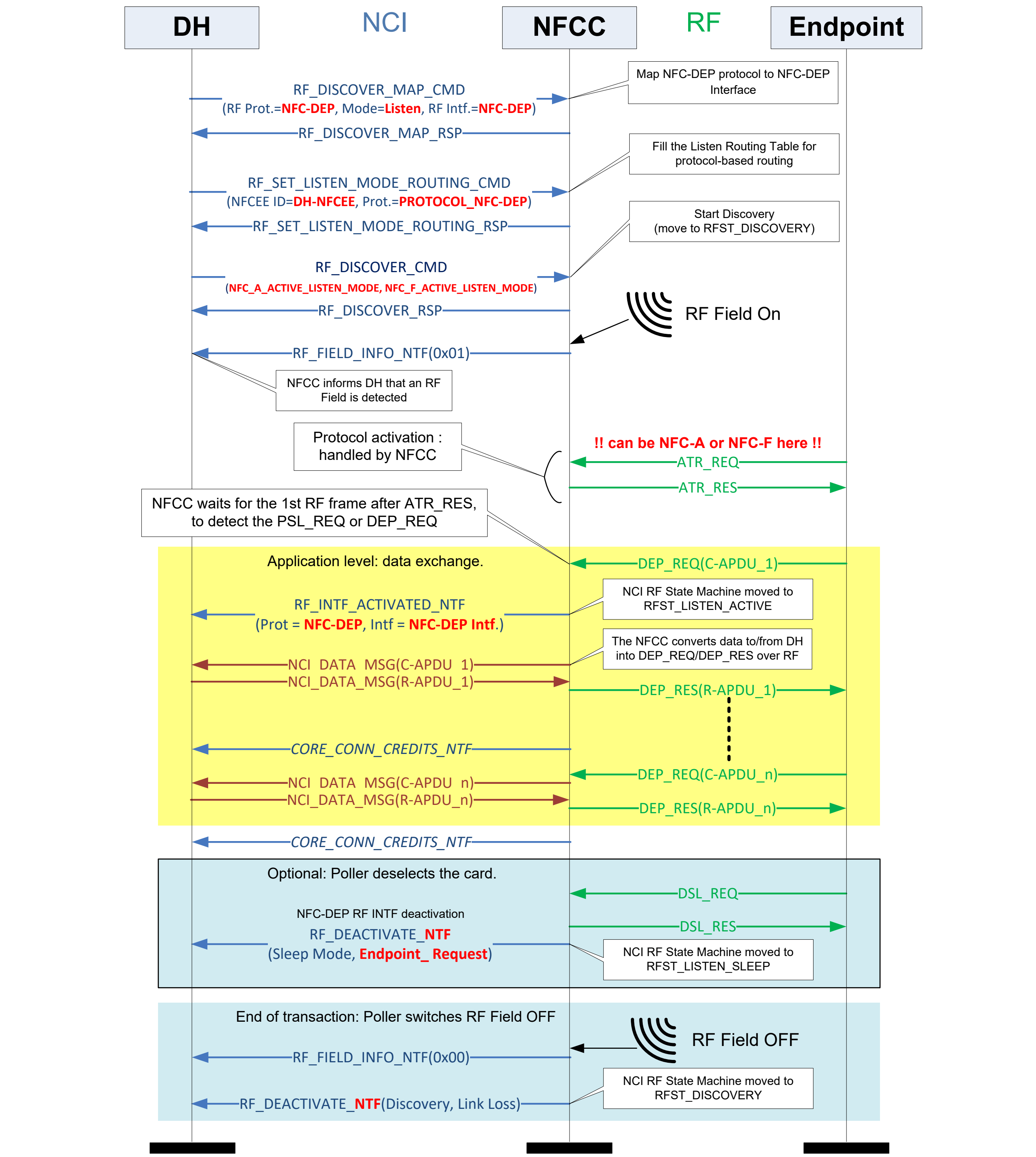

Fig 54. P2P/Target, NFC- A or NFC-F Active, NFC-DEP protocol / NFC-DEP Interface, NO PSL

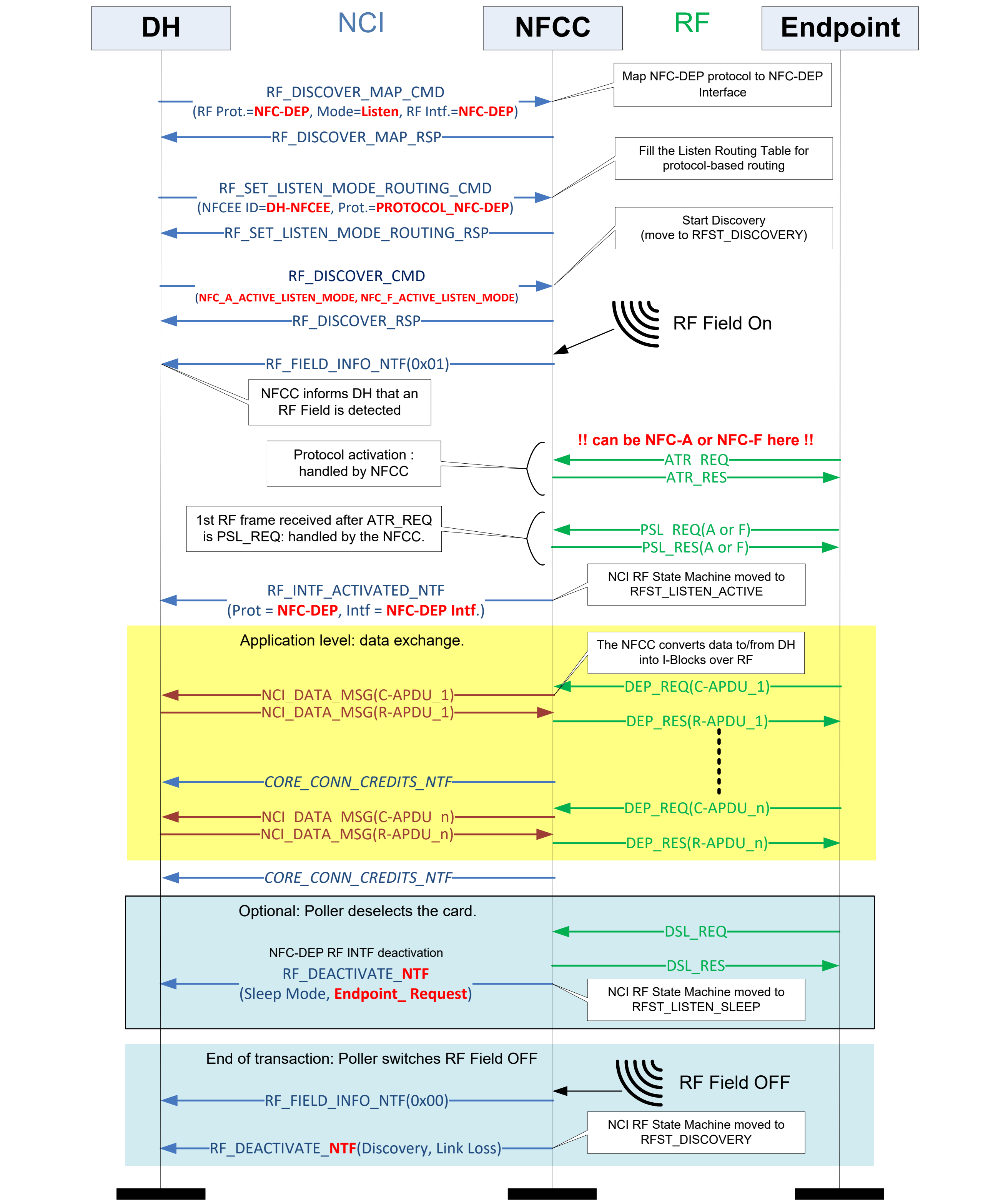

Fig 55. P2P/Target in DH, NFC- A or NFC-F Active, NFC-DEP protocol / NFC-DEP Interface + PSL

12.4.2 Initiator in P2P Active Mode

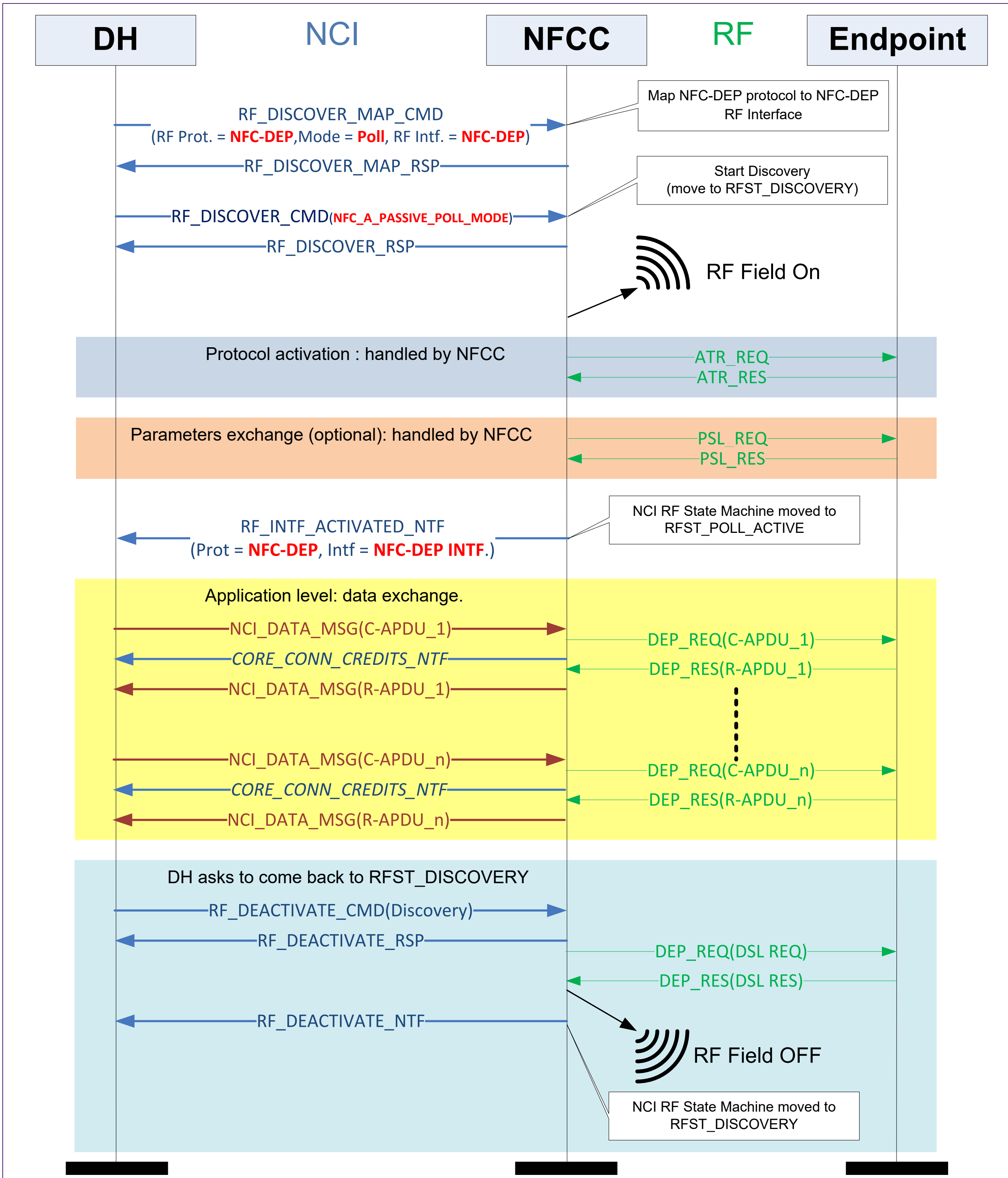

Fig 56. P2P/Initiator in DH, NFC-A Active, NFC-DEP protocol / NFC-DEP RF Interface

13. Annex A: details on RF state machine

The [NCI] RF State Machine is quite complex and the drawing proposed in the NCI technical Specification is combining all the different modes of operation in a single drawing.

For debug purposes, it is convenient to draw this State Machine in a simplified way, depending on the Protocol to RF Interface mapping applied by the DH. This is why the following figures are proposed here:

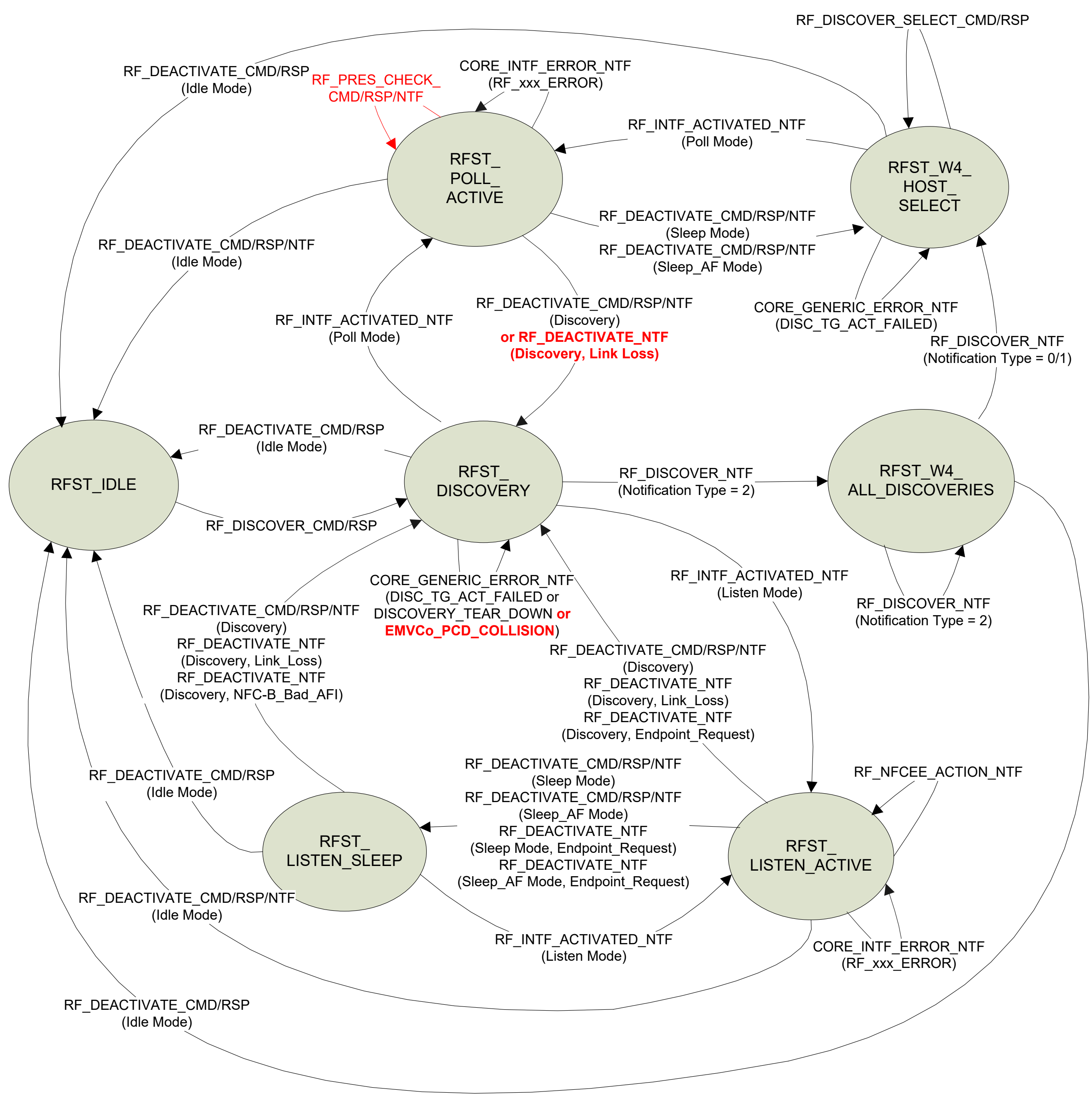

Fig 57. RF State Machine when the Frame RF interface is used

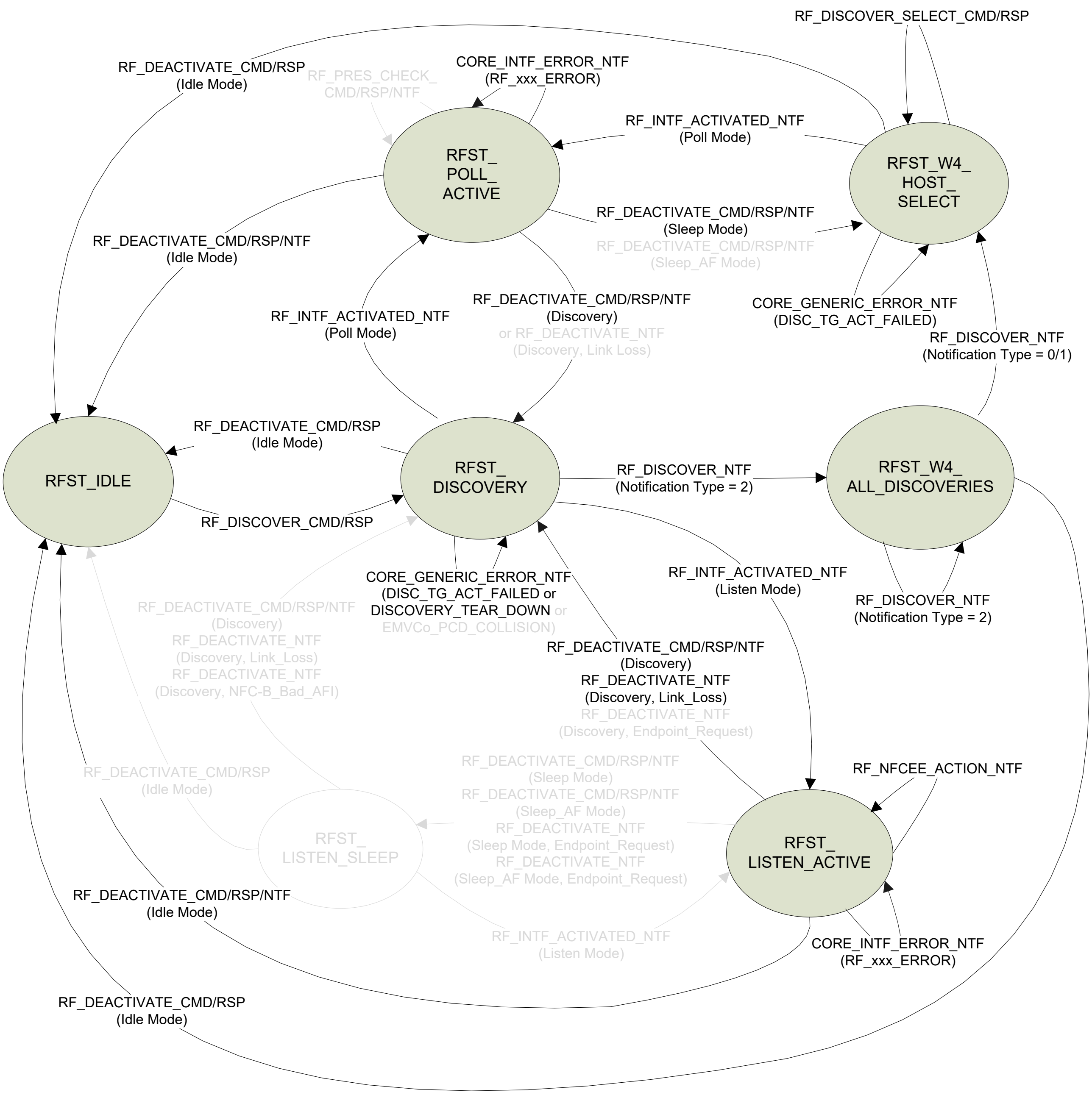

Fig 58. RF State Machine when the DH-NFCEE Direct RF interface is used

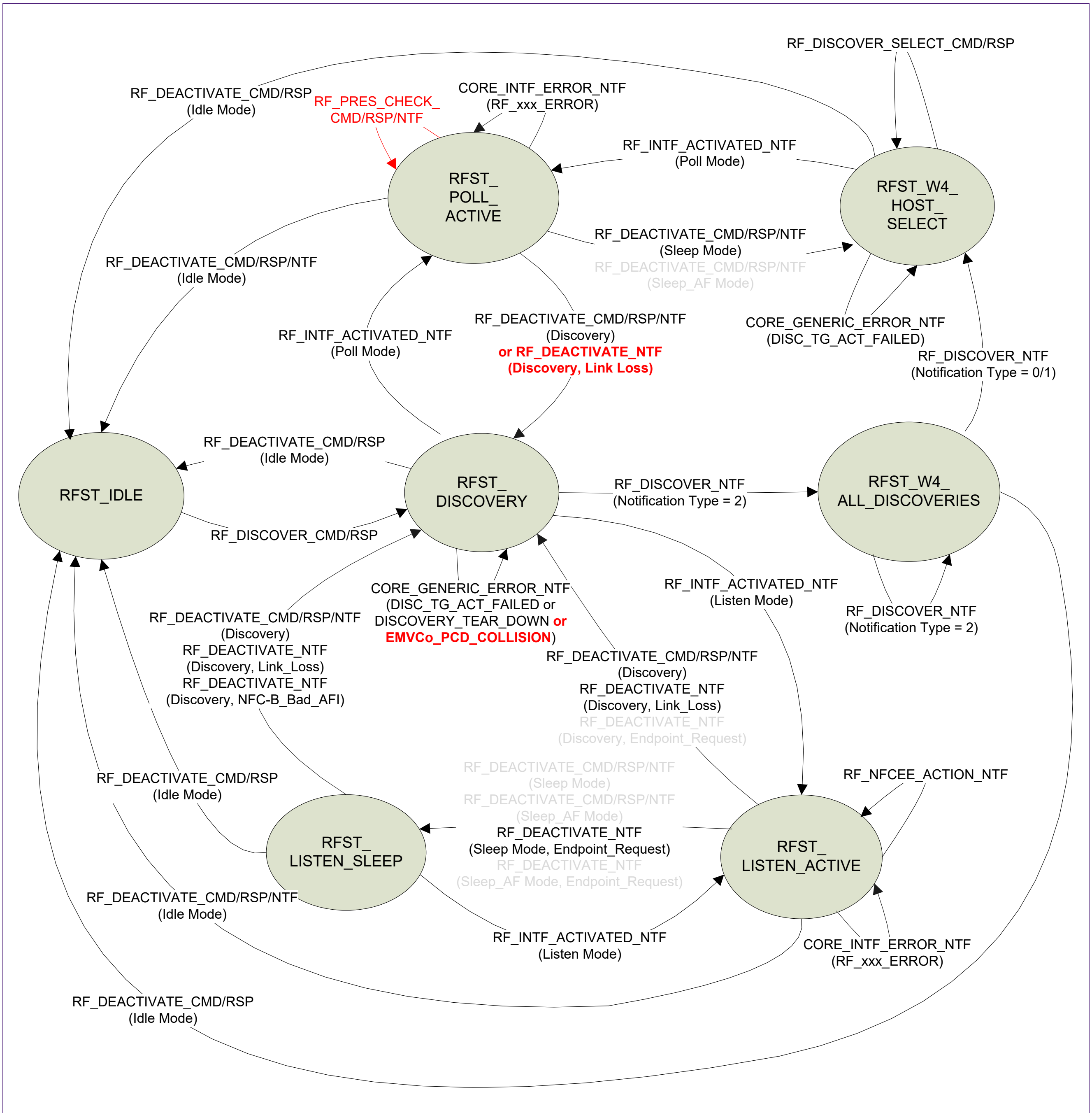

Fig 59. RF State Machine when the ISO-DEP RF interface is used

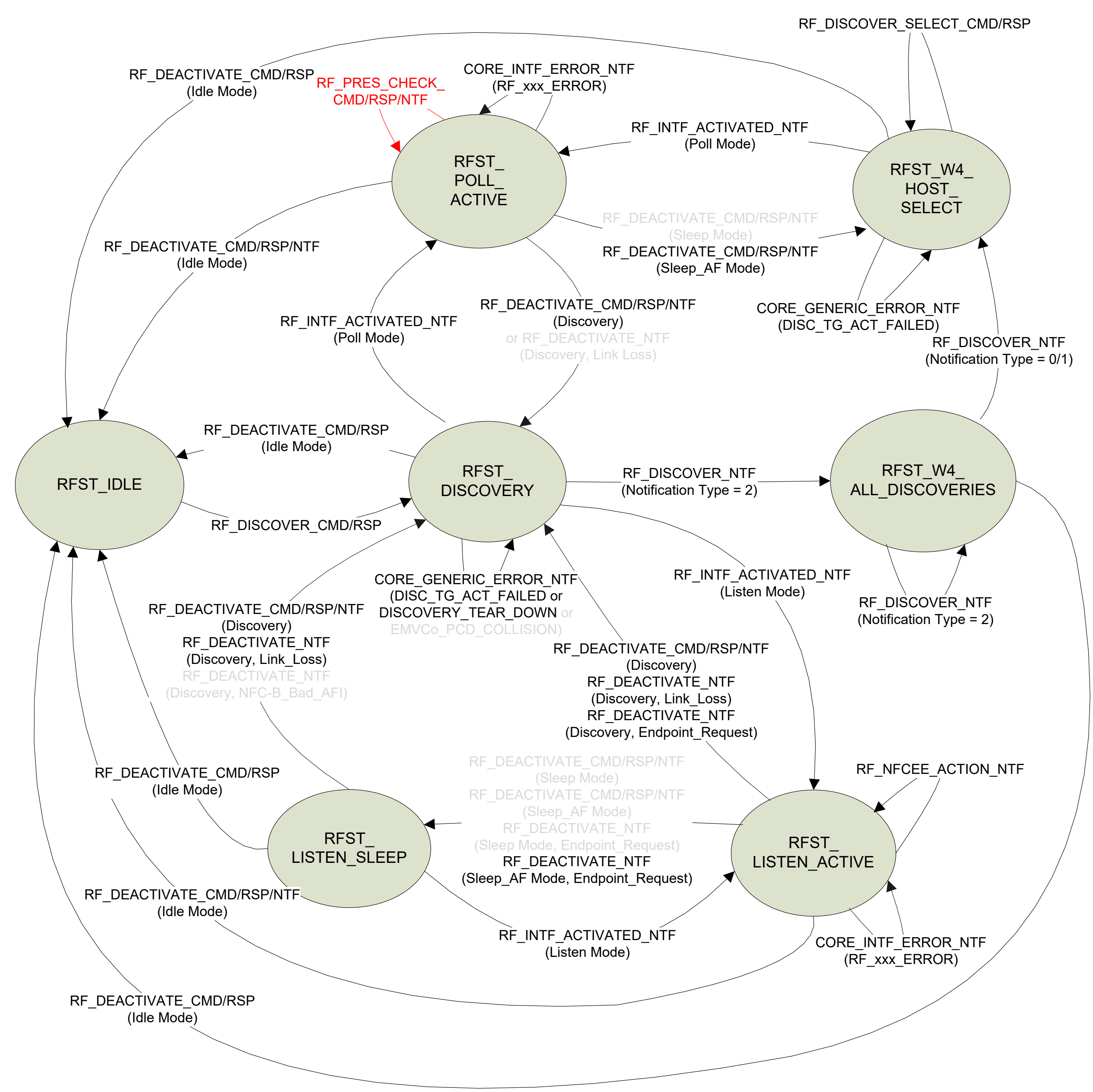

Fig 60. RF State Machine when the NFC-DEP RF interface is used

14. Abbreviations

Table 104. Abbreviations

Acronym	Description
CE	Card Emulation
DH	Device Host
DH-NFCEE	NFC Execution Environment running on the DH
ISO-DEP	ISO-DEP protocol as defined in [DIGITAL]
Listen mode	Listen mode as defined in [Digital]
NCI	NFC Controller Interface
NFC	Near Field Communication
NFC-A	NFC-A technology as defined in [DIGITAL]
NFC-B	NFC-B technology as defined in [DIGITAL]
NFCC	NFC Controller, unless mentioned this is the PN7150
NFC-DEP	NFC-DEP protocol as defined in [DIGITAL]
NFCEE	NFC Execution Environment
NFC-F	NFC-F technology as defined in [DIGITAL]
P2P	Peer To Peer
PCD	Proximity Coupling Device
Peer device	Device which can communicate via P2P mode as defined in [NFC-IP1]
PICC	Proximity Integrated Circuit Card
Poll mode	Poll mode as defined in [Digital]
R/W	Reader/Writer
RF	Radio Frequency
RFU	Reserved For Future Use

15. References

Table 105. References

[14443-4]	ISO/IEC14443-4
[7816-4]	ISO/IEC7816-4
[ACTIVITY]	NFC FORUM Activity Specification 1.0
[AN11755]	PN7150 Antenna and Tuning Design Guide
[AN11756]	PN7150 Hardware Design Guide
[AN11757]	PN7150 Low Power Mode Configuration
[DIGITAL]	NFC FORUM Digital Protocol Specification 1.0
[I²C]	I²C -bus specification and user manual Rev 03, defined by NXP. Last revision from April 2014 can be found here: http://www.nxp.com/documents/user_manual/UM10204.pdf
[NCI]	NFC Controller Interface, version 1.0
[NCI_Chap1]	Discovery and Interface Activation: chapter 8.3.2.2 in [NCI]
[NCI_Chap2]	State Machine: chapter 5.2 in [NCI]
[NCI_Table1]	Status Codes table: table 106 in [NCI]
[NCI_Table2]	RF technologies table: table 107 in [NCI]
[NCI_Table3]	RF Technology & Mode table: table 108 in [NCI]
[NCI_Table4]	Bit Rates table: table 109 in [NCI]
[NCI_Table5]	RF protocols table: table 110 in [NCI]
[NCI_Table6]	RF Interfaces table: table 111 in [NCI]
[NCI_Table8]	Config. parameters table: table 113 in [NCI]
[NCI_Table9]	CORE_RESET_NTF table: table 5 in [NCI]
[NFC-IP1]	ISO/IEC 18092
[PN7150_DS]	PN7150 Datasheet

16. Legal information

16.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

16.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Security — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

16.3 Licenses

Purchase of NXP ICs with NFC technology
Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

16.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

MIFARE — is a trademark of NXP B.V.

MIFARE Classic — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V.

MIFARE Plus — is a trademark of NXP B.V.

MIFARE Ultralight — is a trademark of NXP B.V.

NXP — wordmark and logo are trademarks of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

17. List of figures

18. List of tables

19. Contents

Date of release: 6 November 2020
348120

Document identifier: UM10936

Date of release: 6 November 2020
348120

Document identifier: UM10936