UM191812

PN544 C2 User Manual

Rev. 1.2 — 2010-06-16 User Manual

Document information

Info	Content
Keywords	PN544 C2, Logical Link Control Protocol, Host controller interface, Single Wire Protocol, NFC 2nd generation, SIM centered solution, Powered by the Field
Abstract	This is a user manual for the PN544 C2 NFC IC. The aim of the document is to describe the PN544 Firmware API enabling you to design your NFC system.

Revision history

Rev	Date	Description
1.0	2010-04-28	Release for PN544 C2
1.1	2010-05-07	Add register: TX_Current_Check, 0x9F14 change default value, *NXP_EVT_NFC_DEACTIVATED* description update, Add NAD usage NfcT (*0x98A3),* 0x997A, 0x9F19 default value update, registers 0x998 and 0x9931 removed (trimmed value – do not change), Fig 47 update
1.2	2010-06-16	Updates: 7.4.5 Error Detection and Error Handling, 49 Information Event, 9.8 MIFARE PCD, 9.4.5 CLK request in NFC active target mode, 9.12.1 Initiator: added maximum frame length warning, typos, added list of tables and figures, Fig 7 HCI packet bigger than maximum packet size, added EEPROM configurations SWP_Act_Retry, IFSLEW, Table 133 HW configuration in EEPROM, Table 13 HCI Gate Loopback, Type B PCD Anti-collision, Type B PICC HIGHER_LAYER_RESPONSE, RF Configuration in Polling Loop, Host Link Timing restriction

Contact information

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

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

All information provided in this document is subject to legal disclaimers.

1. Introduction

The PN544 is a highly integrated transmission module for contact-less communication at 13.56 MHz. The User Manual describes the software interfaces (API). As this document assumes pre-knowledge on certain technologies please check section “4 References”.

For further information please refer to the PN544 data sheet [7].

2. PN544 C2

This document is related to the PN544 C2, it gives an overview of the Software API available.

Please refer to the ‘PN544 Release Note’ [10] for further information on PN544 samples functionality level.

3. Abbreviations

Table 1. Abbreviations

Abbr.	*Abbreviation*
API	*Application Program Interface*
CLF	*Contact less front-end*
CLT	*Contact less Tunnel*
CRC	*Cyclic Redundancy Check (according to SWP specification)*
DSR	*Control Line on RS232: Data Set Ready*
DTR	*Control Line on RS232: Data Terminal Ready*
EE	*EEPROM (non volatile memory)*
HCI	*Host Controller Interface*
HCP	*Host Controller Protocol*
HDLC	*High Level Data Link Control*
HSU	*High speed UART*
HW	*Hardware*
I	*Information frame (HDLC)*
I2C	*Serial Data bus by Philips*
LLC	*Logical Link Control*
MAC	*Media Access Control*
NFC-WI	*Near Field Communication – Wired Interface*
PBTF	*Power By The Field*
PRBS	*Pseudo Random Binary Sequence*
PVDD	*Pad VDD (Supply)*

Abbr.	*Abbreviation*
RAM	*Random Access Module (volatile memory)*
Reg	*Register*
REJ	*Reject (HDLC)*
RF	*Radio Frequency (here 13,56MHz)*
RFU	*Reserved for future use*
RI	*Control Line on RS232: Ring Indicator*
RNR	*Receiver Not Ready (HDLC)*
RO	*Read only*
RR	*Receiver Ready (HDLC)*
RSET	*Reset (ETSI)*
RW	*Read / write*
SHDLC	*Simplified HDLC*
SPI	*Serial Peripheral Interface*
SREJ	*Selective Reject (HDLC)*
SWP	*Single Wire Protocol*
UART	*Universal Asynchronous Receiver Transmitter*
UI	*Unnumbered Information (HDLC)*
UICC	*Universal Integrated Circuit Card*

4. References

[1] Standard ETSI (ETSI TS 102 622 V7.5.0) Smart Cards; UICC-CLF Interface; Host Controller Interface – Release 7 (June 2009)

[2] Standard ETSI (ETSI TS 102 613 V7.7.0) Smart Cards; UICC-CLF interface; Physical and data link layer characteristics – Release 7 (October 2009)

[3] Standard ECMA-373 Near Field Communication Wired Interface (NFC-WI) http://www.ecma-international.org/

[4] Standard ISO18092 Near Field Communication -- Interface and Protocol (NFCIP-1) http://www.iso.org

[5] NXP Mifare Portfolio www.mifare.net

[6] ISO 14443 part 1 to part 4 Identification cards – Contact-less integrated circuit(s) cards -- Proximity cards http://www.iso.org

[7] PN544 C2 Datasheet – Doc Number 1878xx

[8] Application Note “PN544 Software update” – 1581xx

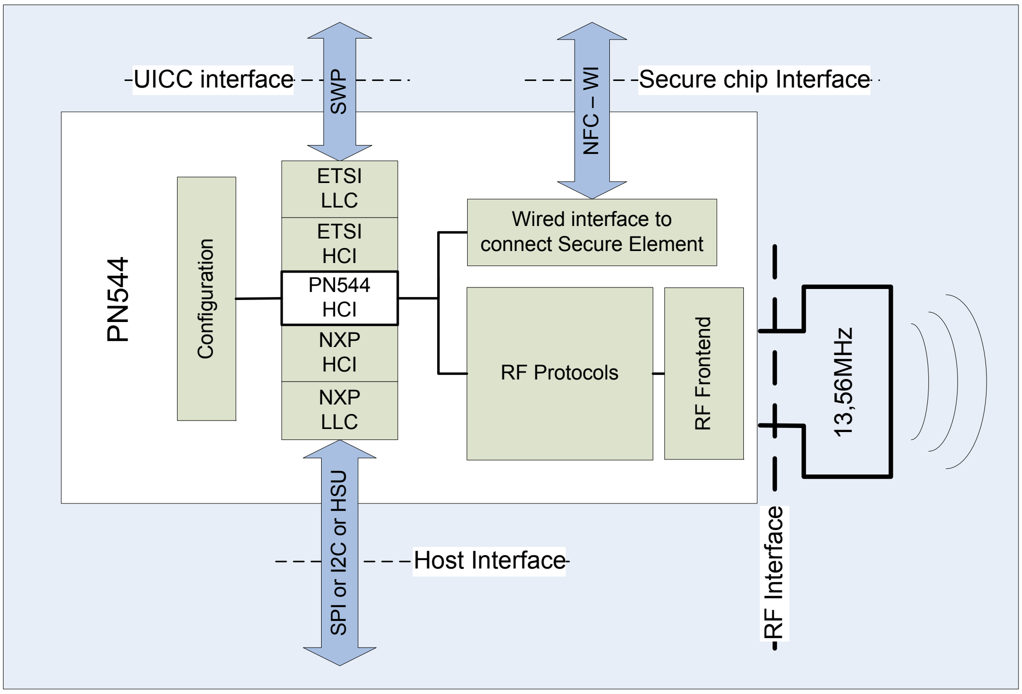
[9] PN544 Custom Clock settings (Excel Worksheet Formula) – Doc Number 1610xx

[10] PN544 C2 Release Note – Doc Number 1882xx

[11] PN544 RF Configuration – Doc Number 1659xx

5. PN544 Software Architecture

5.1 System Overview



(1) A high level view on the PN544 interface / software structure

Fig 1. System Overview

The PN544 offered physical interfaces and their functionalities are

- Host Interfaces
  consisting of I^{2}C, SPI and HSU as physical interfaces
- RF Interface
  This interface features 13,56MHz based protocols such as NFCIP-1 [4], MIFARE[1] [5] and ISO14443 [6]
- SWP Interface
  This interface is according to the ETSI [2] specification
- NFC-WI
  This interface is according to ECMA 373 [3]

[1] Reader modes as well as card emulation modes

All information provided in this document is subject to legal disclaimers.

From software point of view we refer to “PN544 HCI”. Depending on the interface used the ETSI or the NXP proprietary one is used. For ease of nomenclature we refer to HCI meaning NXP PN544 HCI.

HCI is used for host interfaces and UICC interface (See [1]).

The following chapters describe how to access PN544 over HCI and how to configure it. Main parts cover the PN544 NXP HCI specifics.

6. Host hardware interface configuration

6.1 General points

The system host controller can communicate with the PN544 by using the SPI, I^{2}C or HSU (High Speed UART) serial links.

Only one link can be used at once, and the choice is done by a hardware configuration (interface mode lines IFSEL [2:0]) during the power up sequence of the chip.

Note: The host interface check is performed by the firmware at PN544 power-up (PVDD available) and after any Reset (Hardware or Software). See ‘Boot Sequence’ chapter of PN544 Datasheet [7] .

Table 2. Host Interface Selection
Here is the description of the physical connections on pins IFSEL [2:0] (define the interface) and IF [3:0] (physical host interface)

Interface Selection Pin Name	Interface Pin Name				Link
IFSEL[2:0]	**IF0**	**IF1**	**IF2**	**IF3**	
000	nc	RX	nc	TX	**RS232**
001	-	-	-	-	**rfu**
010	ADDR0	ADDR1	SDA	SCL	**I^{2}C**
011	-	-	-	-	**rfu**
100	NSS	MOSI	SCK	MISO	**SPI** **(CPOL=0, CPHA=0)** [(1)]
101	NSS	MOSI	SCK	MISO	**SPI** **(CPOL=0, CPHA=1)** [(1)]
110	NSS	MOSI	SCK	MISO	**SPI** **(CPOL=1, CPHA=0)** [(1)]
111	NSS	MOSI	SCK	MISO	**SPI** **(CPOL=1, CPHA=1)** [(1)]

[1] CPOL and CPHA are part of SPI Configuration. For more information about these 2 parameters, please refer to following table:

Table 3. CPOL / CPHA description

Bit name	description
CPOL	Clock polarity: selects the polarity of the shift clock 1: shift clock is active low. 0: shift clock is active high.
CPHA	Clock phase: This bit indicates, at which edge of the clock signal the data will be sampled. The other edge initiates the value change on the data line 1: As soon as input pin NSS goes low, the transaction begins and the edge of SCK invokes the first data sample. Sampling of the data happens at the even edges (2, 4…16) of the SCK clock. 0: if input pin NSS goes low, the outputs are enabled. Sampling of the data happens at the odd edges (1, 3…15) of the SCK clock.

Table 4. Pin functional description

Here is the functional description of each pins used by all interfaces

Pin Name	Dir	Description
RX	I	Asynchronous UART receiver input, which is similar to an RS232 protocol on PVDD level.
TX	O	Asynchronous UART transmitter output which is similar to an RS232 protocol on PVDD level.
ADDR0	I	I^{2}C slave address bit 0 (LSB)
ADDR1	I	I^{2}C slave address bit 1
SDA	IO	IO pin SDA
SCL	I	Clock Input pin SCL
NSS	I	Slave select active low for SPI slave mode
MOSI	I	Master Out Slave In Data input pin in slave operation
SCK	I	Clock input pin in slave operation
MISO	O	Master In Slave Out Data output pin in slave operation

6.2 SPI interface

The PN544 is a slave SPI.., The mode used for the clock can be chosen for the phase (CPHA) and the polarity (CPOL). The pin **IFSEL0** is used to set **CPHA** and **IFSEL1** is used to set **CPOL**.

(See Table 3 for **CPHA** & **CPOL** settings).

How to configure SPI interface:

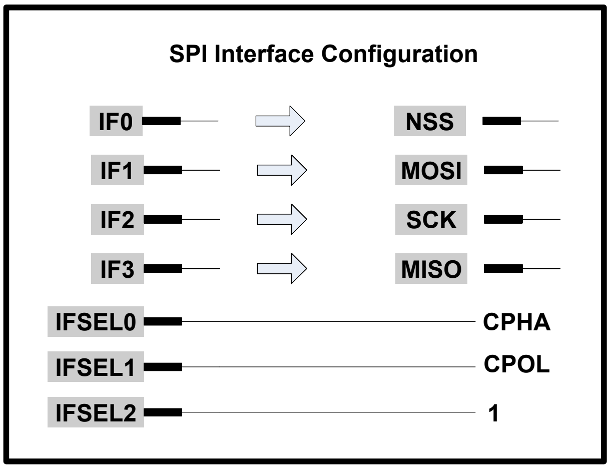


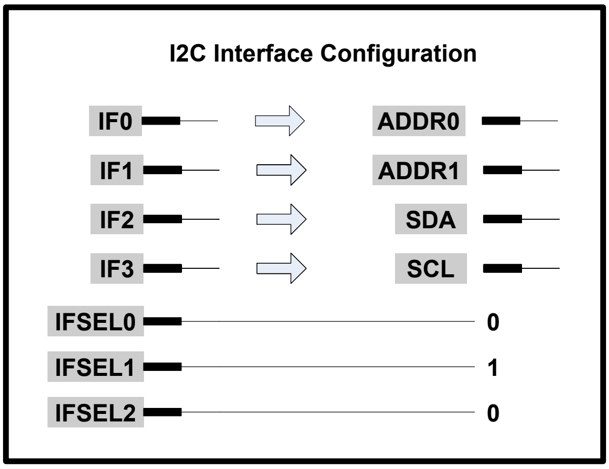
Fig 2. SPI interface configuration

6.3 I^{2}C interface

The PN544 is an I^{2}C slave.

To use this interface, an I^{2}C address has to be set. The default address of the device is set to binary “0 1 0 1 0 IF1 IF0”. The prefix “0 1 0 1 0” is a fixed value inside PN544.

How to configure I^{2}C interface:



2)

Fig 3. I2C interface configuration

6.4 HSU interface

HSU interface default configuration is:

Data bit	: 8 bits
Parity bit	: none
Stop bit	: 1 bit
Baud rate	: 115200 bauds
Data order	: LSB first

7. NXP Logical Link Layer

PN544 offers four interfaces using LLC on top of the physical layer. The logical link layer of the SWP interface is according to the SWP specification (see [2]). The other interfaces (i.e. I^2C, SPI, and HSU) are derived from the SWP LLC layer to fit the needs of the respective interface. We refer to the “NXP LLC” layer described in Fig 1 ‘System Overview’.

1. Host Interfaces
    a. I^2C
    b. SPI
    c. HSU (i.e. High Speed UART)
2. ETSI SWP Interface

7.1 Overview on physical interfaces

Table 5. Physical host Interfaces Overview

	I^2C	SPI	HSU	SWP
Communication Flow	Half Duplex	Full Duplex	Full Duplex	Full Duplex
Clock Generator	Host Only	Host Only	Host and Device	Device (CLF) Only
Speed	Enforced by host clock	Enforced by host clock	Specified by the host	Configurable by the host.
Frame Start/ End Condition	Guard times[2]	Guard times	Guard times	As defined in SWP (SOF, EOF)
Activation	Host	Host	Host	Device (CLF)
Bus Capabilities	Yes	Yes	No	No
Used Lines	SDA, SCL, IRQ	MOSI, MISO, NSS, SCK, IRQ	RX, TX	SWIO, VCC

The supported speeds of the interfaces are found in the PN544 Datasheet [7].

7.2 Link Layer Features

The link layer guarantees reliable data transfer and a balanced link over an unreliable MAC layer. In order to guarantee reliability, the LLC layer is able to detect and able to recover from erroneous situations.

The layer above (i.e. the HCI) can assume that all frames are delivered without errors and in a sequential order.

7.3 LLC on SWP

The product will support all mandatory features as specified in the ETSI SWP specification [2]. The optional features / parameters are as follows:

Default window size is 4. The UICC may negotiate the window size down to 2.

SREJ is not supported

For more information, please refer to ETSI SWP specification [2].

[2] Configurable, typically between 1ms and 100ms. . Currently disabled.

7.4 LLC on I^{2}C, SPI, UART

The link layer of the host protocol is derived from the ETSI SWP specification. The deviations from the ETSI LLC layer are as follows:

- CLT protocol is not supported
- ACT protocol and SYNC_ID handling are not supported

HSU baudrate can be either configured by EEPROM or by an additional argument in RSET from the host interface

- The host is supposed to start with the activation of the link layer
  Note: This means that the host will start with the RSET frame.

The optional features / parameters are as follows:

Window size is 4, can be negotiated down to 2.

SREJ is not supported

Retry rate is set to 10

Inter-frame character timeout (See 7.4.5)

WARNING: A minimum delay of **1.1ms** between to consecutive frames sent to PN544 is required. Thus, Host Link driver must ensure to respect this timing when sending U-Frame, I-Frame & S-Frame.

7.4.1 Frame definition

The frame format is the same for all host interfaces. It is derived from the LLC frame as specified in the SWP specification. The major difference between host and SWP frame is that the host interface frames contain the length information. Otherwise, things like the LLC command set (I, RR, RNR, REJ, RSET), frame length and LLC handshaking is identical. Bit stuffing is not used by the host interfaces.

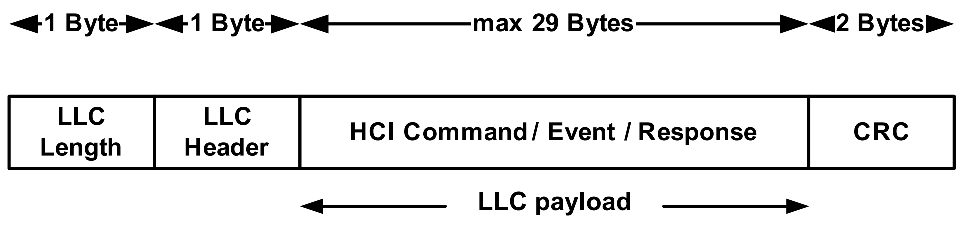


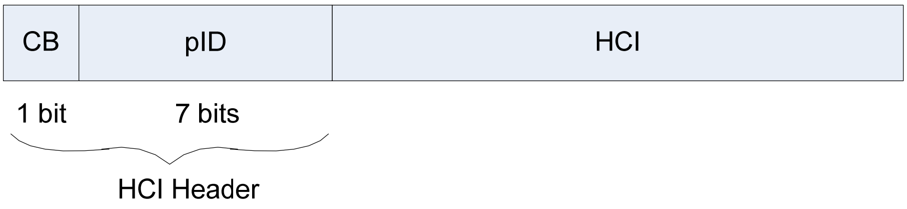
Fig 4. LLC Frame Components

The LLC Length specifies the length of the frame to follow. The length itself is not counted. The two CRC bytes are counted. The LLC frame is represented as follows:

- LLC Length is mandatory (valid range: 03h – 20h)
- LLC Header is mandatory.
- LLC Payload is optional. This is depending on the LLC command, which is contained in the LLC Header.
- CRC is always present
  the bit and byte order is Most Significant Bit / Byte first.

7.4.2 Chaining description

For HCI command, the format of a HCP packet is defined in Fig 5:



(1) CB is the chaining bit and its value is equal to 1 except when message fragmentation is used

(2) P_{ID} specifies the pipe identifier.

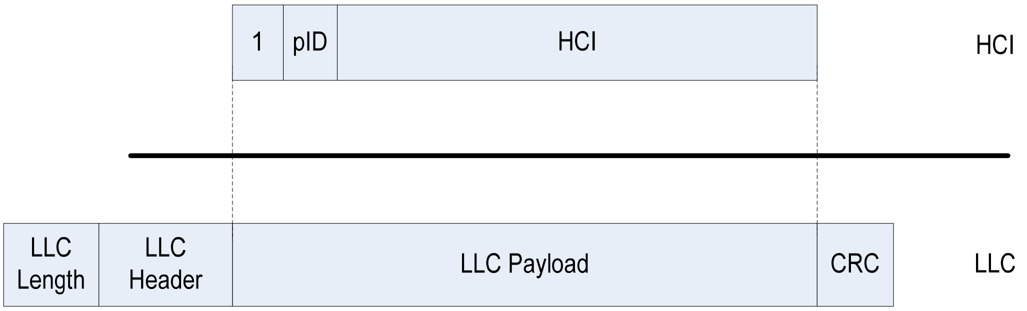
Fig 5. HCP packet

The LLC layer does not provide any chaining, segmentation or reassembly feature. Therefore the HCI has to obey the maximum size rules as defined in this document.

If HCI determines that a certain HCI command does not fit into the negotiated packet size, HCI needs to split up the big packet into smaller chunks by using the chaining feature as specified in HCI. The LLC is and shall not be aware of this chaining functionality. LLC does not care if a packet is split up into one or more LLC frames.

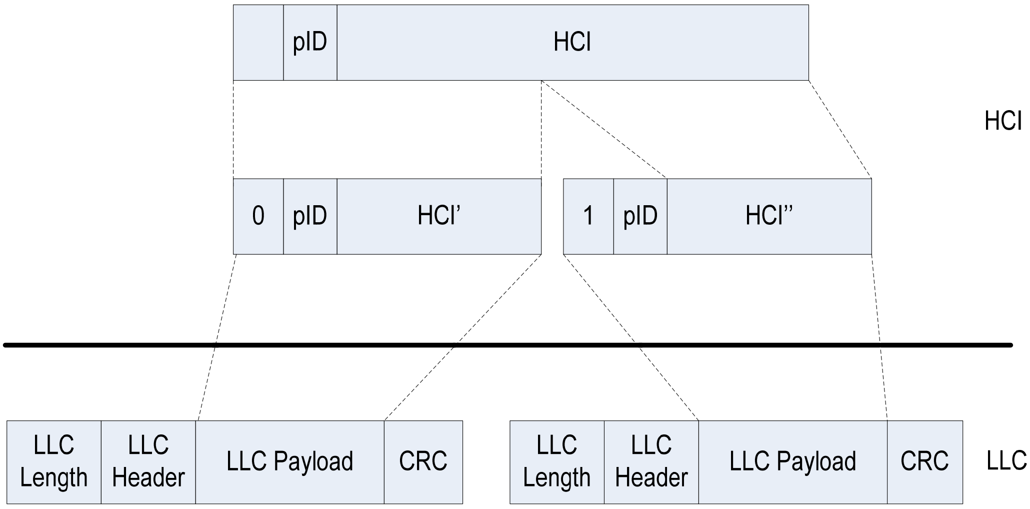
The length of the LLC frame is aligned with the length of the HCI packet. There is no padding at the end of the HCI packet. When HCI is sent over LLC, the beginning of an LLC frame implies the beginning of an HCI packet (without considering the LLC header), the end of an LLC frame implies the end of an HCI packet (without considering the CRC).

Examples of possible usage of CB bit:



(1) HCI single frame without chaining

Fig 6. HCI packet smaller than maximum packet size



(1) The HCI component splits up the content into two different LLC frames. Therefore HCI must perform two "LLC Send ()" commands. From the LLC point of view it looks like two regular I frames. Therefore the LLC will generate two acknowledges.

Fig 7. HCI packet bigger than maximum packet size

7.4.3 LLC Header description

The LLC Header (called also SHDLC control field) is part of the LLC frame structure defined in the following figure:

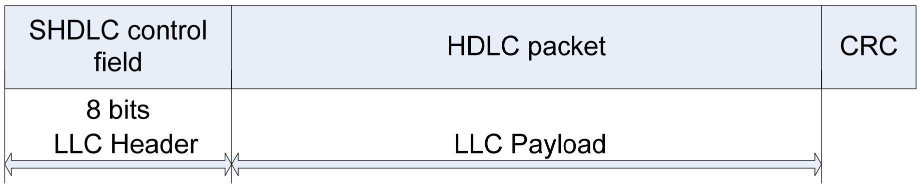


Fig 8. LLC layer (SHDLC)

7.4.3.1 SHDLC frame types

SHDLC uses several types in order to transfer data and to manage or supervise the communication channel between the two endpoints (ends of the communication channel):

I-Frames (Information frames): Carry upper-layer information and some control information. I-frame functions include sequencing, flow control, and error detection and recovery. I-frames carry send and receive sequence numbers.

S-Frames (Supervisory Frames): Carry control information. S-frame functions include requesting and suspending transmissions, reporting on status, and acknowledging the receipt of I-frames. S-frames carry only receive sequence numbers.

U-Frames (Unnumbered Frames): Carry control information. U-frame functions include link setup and disconnection, as well as error reporting. U-frames carry no sequence numbers.

7.4.3.2 Control field

The SHDLC control field has the structure described in following table.

Table 6. SHDLC Control field coding

Frame Types	Bit Field							
	8	7	6	5	4	3	2	1
I	1	0	N(S)			N(R)		
S	1	1	0	TYPE		N(R)		
U	1	1	1	M				

Where:

- N(S): Number of the information frame
- N(R): Number of next information frame to receive
- TYPE: Type of S-Frame
- M: Modifier bits for U-Frame

I-Frames coding

The functions of the information(**I**) command and response is to transfer sequentially numbered frames, each containing an information field, which might be empty, across the data link.

S-Frames coding

Supervisory(**S**) commands and responses are used to perform numbered supervisory functions such as acknowledgment, temporary suspension of information transfer, or error recovery. Frames with the S format control field do not contain an information field.

Supervisory Format commands and responses are as follows:

RR: Receive Ready is used by an endpoint to indicate that it is ready to receive an information frame and/or acknowledge previously received frames.

RNR: Receive Not Ready is used to indicate that an endpoint is not ready to receive any information frames or acknowledgments.

REJ: Reject is used to request the retransmission of frames.

SREJ: This optional command is not supported by the PN544. If received SREJ is treated like an erroneous frame.

The type coding is given by the following table:

Table 7. Type coding of the S-Frames

Frames	type	status
RR	00	Mandatory
REJ	01	Mandatory
RNR	10	Mandatory
SREJ	11	Not supported

Optional type of frame shall not be used before capability negotiation is defined during initialization.

U-Frames coding

The unnumbered (**U**) format commands and responses are used to extend the number of data link control functions. The unnumbered format frames have 5 modifier bits which allow for up to 32 additional commands and responses. Only a subset of the HDLC commands and responses are used for SHDLC:

RSET: Reset of the data link layer is used to reset the sequence number variables in the both endpoints.

UA: Unnumbered Acknowledgment is used to acknowledge the receipt and acceptance of a RSET command.

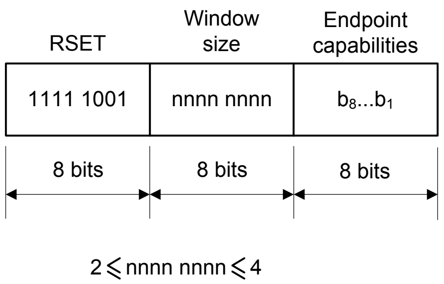
Table 8. modifier coding of the U-Frames

Frames	modifier	status
RSET	11001	Mandatory
UA	00110	Mandatory

The optional command UI is not supported.

7.4.3.3 RSET Frame

The RSET command is used to reset the link status. The behavior is the same as described in the respective SWP specification [2].



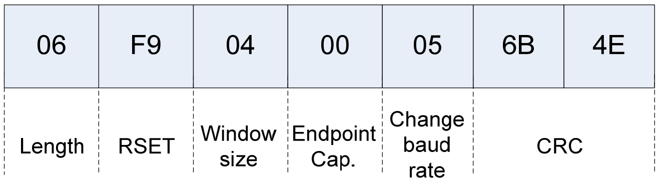
RSET	Window size	Endpoint capabilities
1111 1001	nnnn nnnn	$b_8 \ldots b_1$
8 bits	8 bits	8 bits

$2 \leqslant \text{nnnn nnnn} \leqslant 4$

(1) Window size = 4 for PN544

Fig 9. RSET Frame

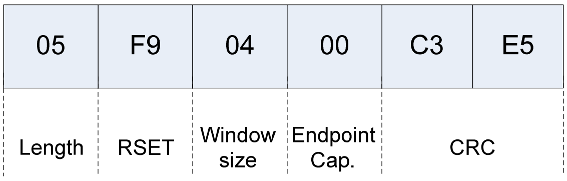
WARNING: Before initiating an HCI communication with PN544, at Power up, or after a Reset (VEN), it is **mandatory** to send a RSET frame. The recommended frame is defined in the following figure:



(1) U RSET frame (with 115200 baud rate for HSU)

Fig 10. LLC RSET frame example

Another example of complete LLC frame:



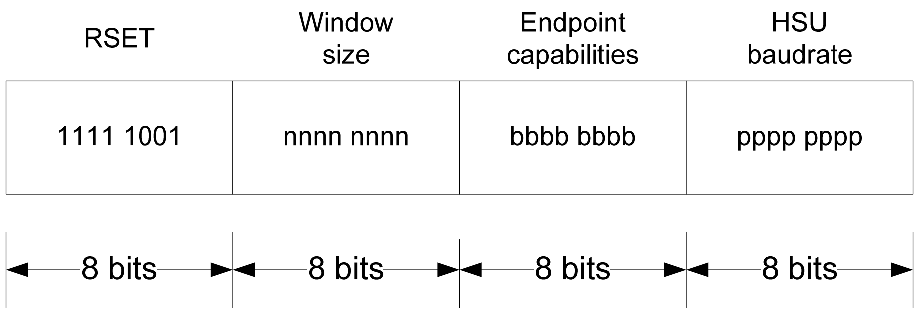
(1) U RSET frame (without HSU baud rate setup)

Fig 11. Recommended RSET frame

It is recommended to use the frame with no baud rate indicator if the baud rate is not changed. It prevents writing the baud rate to EEPROM and therefore saves time and energy.

7.4.3.4 Baud rate change (HSU)

Additionally to the window size and the endpoint capabilities another optional parameter is introduced on the Host interface. This parameter identifies the baud rate used for the HSU interface. The frame on the host side looks now as follows:



RSET	Window size	Endpoint capabilities	HSU baudrate
1111 1001	nnnn nnnn	bbbb bbbb	pppp pppp
8 bits	8 bits	8 bits	8 bits

(1) 2 ≤ nnnn nnnn ≤ 4

(2) 0 ≤ pppp pppp ≤ 9

(3) Endpoint capabilities are not used in host direction

Fig 12. Baud rate Frame

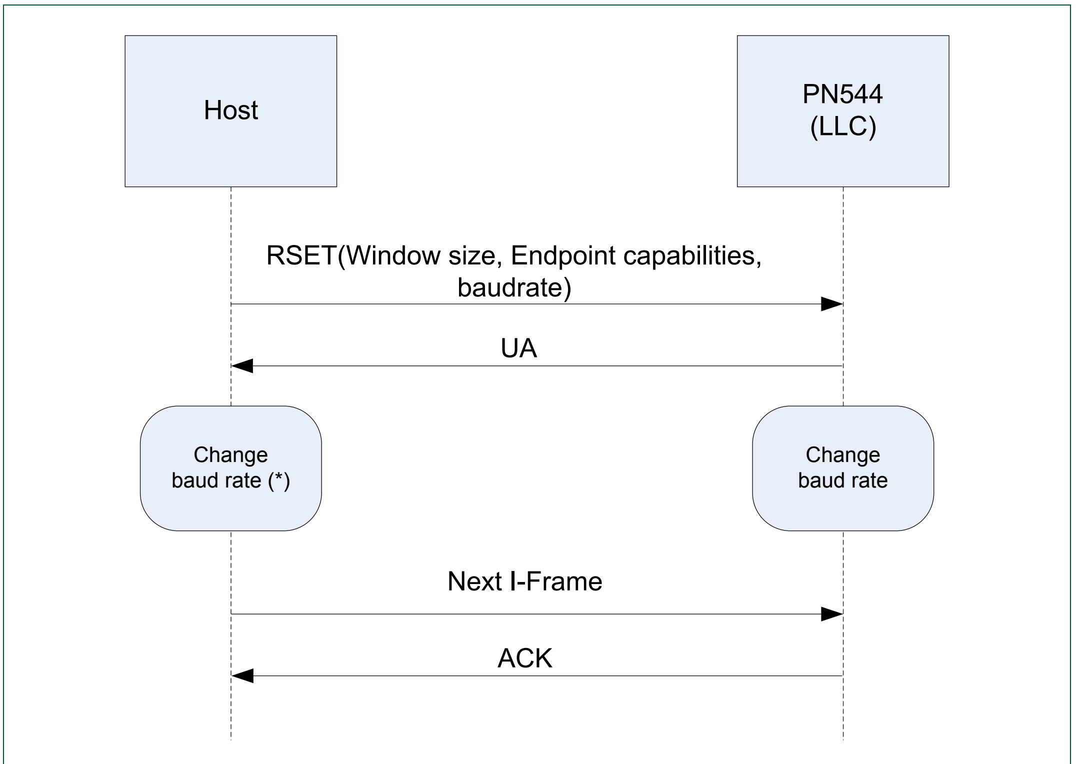
Note: It is highly recommended to only convey the HSU baud rate if the baud rate shall be changed. Otherwise the HSU baud rate field should not be sent in order to save power, processing time and EEPROM write cycles.

The baud rate is stored in non-volatile memory, which means that the set baud rate persists even after a reset or a power-down and power-up phase.

Byte coding of optional HSU baud rate:

0x00	9 600 baud
0x01	RFU
0x02	28 800 baud
0x03	RFU
0x04	RFU
0x05	115 200 baud
0x06	230 400 baud
0x07	460 800 baud
0x08	RFU

A baud rate change is always initiated by the host. The flow is as follows:



(4) (*) A 150µs delay is needed before sending the next I-Frame.

Fig 13. baud rate change flow

Example of change baud rate LLC frame

06	F9	04	00	00	C6	19
Length	RSET	Window size	Endpoint Cap.	Change baud rate	CRC	

(5) Change baud rate to 9600 (00h)

Fig 14. example of change baud rate LLC frame

7.4.4 CRC description

The detection of errors in a frame is based on the standard CRC-16 defined in [2]. The CRC polynomial is:

$$X^{16}+X^{12}+X^{5}+1.$$

Its initial value is 0xFFFF.

The CRC is computed on the bits between SOF and EOF both excluded. So in order to detect invalid frames the CRC shall be taken as a primary measure to check validity. The maximum size of a single frame is (LEN+HDR+PAYLOAD_MAX+CRC = 1 + 1 + 29 + 2) 33 Bytes.

7.4.4.1 Example

The CRC of an RSET and an UA is as follows:

05	F9	04	00	**C3**	**E5**
Length	RSET	Window size	Endpoint Cap.	CRC	

03	E6	**17**	**A7**
Length	UA	CRC	

Fig 15. CRC examples

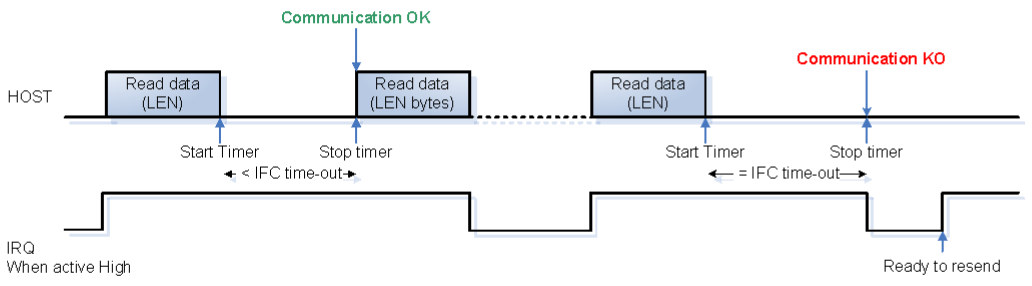
7.4.5 Error Detection and Error Handling

7.4.5.1 Inter-frame-character timeout

PN544 measures the time between characters within a frame. If the time exceeds the timer T_{IC}, PN544 considers the frame invalid. The error handling depends on the communication path:

The value of T_{IC} can be configured in EEPROM. If not needed, T_{IC} detection can be disabled. The T_{IC} is mainly used to trigger the communication between the host and the device. It is used with I2C and SPI host interface connections.

Note: To disable T_{IC} mechanism, the T_{IC} timeout has to be set to NULL value.



(1) T_{IC} = timer inter-frame character

Fig 16. Inter-frame-character definition

In case of Timeout on TX, PN544 will re-send the frame (IRQ is reset, and full frame is re-sent).

T_{IC} timeout value can be changed by setting the *Host_TIC_MSB_TX &* *Host_TIC_LSB_TX* EEPROM area (See

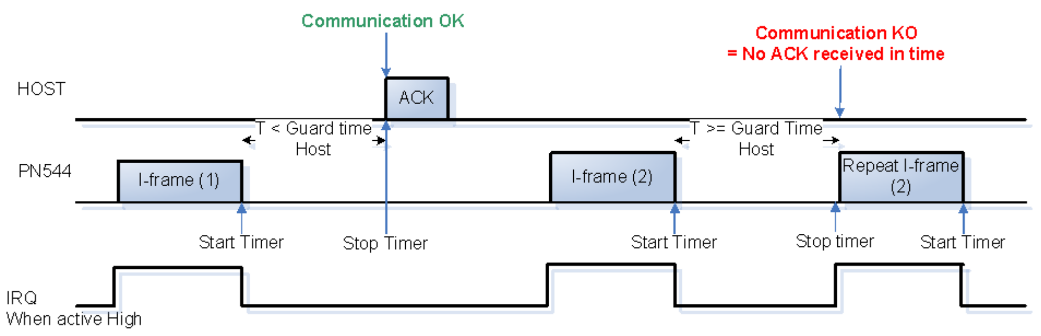
HW configuration' chapter)

The number of retries is defined in EEPROM as Host_Rx_Retry and Host_Tx_Retry.

7.4.5.2 Guard Host Timeout

This timeout defines the maximum time the Host has to acknowledge a frame received.

When timeout is reached, PN544 resends the frame. It is used with HSU interface.



(1) T: Guard Timeout Host

Fig 17. PN544 Guard Host Timeout

Case Communication OK: After sending a frame to the host PN544 starts an internal timer, waiting for the frame to be acknowledged by the host. The ACK is received in-time and PN544 will proceed sending the next frame.

Case Communication KO: Timer is started after frame was sent but no ACK is received in-time. -> Timeout triggers PN544 to resend the frame to the host.

The number of retries is defined in EEPROM as Host_Rx_Retry and Host_Tx_Retry.

Guard Host Timeout value can be changed by setting the *GuardHostTimeoutMSB & GuardHostTimeoutLSB* EEPROM area. (See

HW configuration chapter).

Note: To disable the resend mechanism, the timeout has to be set to NULL value.

7.4.6 HSU

All communication is performed via RX and TX. Data may be sent or received at any point in time and could overlap (full duplex). The interrupt pin will notify that is being sent. However, it is not mandatory to use that pin. Multiple baud rates are supported. Automatic baud rate detection is not supported. Flow control signals known from RS232 (such as DTR, DSR, RI, ...) are not supported. Data is conveyed in least significant bit, most significant byte first order.

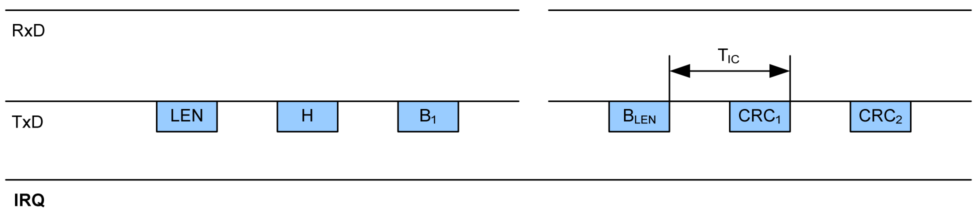
In order to detect invalid frames a CRC is conveyed along with the frame. The maximum size of a single frame is 33 Bytes. Considering the header offset 29 bytes remain for HCI. This is the same frame length as for the SWP interface if the LEN byte is not counted.

Note: A special attention is needed for HSU & System Clock. See **Clock Request & Release** chapter for details.

7.4.6.1 Example: Frame Transfer from Host to PN544

The host conveys the frame to PN544 by applying the complete byte stream on the TX line. No handshaking takes place between the characters. If the receiver (PN544) observed a problem, it will resort to error recovery and may request to re-transmit this frame.

Frame Transfer: Host -> PN544



(2) T_{ic} : inter-character timeout

Fig 18. Sample Transfer Host to PN544

7.4.6.2 Example: Frame Transfer from PN544 to Host

The PN544 conveys the frame to the host by applying the complete byte stream on the RX line. No handshaking takes place between the characters. If the receiver (host) observed a problem, it shall resort to error recovery and may request to retransmit this frame.

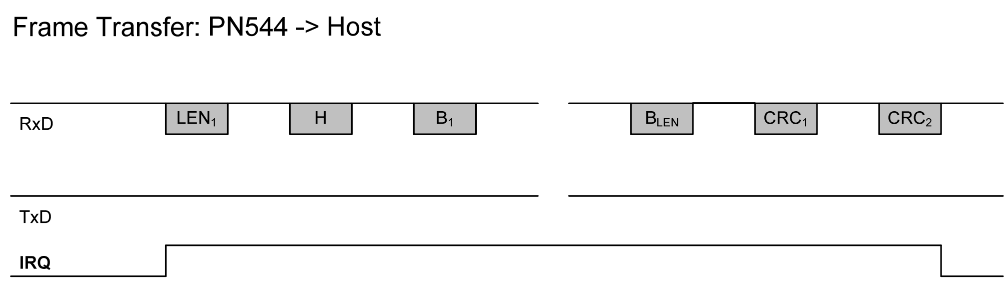


Fig 19. Sample transfer PN544 to Host

7.4.6.3 Example: Full Duplex Transfer

PN544 is capable of receiving and transmitting frames at the same time. The host shall be able to handle this situation as well. The frames shall be considered as completely independent.

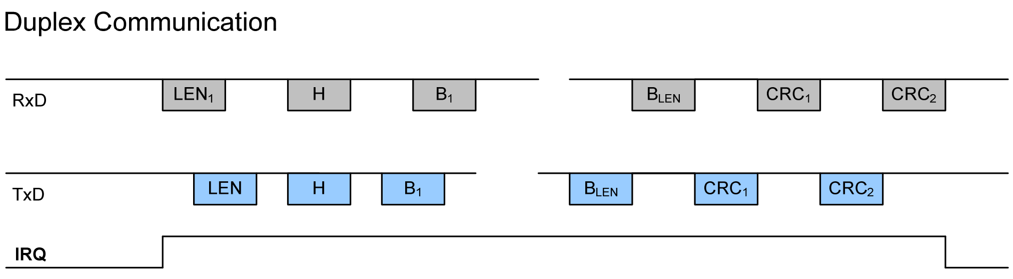


Fig 20. HSU Full Duplex Transfer

Note: At the beginning of a received frame, the IRQ is set to logic high, and data is sent **immediately**. IRQ line is set to logic low at the end of the received frame.

7.4.7 I2C

PN544 performs its I^{2}C communication over SDA and SCL. The bit rate is given by the master (= host). The default address of the device is set to binary “0 1 0 1 0 IF1 IF0” (the prefix “0 1 0 1 0” is a fixed value inside the IC, the suffix “IF0 IF1” are pins which need hardware configuration by schematic/ layout – see [7].

The PN544 software only supports the slave role. As specified in the I^{2}C specification, data is conveyed using most significant bit, most significant byte first.

Whenever PN544 wants to send data to the host, the IRQ pin is set to HIGH. This notifies the host that it should apply the clock, select the PN544 and set the direction bit to Read (from the host point of view). The IRQ management is not mandatory on host side; host can manage I^{2}C by polling as well.

Supported I^{2}C addresses:

(Address described on 8bits: 0 1 0 1 0 IF1 IF0 R/W, with the bit0 (R/W bit) set to 0)

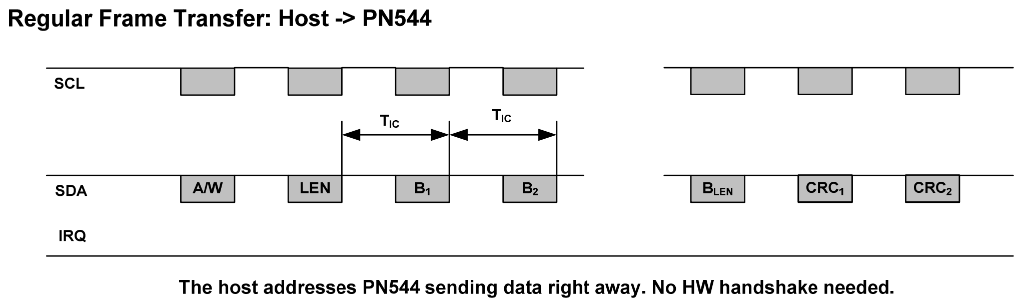
Address Value	**IF1 Pin**	**IF0 Pin**
50h	0	0
52h	0	1
54h	1	0
56h	1	1

7.4.7.1 Example: Communication from Master to Slave (Host to PN544)

The host will first select the PN544 by applying the slave address. Moreover, it will set the R/W bit to write. The PN544 I^{2}C state machine observes the address value and acknowledges if the address is matching the local address. The host can now start to convey the complete LLC frame.

If the LLC frame is longer than the length header byte claims to be, the length field shall be taken as a reference.

The host may perform multiple start/stop conditions within the frame. However, if the direction bit is toggled, the incomplete frame will be trashed by PN544 (as a T_{IC} timeout on RX path will happen).


Regular Frame Transfer: Host -> PN544
SCL
T_{IC}
T_{IC}
SDA
A/W
LEN
B_1
B_2
B_{LEN}
CRC_1
CRC_2
IRQ
The host addresses PN544 sending data right away. No HW handshake needed.

(3) T_{ic} : inter-character timeout

Fig 21. Sample I^{2}C Host to PN544 Communication

7.4.7.2 Example: Communication from Slave to Master (PN544 to Host)

The PN544 shall notify via the IRQ line that it wants to transmit data. The host shall now start to address the PN544 and shall set the direction to READ. The PN544 shall now convey the pending data to the host.

If the host performs a read request even if the chip does not have to send anything (i.e. the chip did not ask for a clock or the data turned out to be for some reason irrelevant), the chip shall transmit bytes containing the value (0xFF)h until the clock has been turned off.

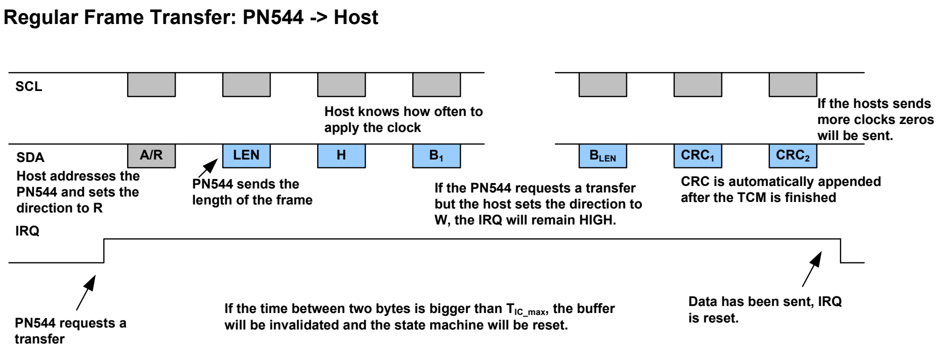

Regular Frame Transfer: PN544 -> Host
SCL
Host knows how often to apply the clock
If the hosts sends more clocks zeros will be sent.
SDA
A/R
LEN
H
B_1
B_{LEN}
CRC_1
CRC_2
Host addresses the PN544 and sets the direction to R
PN544 sends the length of the frame
If the PN544 requests a transfer but the host sets the direction to W, the IRQ will remain HIGH.
CRC is automatically appended after the TCM is finished
IRQ
PN544 requests a transfer
If the time between two bytes is bigger than T_{IC_max}, the buffer will be invalidated and the state machine will be reset.
Data has been sent, IRQ is reset.

Fig 22. I^{2}C PN544 to Host Regular Transfer

Note: At the end of a received frame, the IRQ is set to logic low and goes to high at the beginning of the following frame.

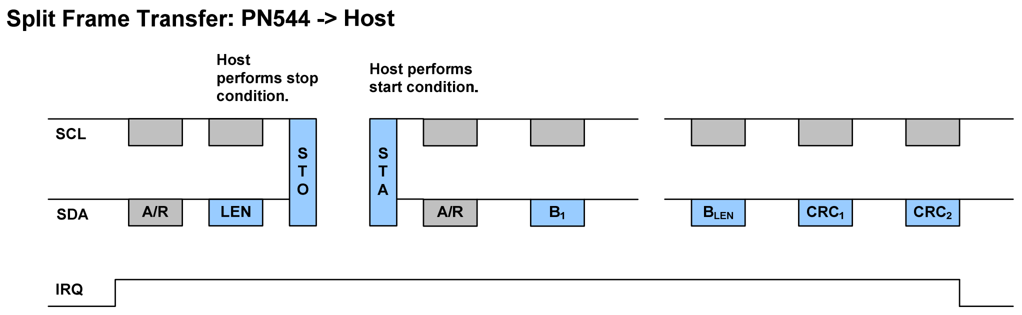

Split Frame Transfer: PN544 -> Host
Host performs stop condition.
Host performs start condition.
SCL
SDA
IRQ
STO
STA
A/R
LEN
A/R
B1
BLEN
CRC1
CRC2

Fig 23. I^{2}C PN544 to Host Split Transfer

7.4.8 SPI

The PN544 performs its communication by using the signals NSS, MISO, MOSI, SCK. PN544 only acts as slave; hence it only consumes but never drives the signals NSS, MOSI and SCK. Only MISO is driven as output. The bit rate is given by the host. Moreover an IRQ line needs to be connected which indicates pending data on the PN544. The bits are conveyed using most significant bit, most significant byte first. The interface can be used in full duplex mode.

These SPI relevant settings are read out during boot phase of the PN544. They are coded by a hardware pin configuration (Refer to ‘Host hardware interface configuration chapter’).

7.4.8.1 Example: Communication from Master to Slave (Host to PN544)

If the host wants to write data, it will put NSS to low and will apply the clock on SCK and puts the data on MOSI. Between every character, NSS is toggled. If there is no data pending on PN544, the PN544 shall put 00h at MISO.

- If the LLC frame is longer than the length claims to be, the length field shall be taken as reference.

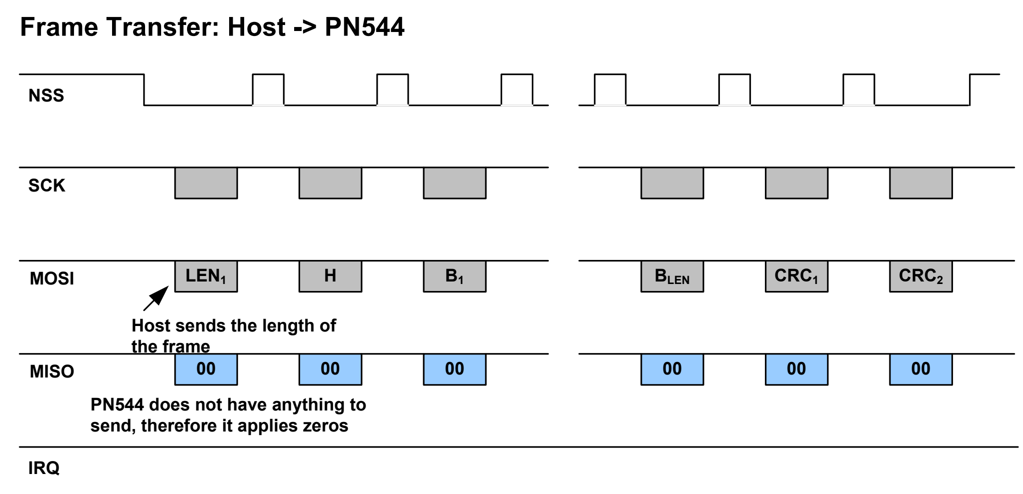

Frame Transfer: Host -> PN544
NSS
SCK
MOSI
LEN1
H
B1
BLEN
CRC1
CRC2
Host sends the length of the frame
MISO
00
PN544 does not have anything to send, therefore it applies zeros
IRQ

Fig 24. Sample SPI Host to PN544 Transfer

7.4.8.2 Communication from Slave to Master (PN544 to Host)

The PN544 will set to high the IRQ line. The host shall now apply the clock and toggle NSS as long as no data is pending any more. This can be observed by either processing the first byte which contains the length information or by looking at the IRQ line which goes back to low level again once the buffer is emptied.

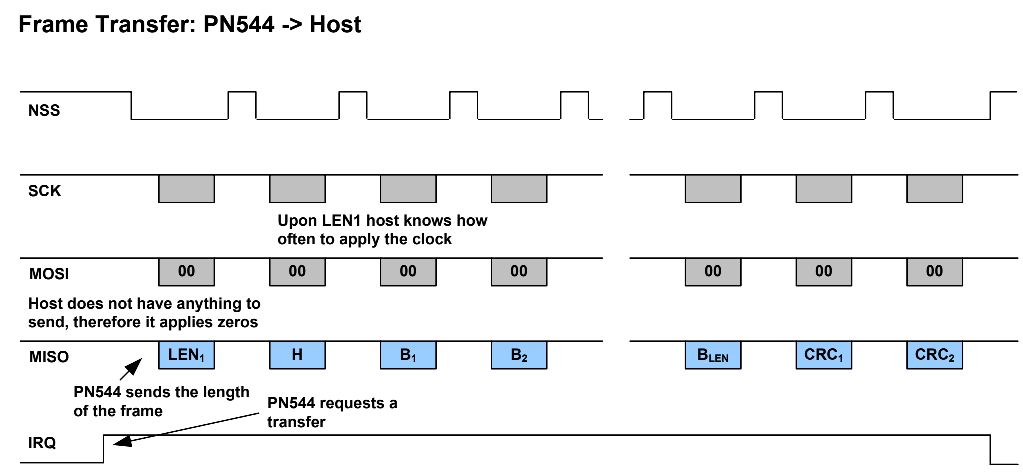

Frame Transfer: PN544 -> Host
NSS
SCK
Upon LEN1 host knows how often to apply the clock
MOSI
00
Host does not have anything to send, therefore it applies zeros
MISO
LEN1
H
B1
B2
BLEN
CRC1
CRC2
PN544 sends the length of the frame
PN544 requests a transfer
IRQ

Fig 25. Sample SPI PN544 to Host Transfer

7.4.8.3 Duplex Communication

The PN544 is capable of receiving and sending data at the same time. This situation can be recognized when the other channel is conveying a non zero value. As soon as a byte is arriving which has a value greater than 0, it shall be handled as a regular frame and shall therefore handle this byte as length information. The host shall then apply the clock as long as there is data pending, whichever side has more data to convey.

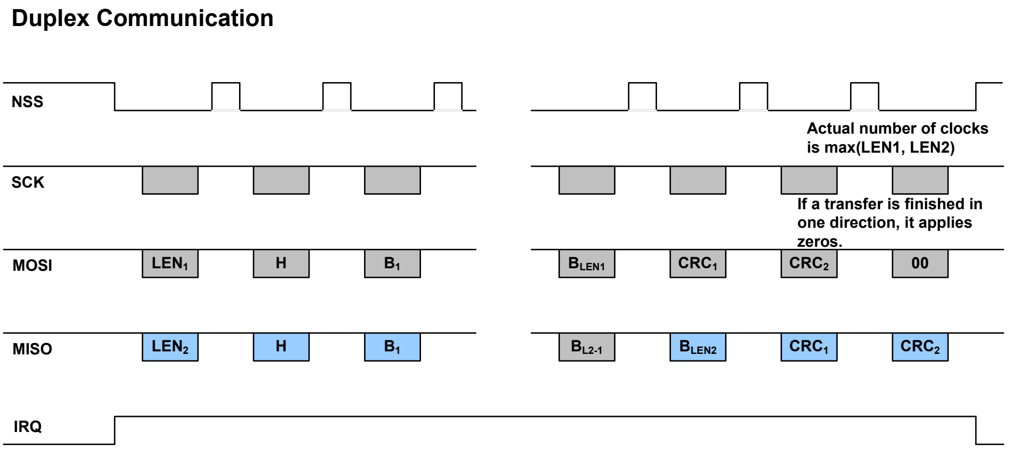

Duplex Communication
NSS
SCK
MOSI
MISO
IRQ
LEN1
H
B1
LEN2
H
B1
BLEN1
CRC1
CRC2
00
BL2-1
BLEN2
CRC1
CRC2
Actual number of clocks is max(LEN1, LEN2)
If a transfer is finished in one direction, it applies zeros.

Fig 26. Sample SPI Duplex Transfer

8. ETSI Host Controller Interface Compliancy

This chapter describes the functional interface provided by PN544. This API implements the part of ETSI HCI specification [1] concerning the CLF.

It handles the HCI Network management and defines how to control the CLF to act as Reader or Card emulation.

This chapter describes the HCI commands and registries supported by PN544.

8.1 ETSI HCI Commands/ Events Supported

To have more details on commands/ events, please refer to the HCI specification [1].

Table 9. HCI All Gates

All Gates	*Status*
ANY_SET_PARAMETER	Supported on all pipes
ANY_GET_PARAMETER	Supported on all pipes
ANY_OPEN_PIPE	Supported on all pipes
ANY_CLOSE_PIPE	Supported on all pipes

Table 10. HCI Gate Administration

Gate Administration	*Status*
ADM_CREATE_PIPE	Fully Supported (pipe routed or not)
ADM_NOTIFY_PIPE_CREATED	Fully Supported (pipe routed)
ADM_DELETE_PIPE	Fully Supported (pipe routed or not)
ADM_NOTIFY_PIPE_DELETED	Fully Supported (pipe routed)
ADM_CLEAR_ALL_PIPE	Fully Supported (pipe routed or not)
ADM_NOTIFY_ALL_PIPE_CLEARED	Fully Supported (pipe routed)

Table 11. HCI Gate Link Manager Service

Gate Link Manager Service	*Status*
No specific command for this gate	

Table 12. HCI Gate Identity Management

Gate Identity Management	*Status*
No specific command for this gate	

Table 13. HCI Gate Loopback

Gate Loop-Back	*Status*
EVT_POST_DATA	Supported (*fragmented message supported up to 255 bytes*)

Table 14. HCI Gate Reader A

Gate Reader A	*Status*
WR_XCHGDATA	Supported (*fragmented message supported up to 260 bytes*)
EVT_READER_REQUESTED	Supported
EVT_END_OPERATION	Supported
EVT_TARGET_DISCOVERED	Supported

Table 15. HCI Gate Reader B

Gate Reader B	*Status*
WR_XCHGDATA	Supported (*fragmented message supported up to 260 bytes*)
EVT_READER_REQUESTED	Supported
EVT_END_OPERATION	Supported
EVT_TARGET_DISCOVERED	Supported

Table 16. HCI Gate Card Emulation A

Gate Card Emulation Type A	*Status*
EVT_SEND_DATA	Supported in emission and reception (*fragmented message supported up to 261 bytes*)
EVT_FIELD_ON	Supported
EVT_CARD_DEACTIVATED	Supported
EVT_CARD_ACTIVATED	Supported

Gate Card Emulation Type A	*Status*
EVT_FIELD_OFF	Supported

Table 17. HCI Gate Card Emulation B

Gate Card Emulation Type B	*Status*
EVT_SEND_DATA	Supported in emission and reception (*fragmented message supported up to 255 data bytes*)
EVT_FIELD_ON	Supported
EVT_CARD_DEACTIVATED	Supported
EVT_CARD_ACTIVATED	Supported
EVT_FIELD_OFF	Supported

Table 18. Connectivity Gate

Connectivity Gate	*Status*
PRO_HOST_REQUEST	UICC proactive session
EVT_CONNECTIVITY	UICC proactive session
EVT_END_OF_TRANSACTION	UICC proactive session
EVT_TRANSACTION	UICC proactive session
EVT_OPERATION_ENDED	UICC proactive session
EVT_STANDBY	UICC proactive session

8.2 ETSI HCI Registries Supported

To have more details on the following registries, please refer to the HCI specification [1].

Table 19. HCI Gate Administration

Gate Administration	*Status*
SESSION_IDENTITY	Supported
MAX_PIPE	Supported
WHITELIST	Supported
HOSTS_LIST	Supported

Table 20. HCI Gate Link Management

Gate Link Manager Service	*Status*
REC_ERROR	Supported

Table 21. HCI Gate Identity Management

Gate Identity Management	*Status*
VERSION_SW	Supported
VERSION_HARD	Supported
VENDOR_NAME	Supported
MODEL_ID	Supported
HCI_VERSION	Supported
GATES_LIST	Supported
LOW_POWER_SUPPORT	Supported

Table 22. HCI Gate Loop-Back

Gate Loop-Back	*Status*
No registry in this gate	

Table 23. HCI Gate Reader A

Gate Reader A	*Status*
UID	Supported
ATQA	Supported
APPLICATION_DATA	Supported
SAK	Supported
FWI, SFGT	Supported
DATARATE_MAX	Supported

Table 24. HCI Gate Reader B

Gate Reader B	*Status*
PUPI	Supported
APPLICATION_DATA	Supported
AFI	Supported
HIGHER_LAYER_RESPONSE	Supported
HIGHER_LAYER_DATA	Supported

Table 25. HCI Gate Emulation A

Gate Card Emulation Type A	*Status*
MODE	Supported
UID	Supported
SAK	Supported
ATQA	Supported
APPLICATION_DATA	Supported
FWI, SFGI	Supported
CID_SUPPORT	Supported
CLT_SUPPORT	Supported
DATARATE_MAX	Supported

Table 26. HCI Gate Emulation B

Gate Card Emulation Type B	*Status*
MODE	Supported
PUPI	Supported
AFI	Supported
ATQB	Supported
HIGHER_LAYER_RESPONSE	Supported (*maximum size of 31 bytes*)
DATARATE_MAX	Supported

9. NXP Host Controller Interface

PN544 provides proprietary gates to enhance the HCI feature set given in [1] . Here is a description of them. This is the block named “PN544 HCI” in “Fig 1 System Overview”.

9.1 Access Rights NXP HCI Registry

Explaining the NXP HCI is done in table format. One row lists the access rights of the parameter. The first abbreviations are:

RO: Read Only (protection against writing)

RW: Read/ Write allowed

The second abbreviation given is the memory location to which the parameter is written.

REG: Register, volatile memory

Values are written to a memory which does not have any limitation with respect to write cycles. It is valid until a power off/up sequence is performed or until Hardware Reset (VEN pin use).

EEPROM: non volatile memory

Values written to this memory are stored in EEPROM area of the chip. Values are persistent even if a reset or power down was performed. Please be aware that these memory cells have a limit of ~100k write cycles. However, a true write cycle is performed only when the value changes, so rewriting the same EEPROM stored registry value multiple times does not affect the lifetime of the EEPROM.

RAM: volatile memory

Values written to this memory are persistent until a reset (VEN pin use) or power down is seen. There are no limitations with respect to write cycles.

9.2 Initialization & Default mode of PN544

This chapter describes the PN544 setup at boot level, and the different steps to perform the initialization. The following chapters describe the pipes that have to be created, opened or closed, and how it has to be managed by the Host Controller.

9.2.1 Gates & Pipes

Gates: The PN544 gates are defined in accordance with the ETSI standard. For each gate, an associated registry is proposed to store the parameters that are related to it. Each gate is managed thanks to a generic (ETSI) or/and a specific (NXP) command set. (See 'NXP Host Controller Interface' chapter for details on specific command set)

Pipes: As defined in the ETSI standard, a logical communication channel has to be created between the different host gates.

9.2.2 Pipe ID allocation

PN544 allocates pipe IDs according to the following rules:

- Pipe IDs related to pipes created between PN544 and host are allocated in the following range: from '02' to '3F'.
- Pipe IDs related to pipes created between PN544 and UICC are allocated in the following range: from '40' to '5F'.
- Pipe IDs related to pipes created between Host and UICC (routed pipes) are allocated in the following range: from '60' to '6F'.
- Pipe IDs are allocated in an incremental way (from the free pipe ID list). E.g. if, from the initial configuration, the host creates two pipes to gates of the PN544, first pipe will be assigned as pipe ID '02' and second one to pipe ID '03').
- When a pipe is deleted, its pipe ID becomes free and can then be re-assigned to another pipe.

9.2.3 First Setup – Initialization phase

For the 1st PN544 access, the Host Controller has to setup the different gates needed.

These gates are listed as follows:

From ETSI HCI specification [1]:

- Management gates (Administration, Link Management, Identity Management, Loopback...),

- RF configuration (Card RF gate, RF Reader Gates).

From NXP HCI proprietary:

- System management,
- SWP,
- Polling Loop,
- NFC-WI,
- MIFARE Reader (Integrated to generic Reader Gates),
- FeliCa Reader,
- Jewel Reader,
- ISO15693,
- NFC-IP1,
- Additional commands - Reader RF (Integrated to generic Reader Gates).

Here is an example of Type A Reader RF gate initialization:

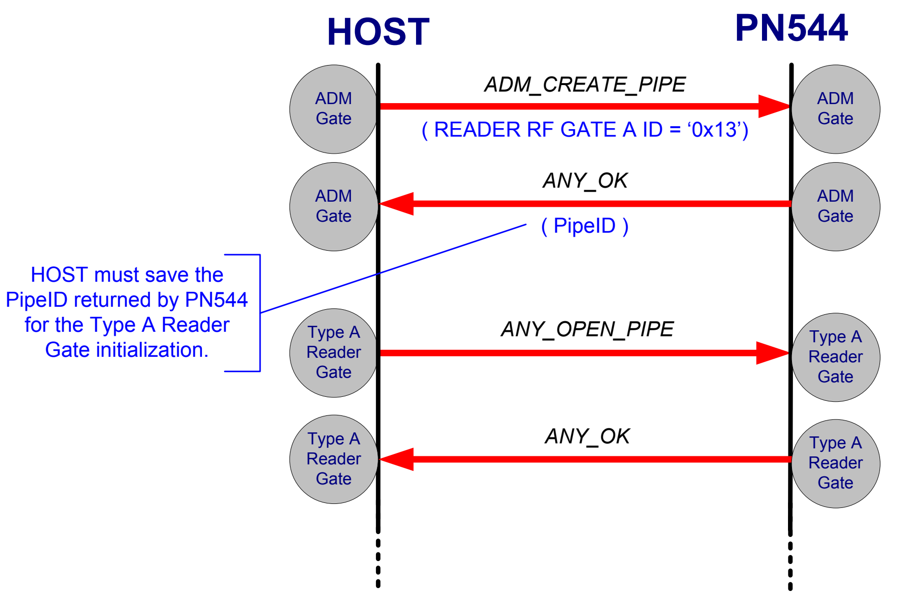

HOST
PN544
ADM Gate
ADM_CREATE_PIPE
(READER RF GATE A ID = '0x13')
ADM Gate
ANY_OK
(PipeID)
HOST must save the PipeID returned by PN544 for the Type A Reader Gate initialization.
Type A Reader Gate
ANY_OPEN_PIPE
Type A Reader Gate
ANY_OK

Fig 27. Type A Reader RF Gate initialization

For all the gates with dynamic Pipe ID, the initialization should be performed as follows:

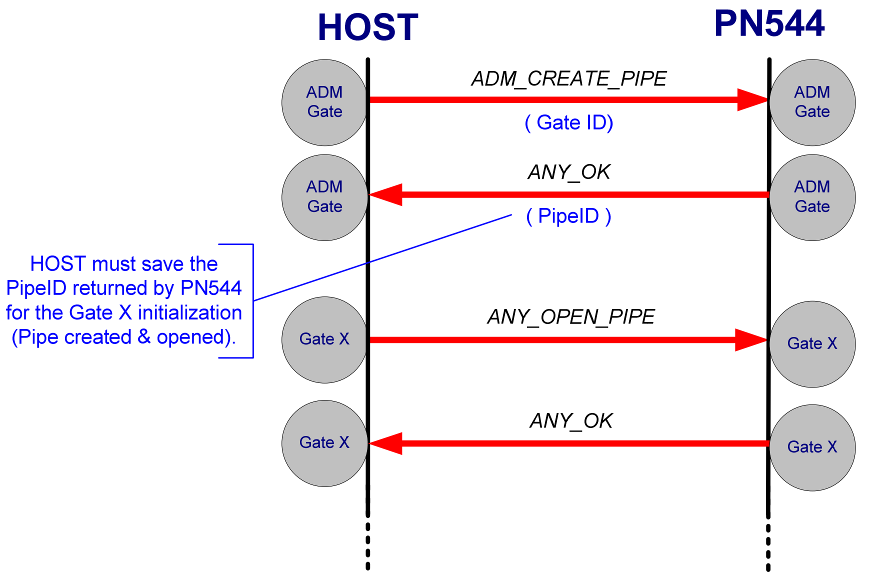

HOST
PN544
ADM Gate
ADM_CREATE_PIPE
(Gate ID)
ADM Gate
ANY_OK
(PipeID)
HOST must save the PipeID returned by PN544 for the Gate X initialization (Pipe created & opened).
Gate X
ANY_OPEN_PIPE
Gate X
ANY_OK

Fig 28. Gate 'X' initialization

Note: Only Link management gate & administration gate have a static Pipe ID, all other gates have dynamic Pipe ID.

The first PN544 initialization can be performed as follows:

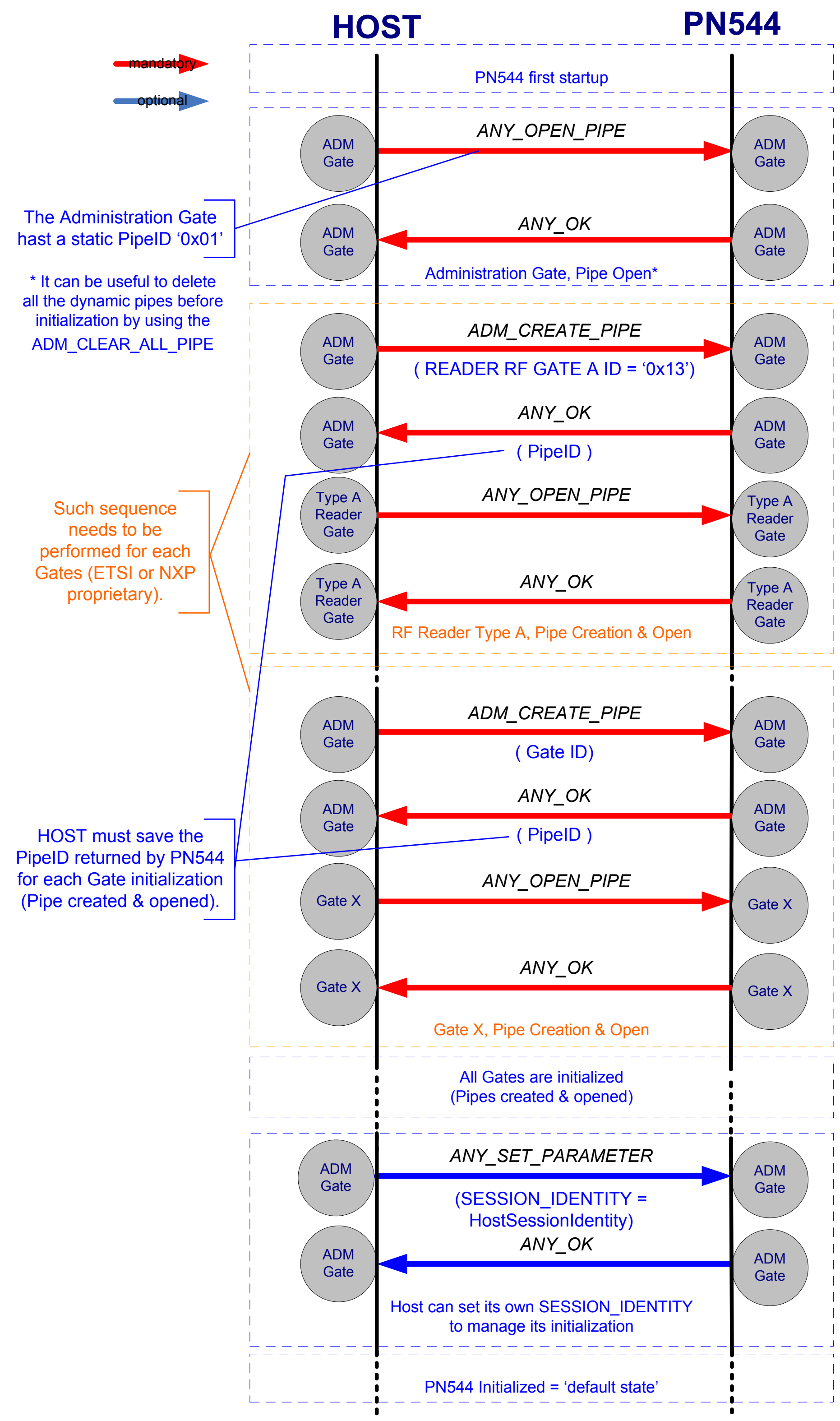

HOST
PN544
mandatory
optional
PN544 first startup
ANY_OPEN_PIPE
ADM Gate
ANY_OK
Administration Gate, Pipe Open*
The Administration Gate hast a static PipeID ‘0x01’
* It can be useful to delete all the dynamic pipes before initialization by using the ADM_CLEAR_ALL_PIPE
ADM_CREATE_PIPE
(READER RF GATE A ID = ‘0x13’)
ANY_OK
(PipeID)
ANY_OPEN_PIPE
Type A Reader Gate
ANY_OK
RF Reader Type A, Pipe Creation & Open
Such sequence needs to be performed for each Gates (ETSI or NXP proprietary).
ADM_CREATE_PIPE
(Gate ID)
ANY_OK
(PipeID)
ANY_OPEN_PIPE
Gate X
ANY_OK
Gate X, Pipe Creation & Open
HOST must save the PipeID returned by PN544 for each Gate initialization (Pipe created & opened).
All Gates are initialized (Pipes created & opened)
ANY_SET_PARAMETER
(SESSION_IDENTITY = HostSessionIdentity)
ANY_OK
Host can set its own SESSION_IDENTITY to manage its initialization
PN544 Initialized = ‘default state’

Fig 29. PN544 initialization

As soon as Pipes are created and opened, PN544 will save this setup in EEPROM (Pipe states are persistent). It means that at the next PN544 restart (VEN reset); PN544 will stay in this mode also called ‘default mode’.

WARNING: As this default configuration is stored in the internal EEPROM,

It is recommended to limit the initialization process to the First time setup.

9.2.4 After First Setup – Default state

After an initial configuration of the PN544, there is no further need to perform the same level of configuration on subsequent initializations. As Pipes states are persistent, PN544 will start automatically in its ‘default/initialized mode’ which means that Gates & Pipes are ‘ready to use’.

At Host level, it should be easy to check if the initialization has already been performed by checking the SESSION_IDENTITY value (Administration Gate).

At the end of the first PN544 initialization, the HOST can set the SESSION_IDENTITY to its own value (see Fig 29). For the next startup it has to check if the SESSION_IDENTITY returned by PN544 is the one it has set.

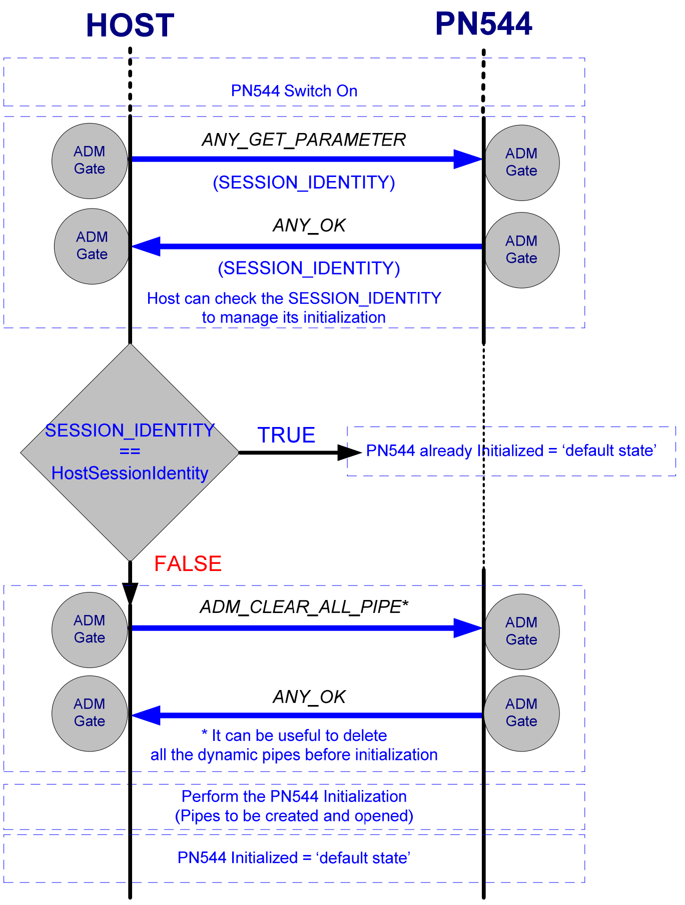

HOST
PN544
PN544 Switch On
ADM Gate
ANY_GET_PARAMETER
(SESSION_IDENTITY)
ANY_OK
(SESSION_IDENTITY)
Host can check the SESSION_IDENTITY to manage its initialization
SESSION_IDENTITY == HostSessionIdentity
TRUE
PN544 already Initialized = 'default state'
FALSE
ADM_CLEAR_ALL_PIPE*
ANY_OK
* It can be useful to delete all the dynamic pipes before initialization
Perform the PN544 Initialization (Pipes to be created and opened)
PN544 Initialized = 'default state'

Fig 30. After First Setup – Default state

9.3 System Management

This chapter describes the System Management of PN544.

The host could manage the management of the system through a dedicated gate ***'PN544Mgt'***.

Table 27. PN544 System Management Gate

Gate	G_{ID}
PN544Mgt gate	'90'

Using this gate the host is able to:

- Put the system into "Autonomous" mode. Autonomous mode means that system (PN544 + secure elements) will work without any interaction with the host. In this mode only Emulation feature is supported (according to previous configuration). This mode is left as soon as PN544 receive any data from the host or in case of PN544 Reset.

- Receive information about PN544 behaviour.
- Access PN544 configuration (EEDATA, HW registers) via Read & Write commands.
- Perform self tests.

Table 28. Self test commands

Value	Command	Description
'20'	*NXP_SELF_TEST_ANTENNA*	Perform antenna self test
'21'	*NXP_SELF_TEST_SWP*	Perform SWP self test. Purpose of the test is to check SWIO pin connectivity. Principle is to communicate (SWP activation procedure) with the secure element connected over SWP interface.

Note: SWP interface can be configured by setting the *SWP_Bitrate, SwpMgt_Request_Power & SWP_CurrentThreshold EEPROM* area. (See **'SWP configuration'** chapter).

The *NXP_SELF_TEST_ANTENNA* command has the following parameters:

Table 29. Antenna Self test Command

Description	Length
BoundaryConditions	4

Boundary conditions used during Antenna Self test:

- 1st one relates to RF sensitivity on TX1.

- 2nd one relates to RF sensitivity on TX2.

- 3rd one relates to Current detection level on TX1 & TX2.

- 4th one relates to ANT1/ANT2 circuitry check.

This parameter is optional, in case it is not set in the command the latest set values are used to perform the test (default values are 0x05 0x05 0x04 0x09).

The response to the *NXP_SELF_TEST_ANTENNA* command is as follows:

Table 30. Antenna Self test Command Response

Description	Length
ErrorCode	1
Measured Values	4

ErrorCode field indicates which condition has failed during the test:

- 0x00: No failure.

- 0x79: 1st condition failed (RF sensitivity on TX1).

- 0x7A: 2nd condition failed (RF sensitivity on TX2).

- 0x7B: 3rd condition failed (Current detection level on TX1 & TX2).

- 0x7C: 4th condition failed (ANT1/ANT2 circuitry check).

Measured Values: Last measured value during self Test, these values can be used to run the antenna test again in case of failure.

NXP_SELF_TEST_ANTENNA boundary conditions are described in [11].

The response to the *NXP_SELF_TEST_SWP* command is as follows:

Table 31. SWP Self test Command Response

Description	Length
ErrorCode	1
PmuVccStatus	1

ErrorCode field indicates which condition has made the test failing:

0x00: No failure

0x01: SWP self test failed

PmuVccStatus field indicates if the PmuVcc is present or not:

0x00: PmuVcc not present

0x01: PmuVcc present

Note: Pay attention that to make the detection working, PmuVcc must be present when PN544 is booting.

Table 32. Type Approval Test commands

Value	Command	Description
‘25’	*NXP_PRBS_TEST*	Perform PRBS test sequence. As soon as the test is started, PN544 generates PRBS9 [511 bits] data stream over the contactless interface. PRBS test is ended via PN544 external reset (when the test is started no response to this command is sent by PN544 to the host).

The *NXP_PRBS_TEST* command has the following parameters:

Table 33. PRBS test Command

Description	Length
Technology	1
Bitrate	1

Technology indicates the RF modulation type to be used for the test

Table 34. PRBS test Command Technology Parameter

Value	Description
0x00	Type A
0x01	Type B

Value	Description
0x02	Type F
0x03	No modulation
others	RFU

Baud rate indicates the RF modulation bitrates to be used for the test

Table 35. PRBS test Command Bitrate Parameter

Value	Description
0x00	106 kbps
0x01	212kbps
0x02	424 kbps
others	RFU

Table 36. Autonomous Event

Value	Event	Description
‘01’	NXP_EVT_SET_AUTONOMOUS	This event is sent by the host to put the system in Autonomous mode

Note: This event is interesting to inform PN544 when Host keep PVDD available and does not want to communicate on Host Link anymore. If Host is going to disable PVDD, this event is not mandatory; PN544 will switch in Autonomous mode automatically.

Table 37. Information Event

Value	Event	Description
‘10’	NXP_EVT_INFO_TXLDO_OVERCUR	This event is sent to the host to inform about overcurrent issue.
‘11’	NXP_EVT_INFO_PMUVCC	This event is sent to the host to inform about PmuVcc signal switch which can then impact SimVcc (implies internal reactivation of the SWP link).

Value	Event	Description
‘12’	NXP_EVT_INFO_EXT_RF_FIELD	This event is sent to the host to inform about external RF field presence. This notification is sent, if at least one Card Emulation is enabled.
‘13’	NXP_EVT_INFO_MEM_VIOLATION[3]	This event is sent to the host to inform about PN544 internal memory violation. This violation could have been caused by: - EEPROM memory corruption (protected by CRC, see *NXP_INFO_EEPROM_ERR* registry entry*)* - Wrong memory access (cause internal PN544 reboot)
‘14’	NXP_EVT_INFO_TEMP_OVERHEAT	This event is sent to the host to inform about temperature overheat. This issue is triggering the internal protection mode.
‘15’	NXP_EVT_INFO_LLC_ERR	This event is sent to the host to inform LLC error counter is reached on SWP.

Note: ***NXP_EVT_INFO_TXLDO_OVERCUR*** event can occur in Reader mode, in case of Antenna connection break (or Antenna close to metal area). In that case PN544 will automatically switch off RF and Reader function is disabled (NXP_PL_RDPHASES is reset).

The *NXP_EVT_INFO_PMUVCC* event has the following parameter:

Table 38. PMUVCC Event

Description	Length
Status	1

Status field indicates current PMUVCC state

- 0x00: indicates PMUVCC switched from ON to OFF

- 0x01: indicates PMUVCC switched from OFF to ON

[3] To recover first reset the IC and restart the application – if the problem still exists re-download PN544 firmware

The *NXP_EVT_INFO_EXT_RF_FIELD* event has the following parameter:

Table 39. EXT_RF_FIELD Event

Description	Length
Presence	1

Presence field indicates current external RF field presence

- 0x00: indicates no RF field present

- 0x01: indicates RF field present

The *NXP_EVT_INFO_TEMP_OVERHEAT* event has the following parameter:

Table 40. TEMP_OVERHEAT Event

Description	Length
Status	1

Status indicates about the protection procedure status

- 0x00: indicates temperature overheat detected then PN544 will enter standby mode (50ms after the event has been sent)

- 0x01: indicates temperature cooled down then PN544 has rebooted

Table 41. R/W Commands

Value	Event	Description
'3E'	*NXP_READ*	This command is used by the host to read PN544 memory
'3F'	*NXP_WRITE*	This command is used by the host to write PN544 memory

The *NXP_READ* command has the following parameters:

Table 42. Read Command

Description	Length
Address	3

Address contain the memory address to be read

The response to the *NXP_READ* command is as follows:

Table 43. Read Command Response

Description	Length
Value	1

Value contain the memory read value

The *NXP_WRITE* command has the following parameters:

Table 44. Write Command

Description	Length
Address	3
Value	1

Address contain the memory address to be read

Value contain the memory value address to be written

The response to the *NXP_WRITE* command is as follows:

Table 45. Write Command Response

Description	Length
Value	1

Value contain the memory read value after write operation

Table 46. System Management Registry

<table>
<tr><th>Id</th><th>Name</th><th>Access Rights</th><th>Comment</th><th>Length</th><th>Default</th></tr>
<tr><td>‘02’</td><td><i>NXP_INFO_NOTIFICATION</i></td><td>RW
(EE)</td><td>Indicates on information s the host wants to be notified of.<table><tr><td>Bit</td><td>Information event</td></tr><tr><td>0</td><td>Overcurrent</td></tr><tr><td>1</td><td>PmuVcc switch</td></tr><tr><td>2</td><td>External RF field</td></tr><tr><td>3</td><td>Memory violation[4]</td></tr><tr><td>4</td><td>Temperature overheat</td></tr><tr><td>5</td><td>LLC Error Counter reached on UICC</td></tr><tr><td>6...7</td><td>RFU</td></tr></table>0x0 -> do not notify
0x1 -> notify</td><td><i>1</i></td><td><i>0x00</i></td></tr>
<tr><td>‘03’</td><td><i>NXP_INFO_EEPROM_ERR</i></td><td>RO
(RAM)</td><td>Indicates status of the last EEPROM check procedure:
- Bit 0: CRC check on Patch area
- Bit 1: CRC check on ConfigPage area
- Others bits are RFU
0x0 -> no CRC error
0x1 -> CRC error
In case of CRC error on Patch area, patches are disabled.
In case of CRC error on ConfigPage area, default configuration is applied.</td><td><i>1</i></td><td><i>0x00</i></td></tr>
</table>

[4] To recover first reset the IC and restart the application – if the problem still exists re-download PN544 firmware

9.3.1 Default mode to Standby mode

When PN544 is not used (No RF communication, no Host communication, no UICC communication), it stays in a “default mode” which can be set by the *PWR_STATUS* EEPROM area.

2 modes are available: *ActiveBAT* or *Standby*.

ActiveBAT mode means that the PN544 is kept awake (default mode).

Standby mode is the PN544 low power mode; this is the best setup to save power.

PN544 default mode is set to *ActiveBAT.*

To save power the default mode can be changed to *Standby*. To setup *Standby* mode as default PN544 mode, the following registry has to be set:

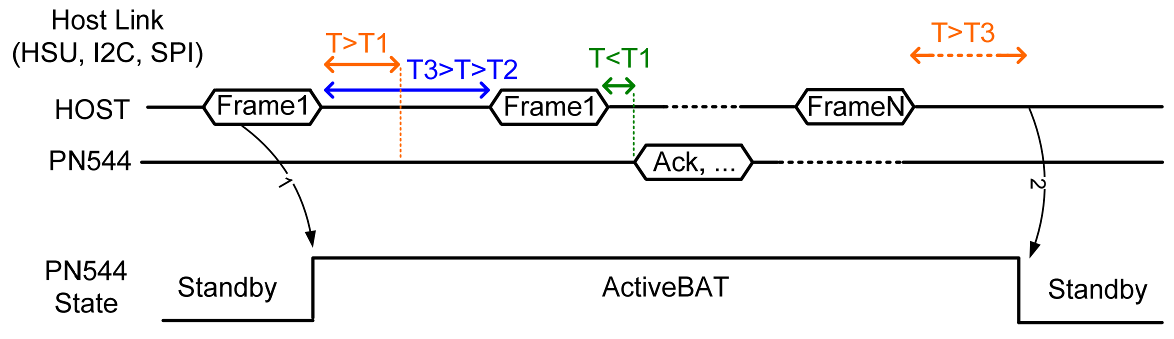
Table 47. Setting for set Default mode to Standby

Registry	Length (bytes)	Value(s)	comments
PWR_STATUS	1	**0x01**	Define Standby mode as the ‘default’ PN544 mode.

9.3.2 Host Link Wake-up from Standby mode

When PN544 is in Standby mode, it is still possible to communicate with through the Host link (HSU, I2C or SPI).

However it can happen that PN544 will not answer to the first frame sent. Host needs to re-send the frame according to SHDLC definition (See [2])



(4) T1: Acknowledge time, T1 ≤5ms.

(5) T2: Guarding/transmit time, T2 ≥ T1+1ms.

(6) T3: Inactivity time, T3=1s.

Fig 31. PN544 Host Link wake-up

Step1: When receiving a frame on host link, PN544 can switch from Standby mode to ActiveBAT mode. In case of T1 timeout, meaning that PN544 has not answered in time, Host shall retransmit the frame in a delay from T2 to T3.

Step2: T3 timer is handled by PN544 to detect inactivity on Host Link. If there is no more activity during T3, PN544 will switch back to Standby mode.

Note: T3 timer is reset each time a frame is received by PN544. The T3 timeout value can be changed by setting the *TO_Before_SDTBY_MSB & TO_Before_SDTBY_LSB* EEPROM area. (See ***'0*** chapter)

Note: The T1 timeout value can be changed by setting the *AckHostTimeout_LSB & AckHostTimeout_MSB* EEPROM area. (See ***'0*** chapter)

9.3.3 Information notification

This feature is mainly for debug/trace purpose to inform Host Controller that:

1. An OverCurrent occurs.
2. PMUVCC has changed.
3. External RF Field is present.

This feature is optional for Host Controller.

To setup the notification, the following registry is required

Table 48. Setting for notification

Registry	Length (bytes)	Value(s)	comments
NXP_INFO_NOTIFICATION	1	**[bit mask]**	Allow PN544 to send information event

Note: By default the information notification is disabled.

As soon as the information notification is enabled, the host will be notified with the following events: *NXP_EVT_INFO_TXLDO_OVERCUR, NXP_EVT_INFO_PMUVCC or NXP_EVT_INFO_EXT_RF_FIELD.*

9.3.4 Default Secured Element in ‘Power by the Field’ mode

When PN544 is in ‘Power by the Field’ mode, it is able to supply only one Secured element. This default secured element can be UICC/SIM or NFC-WI/SmartMX.

To setup such a configuration, the *SE_Conf* data should be set (refer to **‘SE configuration’** chapter).

9.3.5 Autonomous mode

When Host does not want to communicate with PN544 (i.e. when switching OFF the system) it can send the *NXP_EVT_SET_AUTONOMOUS* event. If Host is going to disable PVDD, this event is not mandatory; PN544 will switch in Autonomous mode automatically.

This event is needed to inform PN544 to switch in Autonomous mode. In this mode, PN544 will work in Card Emulation mode only.

In **Autonomous** mode, we can have 2 different behaviors:

- If **Vbat>VbatCritical**, PN544 is able to supply Secure Element with the help of the battery.

- If **Vbat<VbatCritical**, PN544 is able to supply Secure Element and switch automatically in ‘Power by the field’ (No battery resources needed).

9.4 Clock Management

This chapter describes the clock setup, the clock request and acknowledge in the case of a clock provided by the system (Baseband, PMU, Host Controller …) to the PN544.

Clock management is a combination of request & acknowledge. The system has to setup the way it wants the PN544 to request the clock, and the way to give back acknowledge to PN544.

To setup the Clock, some EEPROM data have to be set: *PlClockRequest*, *PlClockAck*, *PlClockTimeout*, *FRAC_ClkSel* and *FRAC4_xxx*.

The ***‘PollingLoopMgt’*** gate has to be used to manage HCI events.

(*Refer to the* ***‘PollingLoopMgt’*** and ***‘0*** *chapters for details*)

9.4.1 Clock Setup

PN544 can use the clock provided by an external oscillator or request the clock from the mobile platform.

9.4.1.1 Use of external oscillator

To setup the use of external oscillator (Crystal/Quartz), the following EEPROM area has to be set:

Table 49.Setting for external oscillator

Name	Length (bytes)	Value(s)	Comments
HW_Conf	1	Bit7 set to 1. (Default value)	Disable the Use of internal FracNPLL
FRAC_ClkSel	1	0	No input clock

Note: No others settings are required for this configuration. It means that the following setup are not needed and not taken into account by PN544.

9.4.1.2 Use of system clock

To setup the use of the system clock, the following EEPROM area has to be set:

Table 50.Setting for external oscillator

Name	Length (bytes)	Value(s)	Comments
HW_Conf	1	Bit7 set to 0.	Enable the Use of internal FracNPLL for clock generation
FRAC_ClkSel	1	0x01 to 0x05	Indicate the clock frequency provided by the system
FRAC4_DIV FRAC4_OOF0 FRAC4_OOF1 FRAC4_OOF2 FRAC4_CAL0 FRAC4_CAL1	6	[Registers values]	Registers values correspond to the input clock frequency provided by the system (To setup only if *FRAC_ClkSel* is set to 0x05, meaning ‘Customized Clock’ to be used).

Note: The *FRAC4_xxx* values (registers settings) are described in a separate document [9].

9.4.2 Supported Clock Request/Acknowledge setup

This chapter gives an overview of the clock mechanism; the detailed part is described in the next chapters.

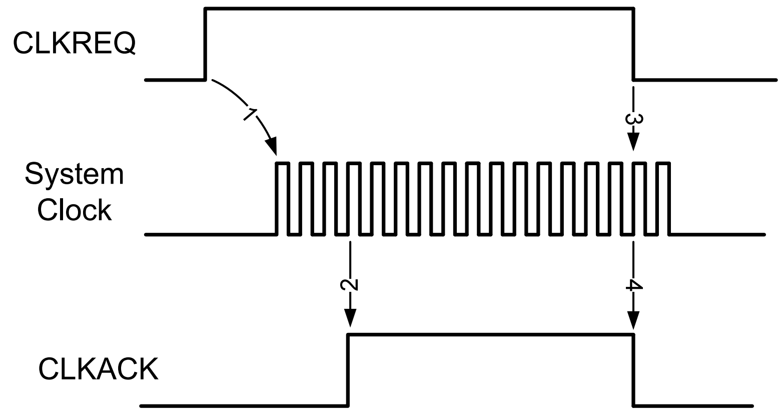
The supported Clock Request and Acknowledgment setups combination are described in the following table:

Table 51.Supported Clock Setup

Request Acknowledge	HCI Event (SW)	GPIO Pin (HW)
HCI Event (SW)	√ (Fig 34)	✗
GPIO Pin (HW)	✗	√(Fig 32)
Timeout	√(Fig 35)	√(Fig 33)

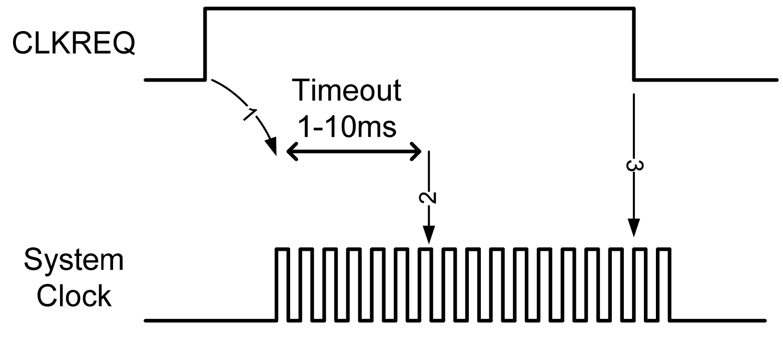
9.4.2.1 Clock Request using GPIO pin

The following figures show the 2 possible use of GPIO pin for Clock Request (GPIO2) & acknowledge (GPIO1):



(1)

Fig 32. Clock request with CLKREQ pin, acknowledge with CLCKACK pin



(1)

Fig 33. Clock request with CLKREQ pin, acknowledge with Timeout

Note: ***Step2*** means that system clock is stable and available for PN544.

9.4.2.2 Clock Request using HCI Event

The following figures show the 2 possible use of HCI Event for Clock Request & acknowledge:

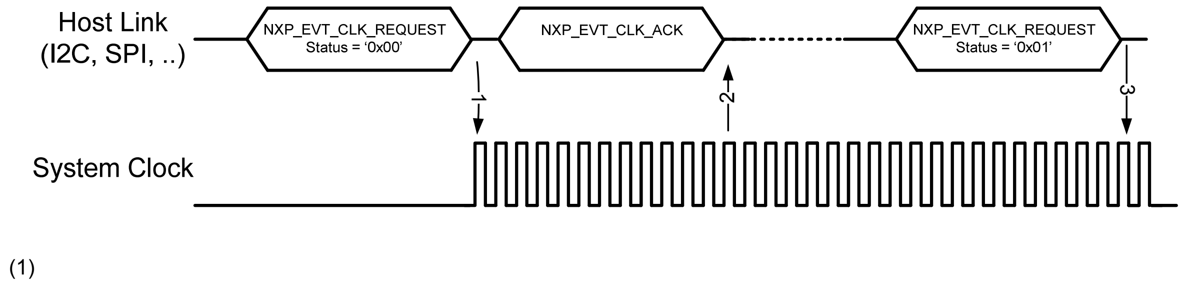


Fig 34. Clock request with HCI Event, acknowledge with HCI Event

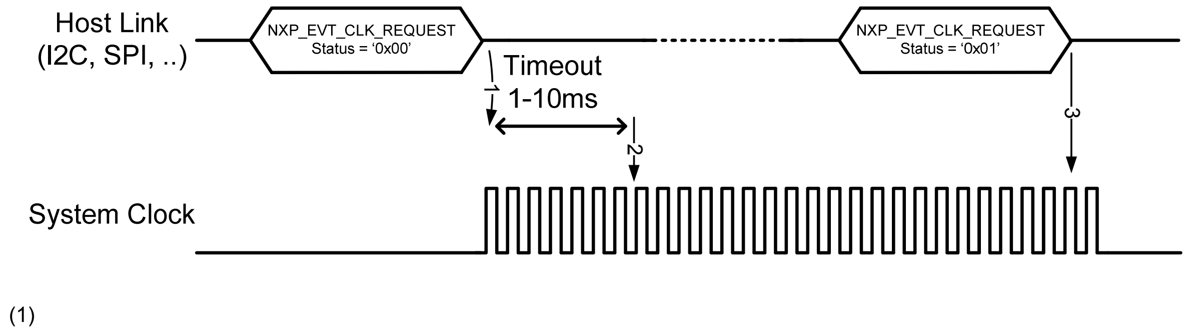


Fig 35. Clock request with HCI Event, acknowledge with Timeout

Note: ***Step2*** means that system clock is stable and available for PN544.

9.4.3 Clock Request & Release

There are 3 ways to setup the clock request.

WARNING: Clock request/acknowledge mechanism is not available with HSU link. When PN544 is in HSU mode, the 'No clock request' setup must be chosen. HSU mode imposes System clock always present.

9.4.3.1 No clock request

In the case of a clock always provided by the system, the request/acknowledge mechanism is not needed. To setup such a configuration, the following EEPROM area has to be set (See ’

HW configuration’ chapter):

Table 52. Setting for no clock request

Name	Length (bytes)	Value(s)	Comments
PlClockRequest	1	0	System clock always available

Note: No others settings are required for this configuration. It means that the following setup is not needed and not taken into account by PN544.

9.4.3.2 Request through CLKREQ pin

In this mode, PN544 will request the clock by setting the CLKREQ pin (GPIO2) to high. To setup such a configuration, the following EEPROM area has to be set:

Table 53. Setting for clock request with CLKREQ pin

Name	Length (bytes)	Value(s)	Comments
PlClockRequest	1	0x01	PN544 will request clock from system by using the CLKREQ pin

The request & release mechanism using CLKREQ pin is described in the following figure:

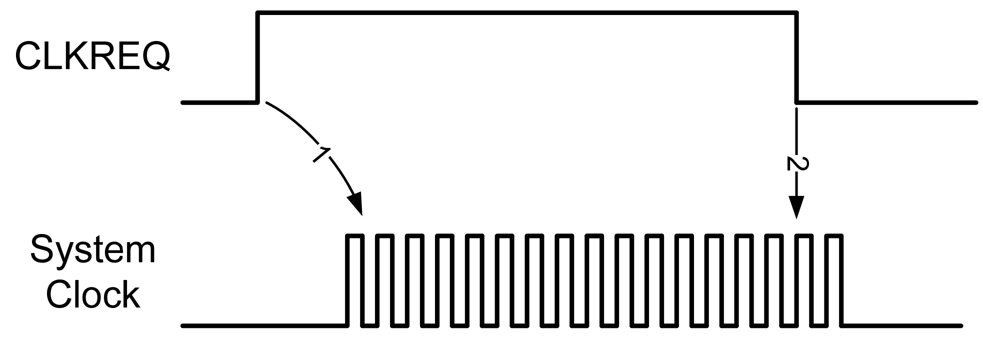


Fig 36. Clock request with CLKREQ pin

Step1: As soon as the PN544 needs system clock, it puts the CLKREQ level to high. Then system has to provide the clock. The acknowledgement from Host to PN544 is detailed in the 'clock acknowledge' chapter.

Step2: As soon as the PN544 does not need the system clock, it puts the CLKREQ level to low. Then system can disable the clock.

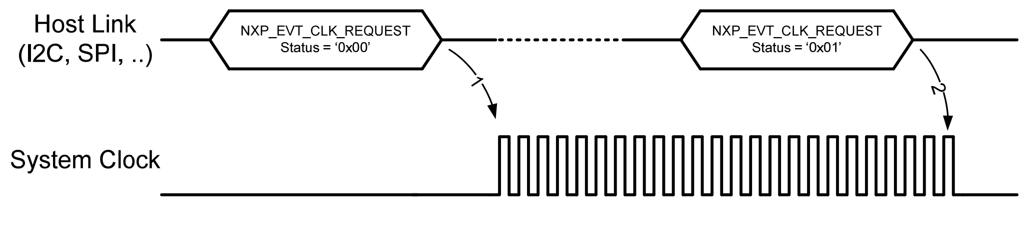
9.4.3.3 Request through NXP_EVT_CLK_REQUEST event

In this mode, PN544 will request the clock by sending an HCI event to the Host Controller. To setup such a configuration, the following EEPROM area has to be set:

Table 54.Setting for clock request with HCI event

Name	Length (bytes)	Value(s)	Comments
PlClockRequest	1	0x02	PN544 will request clock from system by using an HCI proprietary event

The request & release mechanism using NXP_EVT_CLK_REQUEST HCI event is described in the following figure:



(1)

Fig 37. Clock request with HCI event

Step1: As soon as the PN544 needs system clock, it sends the HCI event NXP_EVT_CLK_REQUEST (with status byte set to '0x00' for Clock request) to the Host controller. Then system has to provide the clock. The acknowledgement from Host to PN544 is detailed in the 'clock acknowledge' chapter.

Step2: As soon as the PN544 does not need system clock, it sends the HCI event NXP_EVT_CLK_REQUEST (with status byte set to '0x01' for Clock release) to the Host controller. Then system can disable the clock.

9.4.4 Clock Acknowledge

There are 3 ways to setup the clock acknowledgement.

9.4.4.1 Acknowledge with timeout

In this mode, PN544 will wait for a timeout to consider that system clock is available. The timeout value is configurable from 0 to 255 ms (Recommended value is 10 ms or less).

To setup such a configuration, the following EEPROM area has to be set:

Table 55.Setting for clock acknowledge with timeout

Name	Length (bytes)	Value(s)	Comments
PlClockAck	1	0x00	PN544 will wait for a timeout to ensure that system clock is available
PlClockTimeout	1	[0x00 to 0xFF]	Timeout value can be setup with 1ms step, from 0 to 255 ms

The acknowledge mechanism using timeout is described in the following figure:

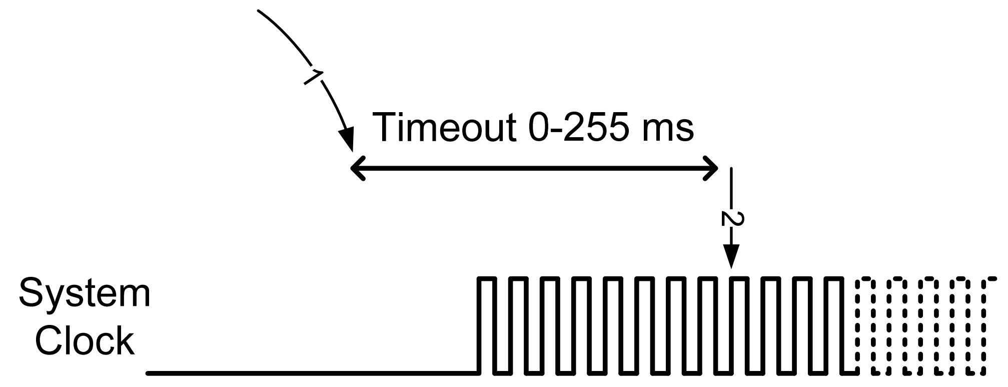


Fig 38. Clock acknowledge with timeout

Step1: PN544 request the clock from the system (for clock request, see previous chapter: ‘Clock Request & Release’, Fig 33 & Fig 35).

Step2: When the configured timeout occurs, PN544 is ensured to have the system clock available.

9.4.4.2 Acknowledge through CLKACK pin

In this mode, PN544 will wait for CLKACK (GPIO1) pin level to high to consider that system clock is available. To setup such a configuration, the following EEPROM area has to be set:

Table 56. Setting for clock acknowledge with CLKACK pin

Name	Length (bytes)	Value(s)	Comments
PlClockAck	1	0x01	PN544 will wait for the CLKACK pin to get high.

The acknowledge mechanism using CLKACK pin is described in the following figure:

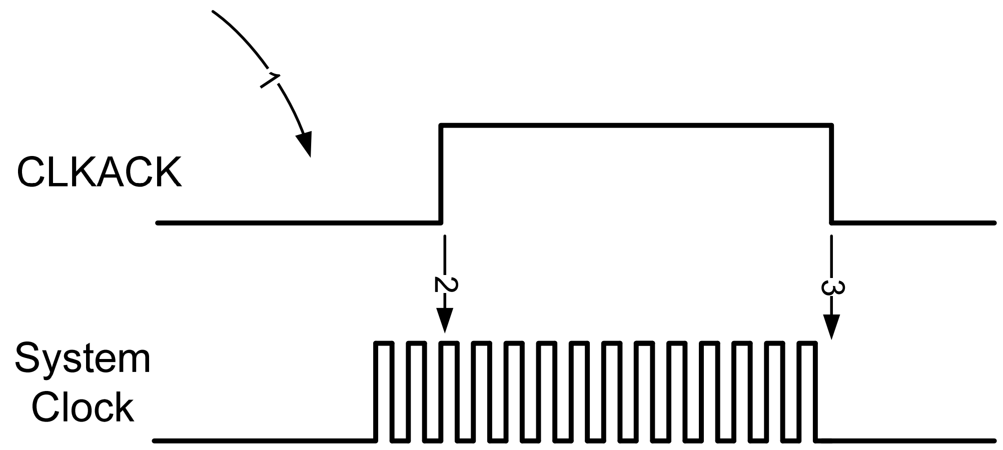


Fig 39. Clock acknowledge with CLKACK pin

Step1: PN544 request the clock from the system (for clock request, see previous chapter: ‘Clock Request & Release’).

Step2: When the system clock is available, CLKACK pin goes high. PN544 is informed that clock is available & stable.

Step3: When the system clock is no more available, CLKACK pin goes low.

9.4.4.3 Acknowledge through NXP_EVT_CLK_ACK event

In this mode, PN544 will wait for an HCI event to consider that system clock is available. To setup such a configuration, the following EEPROM area has to be set:

Table 57.Setting for clock acknowledge with HCI event

Name	Length (bytes)	Value(s)	Comments
PlClockAck	1	2	PN544 will wait for an HCI event

The acknowledge mechanism using HCI event is described in the following figure:

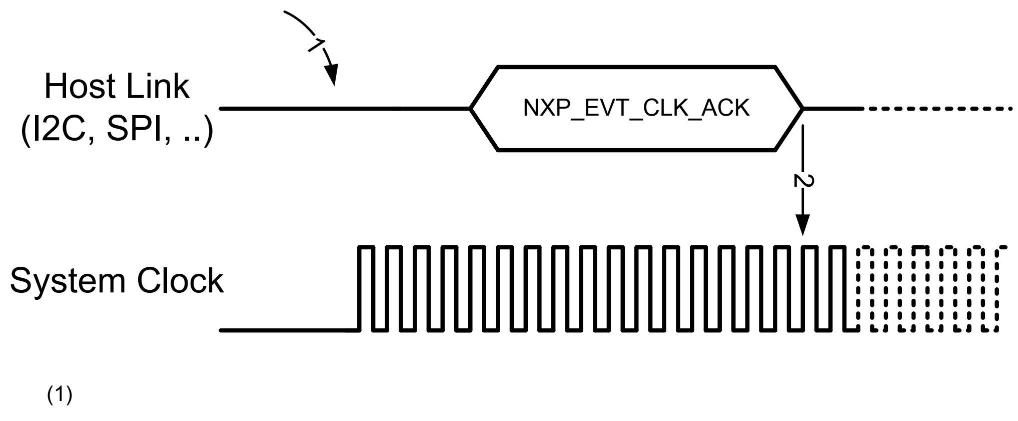


Fig 40. Clock acknowledge with HCI event

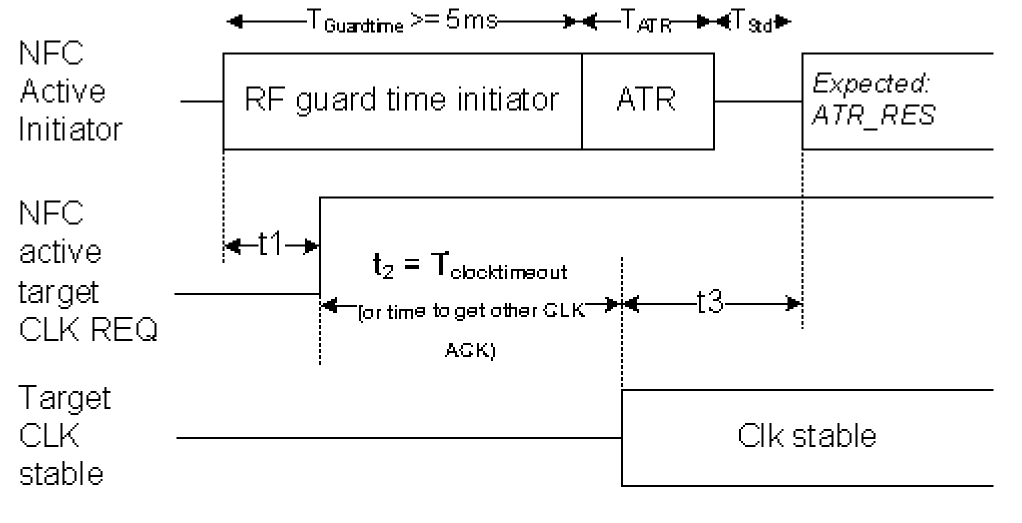
Step1: PN544 request the clock from the system (for clock request, see previous chapter: 'Clock Request & Release').

Step2: When the system clock is available, Host controller sends the NXP_EVT_CLK_ACK event. PN544 is informed that clock is available & stable.

Note: Whatever the Clock Acknowledge method used, the timeout will ever trigger the acknowledgment. It means PN544 always consider the timeout as the maximum allowed time to acknowledge.

9.4.5 CLK request in NFC active target mode

In NFC active reader mode the clock request usage is illustrated:



(1) ATR_RES has to be in-time to serve an ATR_REQ

Fig 41. NFC CLK request in NFC active target mode

Initiator expects response after:

$$T_{Initiator} = T_{guardtime} + T_{ATR} + T_{Std}$$

$T_{Guardtime}$ is equal or greater 5ms (according ISO specification)

T_{STD} is between 57us and 302us

T_{ATR} is between 510us and 1,7ms (no general bytes, all data rates)

The target is able to respond earliest after:

$$T_{Target} = t1 + t2 + t3$$

t1 is between 100us and 1ms (depending on power state)

t2 depends on the settling time of the clock source and the time-out derived

t3 is between 2,5ms and 4ms

To build a functional system T_{Target} must be smaller than $T_{Initiator}$!

9.5 SWP

The host can manage the secure element connected over SWP through a dedicated gate "SWPMgt".

Table 58. SWP Management Gate

Gate	G_{ID}
SwpMgt gate	'A0'

Using this gate the host is able to:

- Activate/Deactivate SWP link
- Get information on SWP link status

Table 59. SWP Events

Value	Event	Description
'03'	*NXP_EVT_SWP_SWITCH_MODE*	This event is sent by the host to temporary activate/deactivate SWP link. **Warning:** PN544 cannot go back to Standby mode if 'Default mode' is not set. It means that 'Off mode' & 'On mode' will keep PN544 in ActiveBAT.
'04'	*NXP_EVT_SWP_PROTECTED*	This event is sent to the host, when protected mode has been enabled (see related *NXP_SWP_PROTECTION_MODE* registry entry), to inform a RF-SWP transaction has been blocked.

The *NXP_EVT_SWP_SWITCH_MODE* event has parameters as follows

Table 60. SWP Switch Mode Event

Description	Length
Mode	1

- Mode indicates the mode to switch to:

Table 61. SWP Switch Mode Event Parameters

Value	Description
0x00	Off (deactivated)
0x01	Default mode
0x02	On (activated)
others	RFU

Table 62. SWP Registry

Id	Name	Access Rights	Comment	Length	Default
‘01’	*NXP_SWP_DEFAULTMODE*	RW (EE)	Indicates if the SWP link has to be enabled when required (only when external field is present). 0x00 -> Off (never activated) 0x01 -> On (activated when required) Others values are RFU	*1*	*0x00*
‘02’	*NXP_SWP_STATUS*	RO (RAM[5])	Indicates the current status of the SWP link(see below state chart): 0x00 -> Not connected (activation sequence not performed) 0x01 -> Connecting (activation sequence on-going) 0x02 -> Connected (activation sequence successful) 0x03 -> Connection lost (communication failed after activation) 0x04 -> Disconnecting (deactivation sequence on-going) 0x05 -> Connection failed (activation sequence failed) . Others values are RFU	*1*	*0x00*

[5] ‘RAM’ means that this information is stored in RAM memory -> Volatile (i.e. lost when going in ‘Low power Mode’)

Id	Name	Access Rights	Comment	Length	Default
4. ‘03’	5. *NXP_SWP_PROTECTION_MODE*	6. RW 7. (Reg[6])	8. Indicates if the protected mode is enabled (see below a description of the protection mode feature) : 9. 0x00 -> Disabled 10. 0x01 -> Enabled Others values are RFU	*1*	*0x00*

Information of a transaction occurs with an application located in the UICC is shared through the ETSI HCI standard [1] “triggering proactivity events” mechanism.

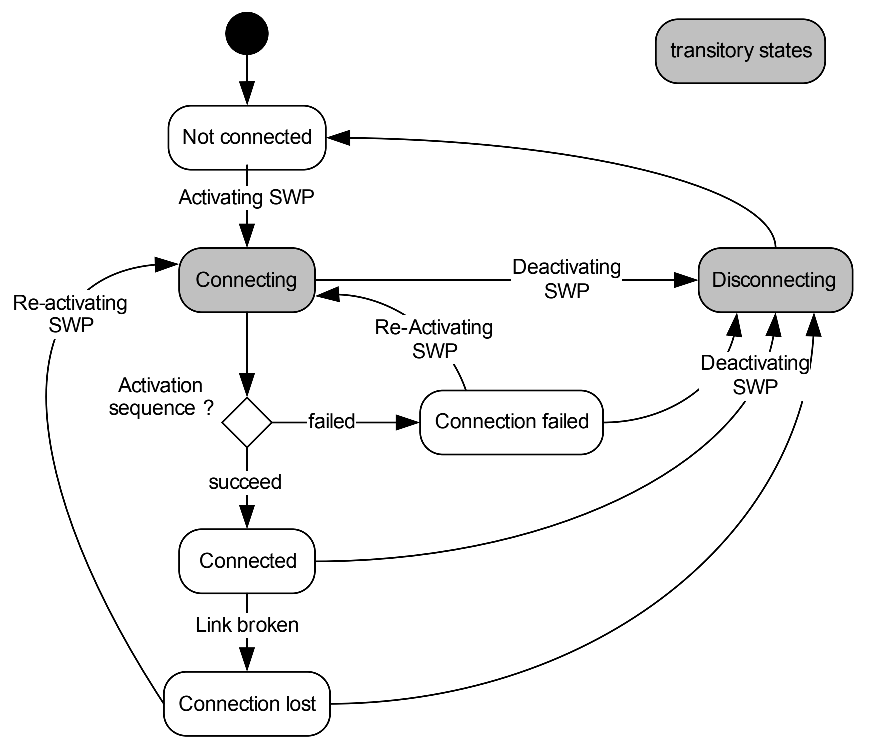


Fig 42. SWP states diagram

[6] ‘Reg’ means that this information is stored in HW register -> Persistent while IC is powered ON (and no VEN reset occurs)

9.5.1 Configuration of SWP link

The SWP registry is stored in EEPROM (except NXP_SWP_STATUS (Id ‘02’), which is a “read only” parameter stored in RAM).

9.5.1.1 Enabling SWP link

By default the SWP link is disabled. The host can enable it using NXP_SWP_DEFAULTMODE (Id ‘01’) of the registry.

9.5.1.2 Configuring Request Power pin

PN544 may want to communicate with the UICC connected via SWP, while the UICC is in low power mode. Consequently, PN544 must send an event to the host to indicate it shall switch the UICC to full power. This event will be automatically sent by the PN544. The pin used to transmit the power request can be CLKREQ pin (GPIO pin 2) or PWR_REQUEST pin (GPIO pin 3). The configuration is done by setting the *SwpMgt_Request_Power* EEPROM area. (See ***‘*SWP configuration*’*** chapter)

9.5.1.3 Changing the baudrate

The default baud-rate used is 848 kbit/s. It may be changed by setting the *SWP_Bitrate* EEPROM area. (See ***‘*SWP configuration*’*** chapter)

9.5.1.4 Powering the UICC when Vbat < Vbat critical

By default, the UICC is not powered when the PN544 is in low power mode. The host can configure it to be powered by setting the *SE_Conf (bit 4-5)* EEPROM area. (See **‘SE configuration*’*** chapter)

9.5.1.5 Managing UICC rights in card emulation mode and in reader mode

The host can allow or forbid, the UICC to use some RF technologies, in both reader mode and card emulation mode.

By default, the UICC cannot use any RF technologies of the PN544. The host can give the UICC the right to use some RF technologies, by setting the *UICC_GateList* EEPROM area. (See ***‘*SWP configuration*’*** chapter).

In card emulation mode, if a RF technology is disabled, e.g. Type A, a Type A external reader will not receive any answer from the UICC.

In reader mode, if a RF technology is disabled, e.g. Type A, the UICC is not allowed to behave as a Type A reader.

Note: *UICC_GateList is* taken into account at SWP activation level (when UICC creates pipes). It means that Host has to setup these registries before enabling the SWP link using ***NXP_SWP_DEFAULTMODE***.

Note: If Host wants to enable these registries change, it has to force the UICC re-initialization, by setting the *UICC_AdminSessionId EEPROM* area to its default value. Host has also to take care of UICC power reset to ensure the settings change. This command has to be sent before ***NXP_SWP_DEFAULTMODE*** use for enabling SWP link.

9.5.1.6 Reading SWP status

SWP status can be read, using ***NXP_SWP_STATUS Id '02'*** of the registry.

See ***Fig 42*** for SWP state transition.

9.5.2 Examples of communication with the UICC connected via SWP

9.5.2.1 Card emulation use case

The host shall set the parameters of the **SwpMgt** gate as described in the previous paragraph.

The UICC shall behave as described in HCI specification.

- The UICC creates and open the pipe between the card gates and card RF gates.
- The UICC activates one or more RF technology. UICC updates the parameters of the corresponding registry.
  - Example for RF technology Type A: The UICC updates UID, SAK, ATQA, APPLICATION_DATA, FWI parameters. Then it sets MODE '02' (i.e. enable). The PN544 is then able to handle the initialization and anti-collision phase by itself.

UICC Type A Card Emulation Example

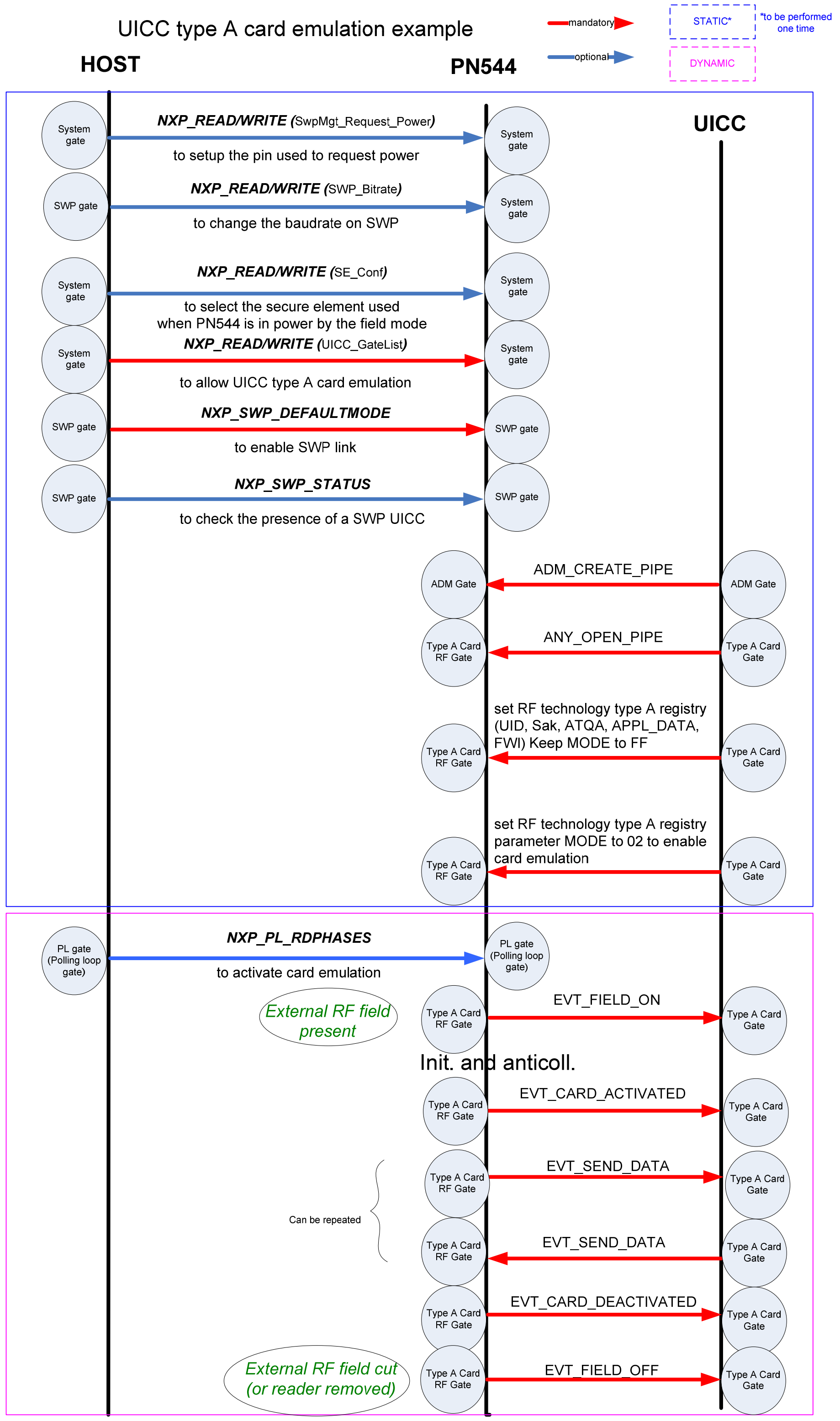


Fig 43. UICC Type A Card Emulation Example

9.5.2.2 Reader use case

The host shall set the parameters of the **SwpMgt** gate as described in the previous paragraph.

The UICC shall behave as described in HCI specification.

- The UICC creates and open the pipe between the reader gates and reader RF gates.

The host must use the command Polling Loop to start the reader mode. (To be detailed in **'Polling Loop'** paragraph).

UICC Reader Type A Example

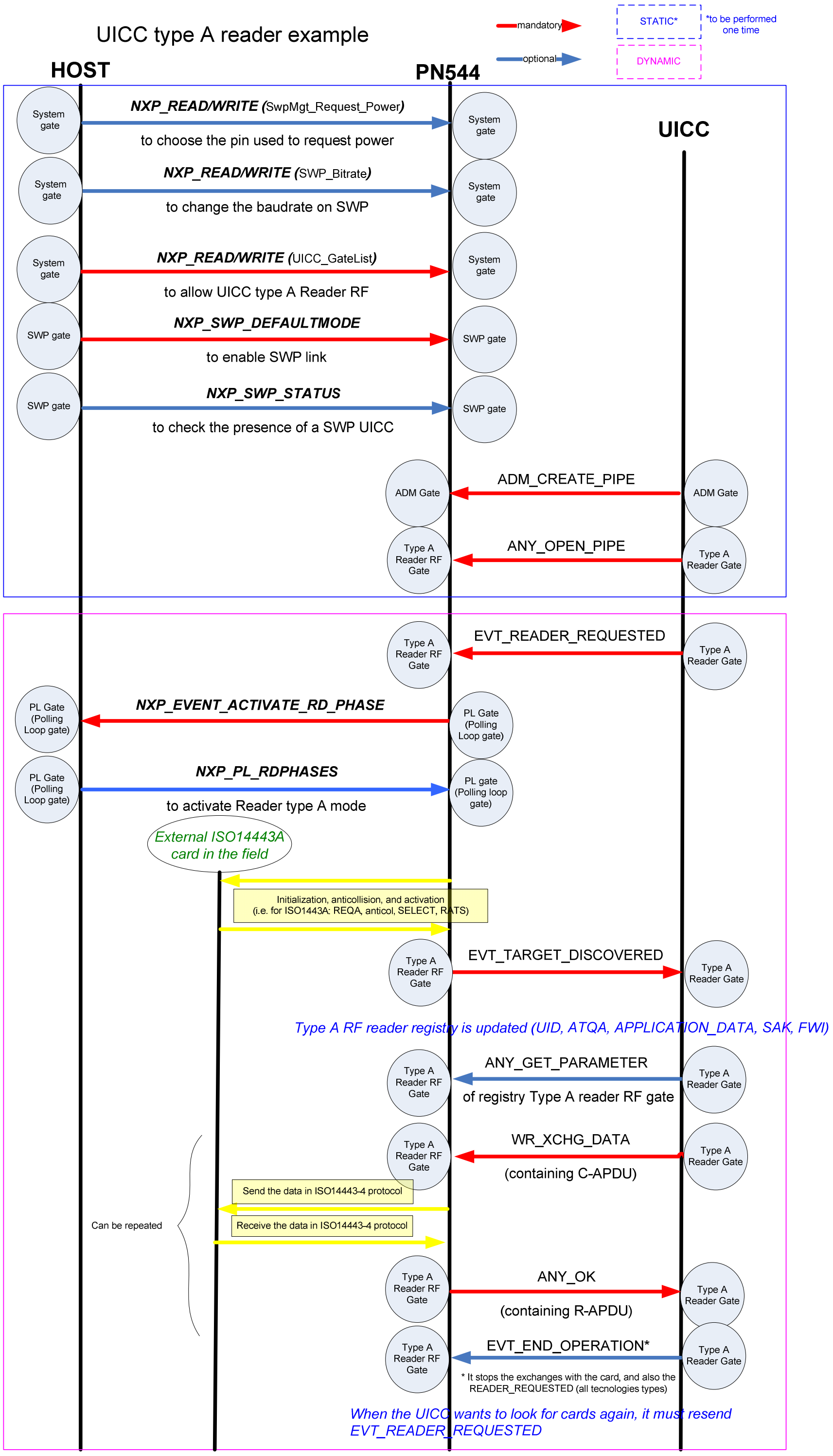

UICC type A reader example
mandatory
optional
STATIC*
*to be performed one time
DYNAMIC
HOST
PN544
UICC
System gate
NXP_READ/WRITE (SwpMgt_Request_Power)
to choose the pin used to request power
System gate
System gate
NXP_READ/WRITE (SWP_Bitrate)
to change the baudrate on SWP
System gate
System gate
NXP_READ/WRITE (UICC_GateList)
to allow UICC type A Reader RF
System gate
SWP gate
NXP_SWP_DEFAULTMODE
to enable SWP link
SWP gate
SWP gate
NXP_SWP_STATUS
to check the presence of a SWP UICC
SWP gate
ADM Gate
ADM_CREATE_PIPE
ADM Gate
Type A Reader RF Gate
ANY_OPEN_PIPE
Type A Reader Gate
Type A Reader RF Gate
EVT_READER_REQUESTED
Type A Reader Gate
PL Gate (Polling Loop gate)
NXP_EVENT_ACTIVATE_RD_PHASE
PL Gate (Polling Loop gate)
PL Gate (Polling Loop gate)
NXP_PL_RDPHASES
to activate Reader type A mode
PL gate (Polling loop gate)
External ISO14443A card in the field
Initialization, anticollision, and activation
(i.e. for ISO1443A: REQA, anticol, SELECT, RATS)
Type A Reader RF Gate
EVT_TARGET_DISCOVERED
Type A Reader Gate
Type A RF reader registry is updated (UID, ATQA, APPLICATION_DATA, SAK, FWI)
Type A Reader RF Gate
ANY_GET_PARAMETER
of registry Type A reader RF gate
Type A Reader Gate
Type A Reader RF Gate
WR_XCHG_DATA
(containing C-APDU)
Type A Reader Gate
Send the data in ISO14443-4 protocol
Can be repeated
Receive the data in ISO14443-4 protocol
Type A Reader RF Gate
ANY_OK
(containing R-APDU)
Type A Reader Gate
Type A Reader RF Gate
EVT_END_OPERATION*
Type A Reader Gate
* It stops the exchanges with the card, and also the READER_REQUESTED (all tecnologies types)
When the UICC wants to look for cards again, it must resend EVT_READER_REQUESTED

Fig 44. UICC Type A Reader Example

SWP Protected Mode

Protected mode is a feature which intends to be used by the host to block any transaction between an external RF reader and the UICC. Then receiving the notification of the UICC activation, the host can authorize UICC access through SWP.

The following flow diagram describes an example of use of this feature:

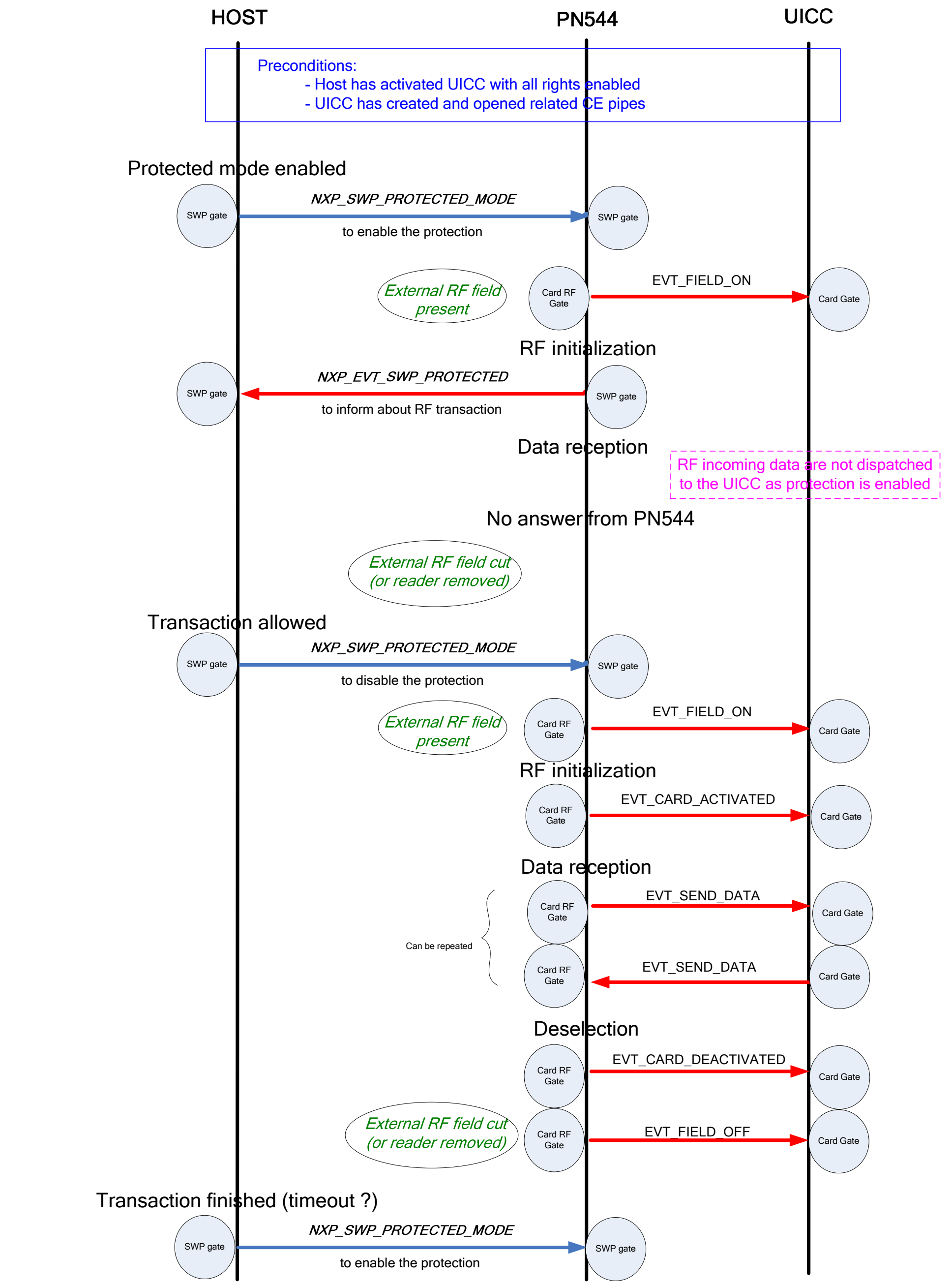


Fig 45. Protected mode flow example

9.5.3 UICC dependency for PAYPASS Compliance

In order to fit with PAYPASS requirements, the following registry has to be correctly set by UICC:

Table 63. Type A Card RF Registry for PAYPASS

Id	Name	Access Right	Comment	Length	Recommended Value
'09'	*DATARATE_MAX*	RW	Maximum data rate supported (Restrict to 106 Baud-rate)	*3*	*0x00 00 01*
'06'	*FWI*	RW	Frame waiting time as defined in ISO/IEC 14443-4 for type A	*1*	*0x07*

Table 64. Type B Card RF Registry for PAYPASS

Id	Name	Access Right	Comment	Length	Recommended Value
'06'	*DATARATE_MAX*	RW	Maximum data rate supported (Restrict to 106 Baud-rate)	*3*	*0x00 00 01*
'04'	*ATQB ->FWI(*)*	RW	Frame waiting time as defined in ISO/IEC 14443-3 for Type B	*1*	*0x07*

(*): For Type B Card RF, *FWI* is part of *PROTO_INFO* byte. *PROTO_INFO* is the 4th byte of *ATQB* bytes.

9.6 Polling Loop

The host could manage the polling loop mechanism through a dedicated gate "PollingLoopMgt".

Table 65. Polling Loop Management Gate

Gate	G_{ID}
PollingLoopMgt gate	'94'

Using this gate the host is able to:

- Configure the polling loop
- Manage clock request/acknowledge via host commands
- Receive information about UICC reader activities

Table 66. Polling Loop events

Value	Event	Description
'01'	*NXP_EVT_CLK_ACK*	This event is sent by the host to acknowledge the clock request (if *PlClockAck* EEPROM area has been previously set to 2)
'02'	*NXP_EVT_CLK_REQUEST*	This event is sent to the host to inform that the input clock is requested (if *PlClockRequest* EEPROM area has been previously set to 2)
'03'	*NXP_EVT_ACTIVATE_RDPHASE*	This event is sent to the host to inform that the SE is requesting for the activation of the reader phases in the polling loop (see NXP_PL_RDPHASES registry entry).
'04'	*NXP_EVT_DEACTIVATE_RDPHASE*	This event is sent to the host to inform that the SE is no more requesting for the activation of the reader phases in the polling loop (see NXP_PL_RDPHASES registry entry).

The *NXP_EVT_CLK_REQUEST* has one parameter:

Table 67. NXP_EVT_CLK_REQUEST parameter

Description	Length
Status	1

Table 68. Status values

Value	Description
0x00	Clock request
0x01	Clock release
Others	RFU

The *NXP_EVT_ACTIVATE_RDPHASE* event has parameters as follows:

Table 69. NXP_EVT_ACTIVATE_RDPHASE Event Parameters

Description	Length
Technology	1

- **Technology** indicates the Reader phase requested

Table 70. Technology Description

Bit	Technology
0	Detection type A
1	Detection type B
others	RFU

0 -> technology not requested

1 -> technology requested

Table 71. Polling Loop Registry

<table>
<tr><th>Id</th><th>Name</th><th>Access Rights</th><th>Comment</th><th>Length</th><th>Default</th></tr>
<tr><td>‘06’</td><td>NXP_PL_RDPHASES</td><td>RW
(Reg)</td><td>Indicates the status of the phases:
<table>
<tr><td>Bit</td><td>Technology</td></tr>
<tr><td>0</td><td>Detection type A</td></tr>
<tr><td>1</td><td>Detection type B</td></tr>
<tr><td>2</td><td>Detection type F 212</td></tr>
<tr><td>3</td><td>Detection type F 424</td></tr>
<tr><td>4</td><td>Detection ISO15693</td></tr>
<tr><td>5</td><td>Detection NFC active</td></tr>
<tr><td>6</td><td>RFU</td></tr>
<tr><td>7</td><td>Detection Pause(*)</td></tr>
</table>
0 -> disabled
1 -> enabled
(*)When Pause phase is enabled, it implicitly replaces the Emulation phase (setting: NXP_PL_PAUSE)</td><td>1</td><td>0x00</td></tr>
<tr><td>‘07’</td><td>NXP_PL_EMULATION</td><td>RW
(EE)</td><td>Indicates the Emulation phase duration.
(from 0 to 3.145s in 48µs step)</td><td>2</td><td>0x5161
(1s)</td></tr>
<tr><td>‘08’</td><td>NXP_PL_PAUSE</td><td>RW
(EE)</td><td>Indicates the Pause phase duration.
(from 0 to 3.145s in 48µs step)</td><td>2</td><td>0x0824
(100ms)</td></tr>
<tr><td>‘09’</td><td>NXP_PL_NFCT_DEACTIVATED</td><td>RW
(Reg)</td><td>Disable all NFC Target feature (passive/active/all baudrates)
0x00 : NFC Target Phase Activated
0x01 : NFC Target Phase deactivated
Other values unsupported</td><td>1</td><td>0x00</td></tr>
</table>

9.6.1 Clock

Refer to '**Clock Management 9.4'** paragraph for details.

It includes an explanation on the usage of the events:

- NXP_EVT_CLK_ACK
- NXP_EVT_CLK_REQUEST

And an explanation on the usage of the EEPROM data area:

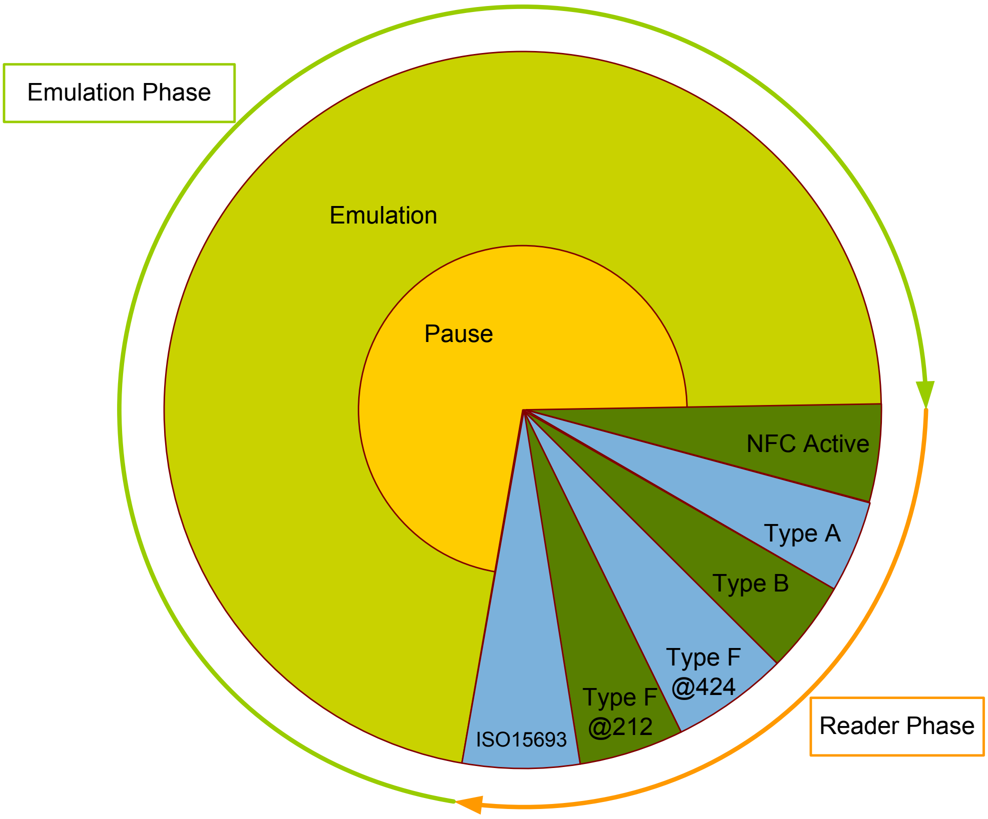
- PlClockRequest
- PlClockAck
- PlClockTimeout
- FRAC_ClkSel
- FRAC4_DIV
- FRAC4_OOF0
- FRAC4_OOF1
- FRAC4_OOF2
- FRAC4_CAL0
- FRAC4_CAL1

9.6.2 PollingLoop Management

At start up, the PN544 is in a “default” mode (The Power level of this mode is set according to *PWR_STATUS* EEPROM area value). It waits for a command from the host, or from an external reader if the RF level detector has been enabled (in that case Refer to SWP & NFC-WI setup).

It stays infinitely in this state (until a command from the host is received).

The Polling Loop concept is a sequence of different phases: mainly Reader phases & Card Emulation phases. The following figure shows a possible setup of PN544:

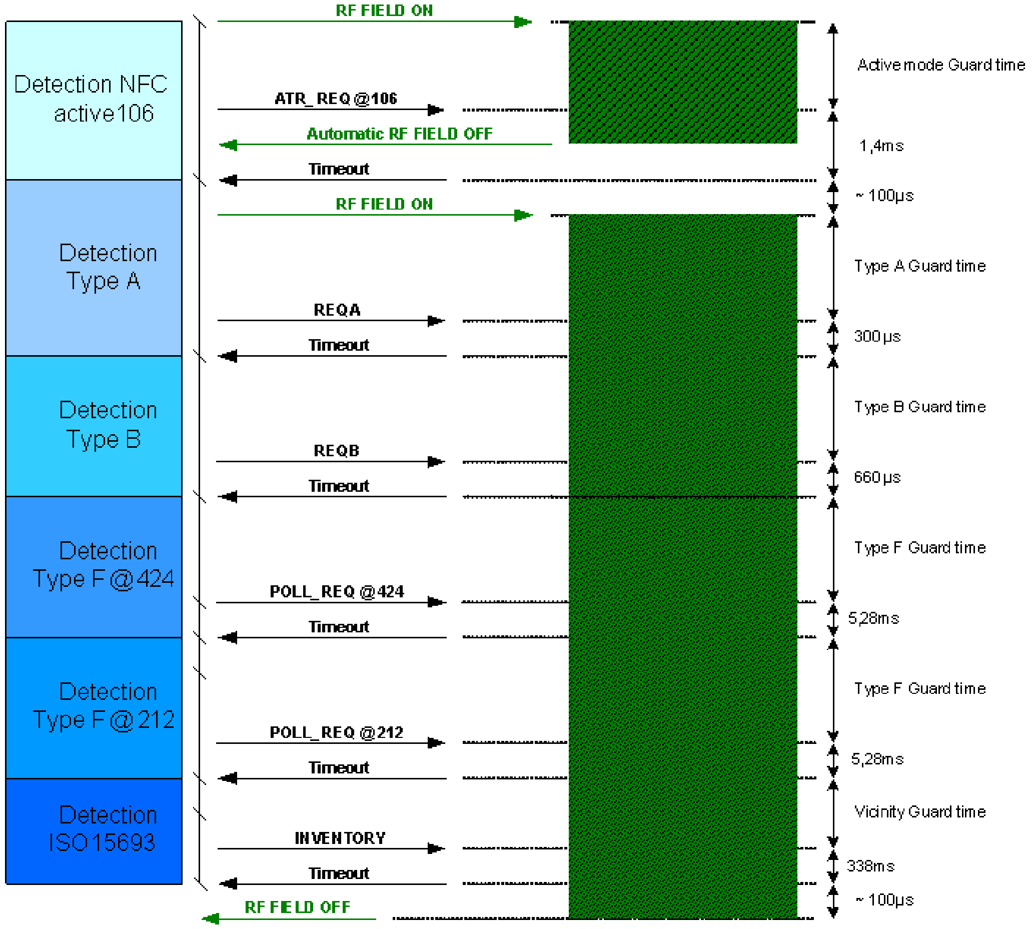


(1)

Fig 46. Polling Loop Overview

9.6.2.1 Detection Guard Time

Polling Loop timings for detection are set as follows:



(1)

Fig 47. Polling Loop Timings

All Guard Times are set with respect to the technology specification. However, these timings can be changed (see ***‘0’*** chapter).

The EEPROM areas used for Guard time settings are: *PlMgtPassiveGTA, PlMgtPassiveGTB, PlMgtPassiveGTF, PlMgtActiveGTA and PlMgtVicinityGT.*

9.6.2.2 Activation Timeout for NFC-IP1 Initiator

When NFC-IP1 Initiator mode is enabled, PN544 is not able to detect absence of NFC-IP1 Target.

After the ATR_REQ has been sent, it waits the '*NFCI_ATR timeout*' for ATR_RES before starting next phase of Polling Loop.

This timeout value can be changed by setting the *NFCI_ATR_TOMSB/LSB* EEPROM area. (See **'**

HW configuration**'** chapter).

9.6.2.3 Activation Timeout for ISO14443-PICC & NFC-IP1 Target

ISO14443 Activation (fro RF Field ON up to RATS or ATTRIB) or NFC-IP1 Target Activation (From RF Field ON up to ATR_REQ) must be completed in less than '*TGINIT_GUARD timeout'* otherwise the ongoing RF initialization is aborted.

(No impact on use cases as external readers/initiators perform activation in short sequence).

This timeout value can be changed by setting the *TGINIT_GUARD_TO_MSB & TGINIT_GUARD_TO_LSB* EEPROM area (see **'**

HW configuration**'** chapter).

9.6.2.4 NXP_PL_RDPHASES parameter

To setup and launch the Polling Loop mechanism, the host has to use the *NXP_PL_RDPHASES* registry*.* This registry allows choosing the desired type of communication.

NXP_PL_RDPHASES registry contains:

- One bit that enables/disables the “Pause” mode (PN544 is in a non active state). To enable the Card Emulation mode (i.e. the host or the UICC connected over SWP or a SmartMX connected over NFC-WI, acting as a contactless card), the bit 7 of *NXP_PL_RDPHASES* must be set to 0 (“Pause” disabled).
- Five bits that enable/disable the reader mode. There are different bits for the different supported RF types:
  - *Detection type A*: set to 1, this bit allows the host to behave as an ISO14443A reader (ISO14443-4 or MIFARE or JEWEL) and as an NFC-IP1 Initiator.
  - *Detection type B*: set to 1, this bit allows the host to behave as an ISO14443B reader. The method used is the probabilistic approach.
  - *Detection type F 212/424:* set to 1, these bits allow the host to behave as a FeliCa reader and as an NFC-IP1 Initiator.
  - *Detection NFC Active:* set to 1*,* this bit allows the host to behave as an NFC-IP1 Initiator in Active mode.
  - *Detection ISO15693:* set to 1*,* this bit allows the host to behave as an ISO15693 reader.

NXP_PL_RDPHASES bits don’t start the desired mode. These are only “enable”.

NXP_PL_RDPHASES shall be used in addition to HCI commands (see following examples).

The host must use *NXP_PL_RDPHASES,* to give itself the right to start reader and emulation modes, and/or to control the UICC rights to behave as a reader.

Note: The Polling sequence is always as follows: NFC Active, Type A reader, Type B reader, FeliCa reader 424, FeliCa reader 212, ISO15693 reader and then Card Emulation (or Pause). Of course, each phase can be disabled, but the sequence is always the same.

Note: *NXP_PL_RDPHASES* registry setup is lost after a Hardware reset (VEN Low) and the default value of this registry is recovered.

9.6.2.5 NXP_PL_EMULATION, NXP_PL_PAUSE parameters

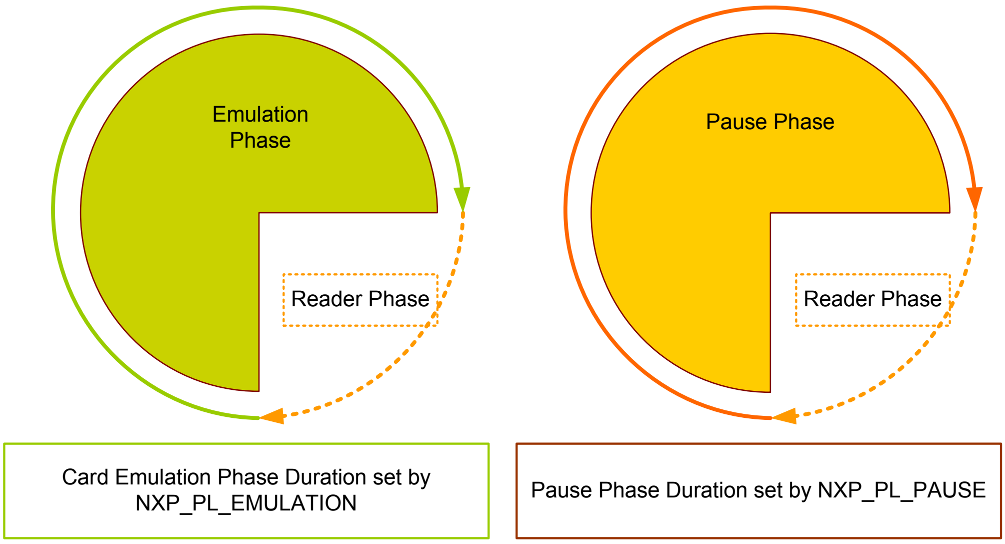
NXP_PL_EMULATION and *NXP_PL_PAUSE* timings have sense only when a reader sequence has been started by *NXP_PL_RDPHASES.*

As the Polling Loop is running, Emulation & Pause modes have a different meaning:

Emulation mode means *card emulation* mode (either from Host or UICC or SmartMX) as well as NFC Target. In this mode, the PN544 is seen as a contact less card and as a NFC Target. For details on UICC and SmartMX setup, refer to SWP & NFC-WI chapters.

Pause mode means *no action*; PN544 does not perform any action.

Note: Refer to *PWR_STATUS* (See ***'0*** chapter*)* to set the default power mode of Emulation or Pause phases.


Emulation Phase
Reader Phase
Pause Phase
Reader Phase
Card Emulation Phase Duration set by NXP_PL_EMULATION
Pause Phase Duration set by NXP_PL_PAUSE

(1)

Fig 48. Emulation or Pause Phase

Here is a view of the different Polling Loop possibilities:

No phase started:

If no sequence (neither reader nor emulation) has been started, the PN544 stays infinitely in the default power mode (Refer to *PWR_STATUS* EEPROM area.). In such a mode, the Emulation can be enabled according to SWP and NFC-WI.

One Emulation phase only:

If no reader phase has been enabled nor pause phase, the emulation phase will be infinite. As long as there is no external reader, the PN544 is in Card Emulation mode, with RF level detector active. To save power, the default power mode should be set to *standby*.

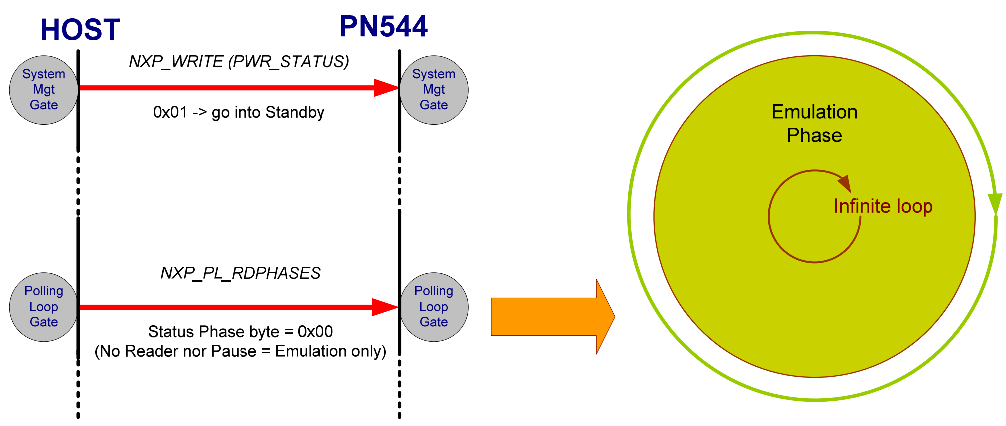


Fig 49. Emulation Phase only

Note: From a functional point of view, this mode is equivalent to the previous mode: ‘No phase started’.

One Pause phase only:

If only a Pause phase is enabled, PN544 can stay in an inactive mode (no Reader action, no Card Emulation). To save power, the default power mode should be set to *standby*.

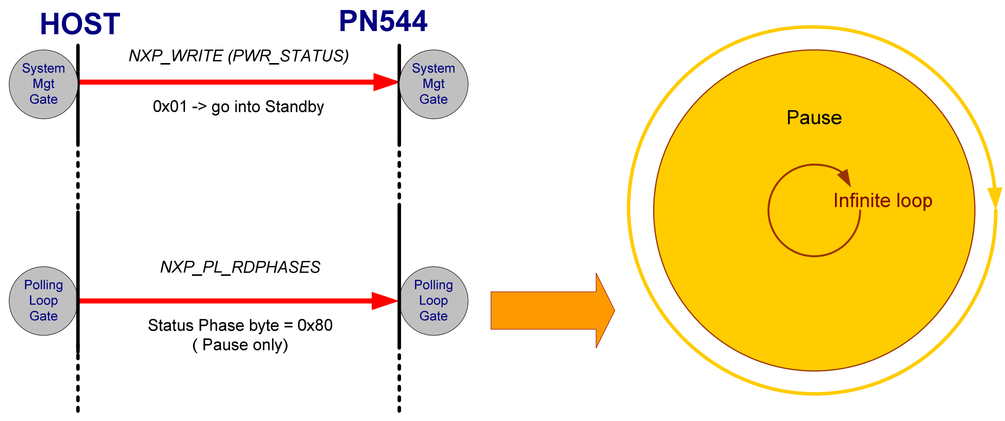


Fig 50. Pause Phase only

Note: Such a setup can be useful to disable the Card Emulation functionality without any access to SWP or NFC-WI registry.

One reader phase, with one RF type only:

If a reader phase is started, the duration of it is fixed (depending on RF type). Then PN544 waits, during *NXP_PL_PAUSE* time. Then it starts again the reader phase, and so on (until a card is detected).

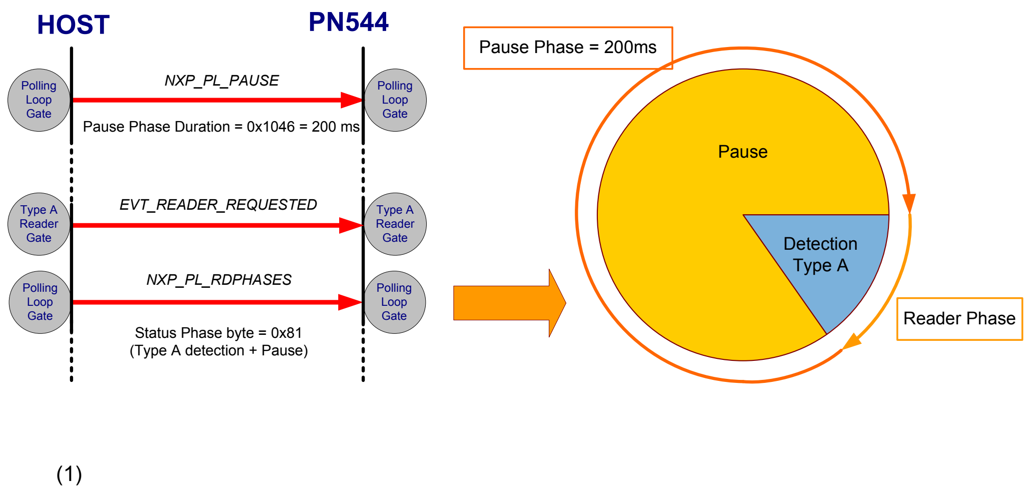


Fig 51. Reader Phase only

In this example, the host wants to act as an ISO14443A reader.

After correct initialization, as shown in Fig 51, the PN544 will regularly look for ISO14443A cards in the field.

If no card answers, PN544 goes then into a PAUSE phase, during a duration defined in the registry *NXP_PL_PAUSE.* The PAUSE phase power mode is the one set according to *PWR_STATUS* EEPROM area value.

Note: During the reader phases, PN544 can ask for the system clock. It is not shown on the figure (for detail on clock request, refer to Clock Management chapter).

One reader phase, with several RF types:

PN544 goes through all the reader phases, then, it waits, during *NXP_PL_PAUSE* time. Then it starts again the reader phases, and so on (until a card is detected).

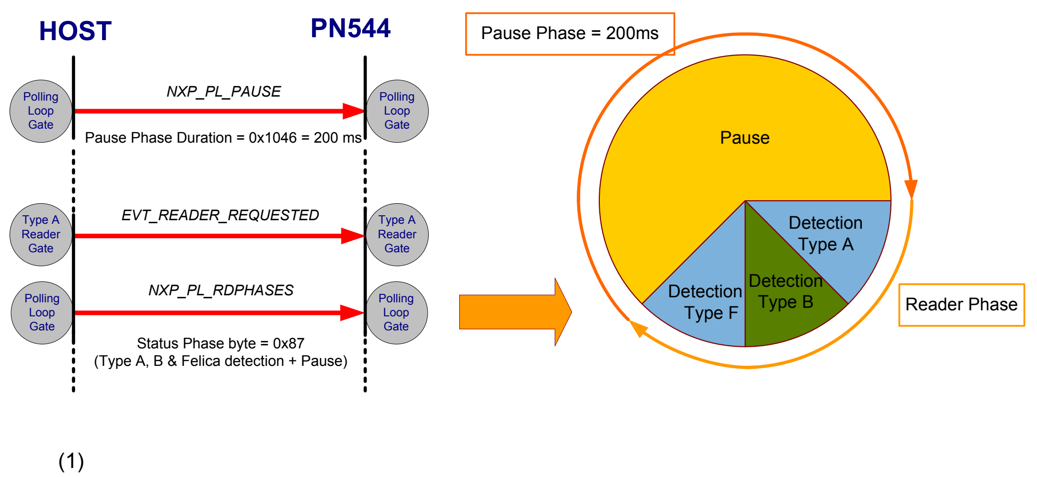


Fig 52. Several Reader Phases

Note: it is assumed that the RF Gate initialization has been performed. It means that to launch the Polling Loop, the *EVT_READER_REQUESTED* & *NXP_PL_RDPHASES* are only needed to start the reader's phases.

Reader phase(s) and emulation phase

PN544 goes through all the reader phases, then, it goes to emulation phase, during *NXP_PL_EMULATION* time. Then it starts again the reader phases, and so on (until a card or an external reader is detected). The EMULATION phase power mode is the one set according to *PWR_STATUS* EEPROM Area value.

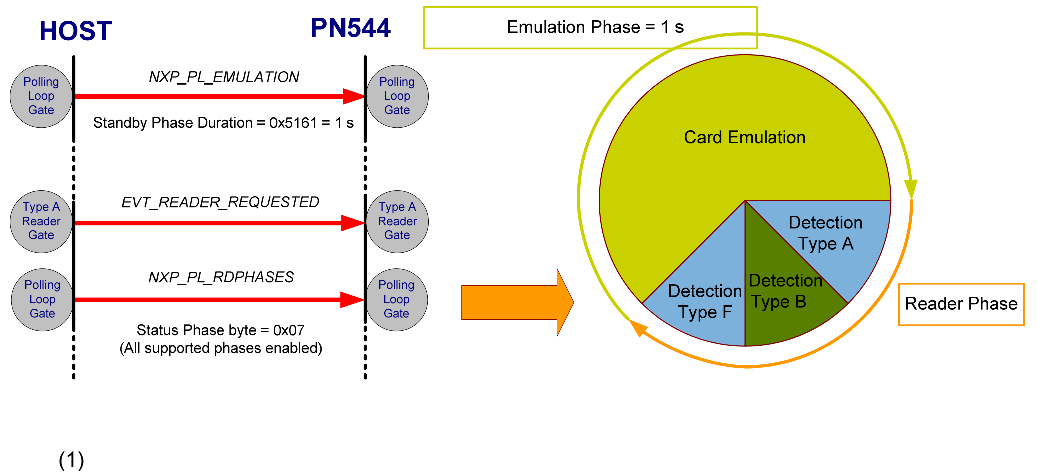


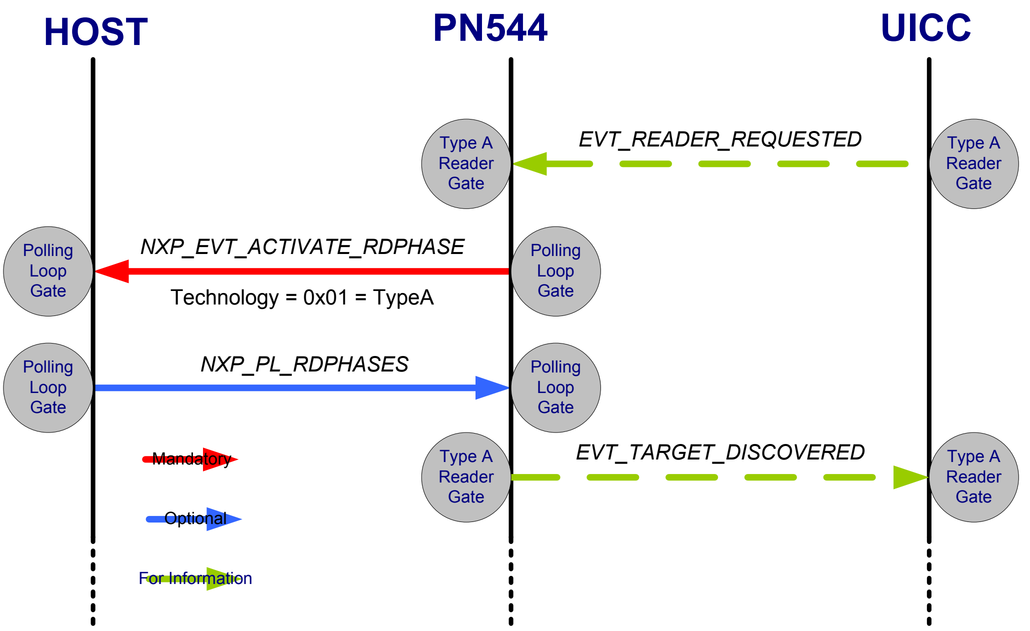
Fig 53. Reader Phases and Emulation Phase

9.6.2.6 NXP_EVT_ACTIVATE_RDPHASE, NXP_EVT_DEACTIVATE_RDPHASE events

These events are only used in case of UICC wants to act as a reader.

When the UICC wants to start a reader RF gate, the PN544 sends a *NXP_EVT_ACTIVATE_RDPHASE* to the host.

The host may in response enable the Reader mode by sending the NXP*_PL_RDPHASES.*



(1) Note: In this example there is no reader activated on Host side, EVT_TARGET_DISCOVERED is sent directly to UICC.

Fig 54. Activate Reader Phase Event (Type A example)

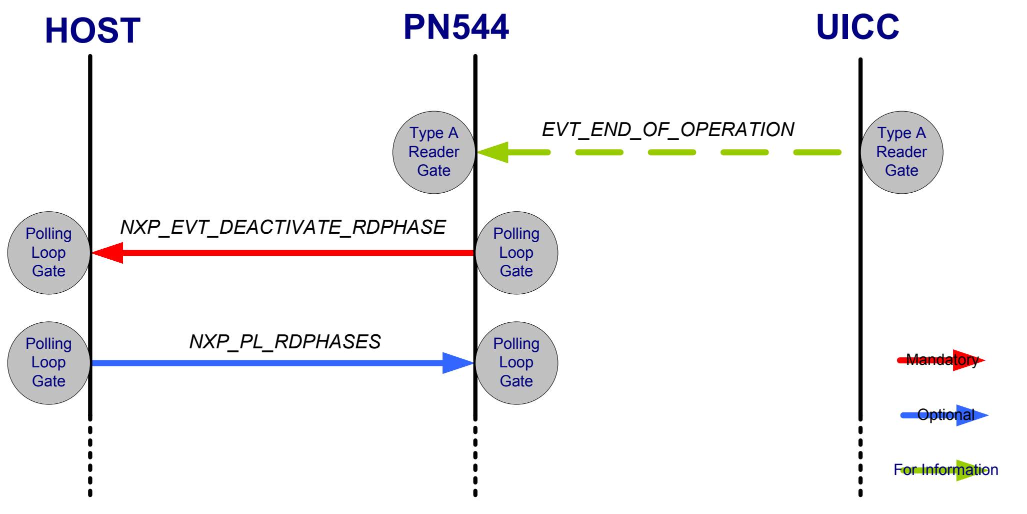
To accept the reader request, when receiving the *NXP_EVT_ACTIVATE_RDPHASE* event, the host can check the RF Reader Card technology requested (Technology parameter). To enable it, the Host must send the *NXP_PL_RDPHASES* with at least the technology requested enabled (cf. SWP Management chapter for details on technology supported over SWP).

To reject the request from the UICC, the host can decide not to send the *NXP_PL_RDPHASES* to disable the reader request.

Note: This functionality is mainly for a power consumption management. The host can decide to enable the reader mode only if enough power is available.

At the end of the reader operation, PN544 can inform the host by sending a *NXP_EVT_DEACTIVATE_RDPHASE.*

The host can decide to stop the reader request by sending *NXP_PL_RDPHASES* and disable the RF technology detection.

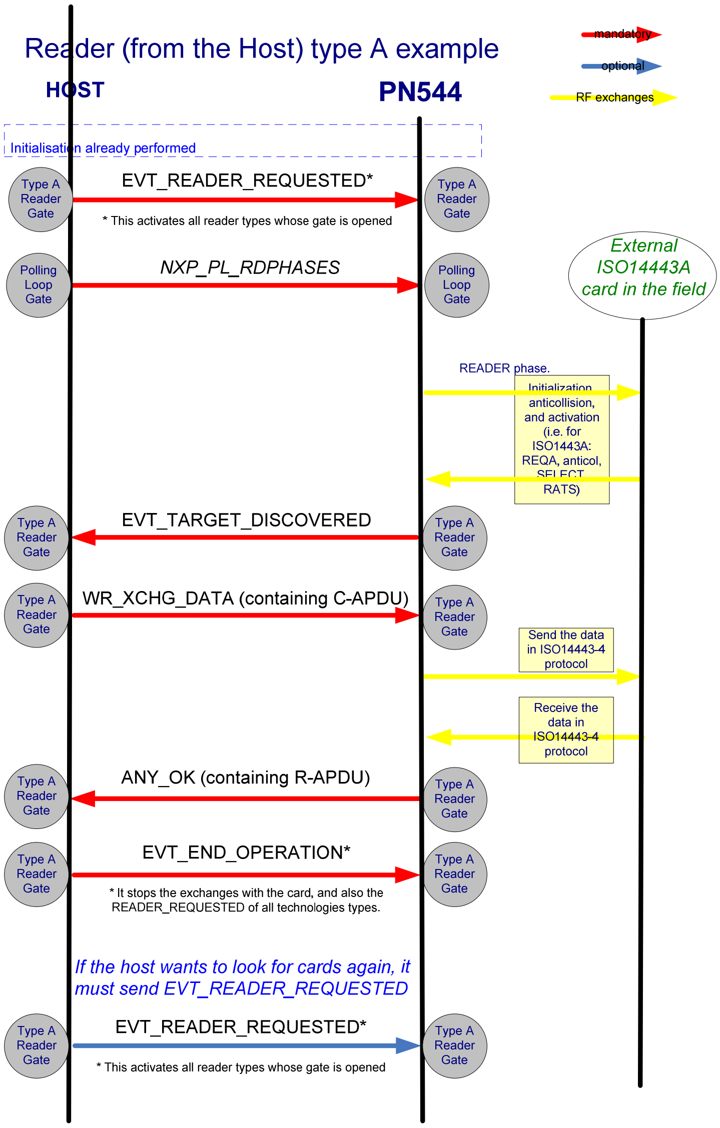


(1)

Fig 55. Deactivate Reader Phase Event (Type A example)

9.6.3 Example of Polling Loop

Here is an example of Polling Loop launch with an ISO14443 type A card detected:


Reader (from the Host) type A example
HOST
PN544
mandatory
optional
RF exchanges
Initialisation already performed
Type A Reader Gate
EVT_READER_REQUESTED*
* This activates all reader types whose gate is opened
Polling Loop Gate
NXP_PL_RDPHASES
External ISO14443A card in the field
READER phase.
Initialization anticollision, and activation (i.e. for ISO1443A: REQA, anticol, SELECT, RATS)
EVT_TARGET_DISCOVERED
WR_XCHG_DATA (containing C-APDU)
Send the data in ISO14443-4 protocol
Receive the data in ISO14443-4 protocol
ANY_OK (containing R-APDU)
EVT_END_OPERATION*
* It stops the exchanges with the card, and also the READER_REQUESTED of all technologies types.
If the host wants to look for cards again, it must send EVT_READER_REQUESTED
EVT_READER_REQUESTED*
* This activates all reader types whose gate is opened

(2)

Fig 56. Polling Loop with ISO14443 Type A card detected

Notes: If the Host wants to restart the Polling Loop, it has to send the *EVT_READER_REQUESTED* event.

If the Host wants to temporary stop the Reader phase it can simply send the *NXP_PL_RDPHASES* and disable the RF technology detection.

9.6.4 NFC-IP1 in Polling Loop

If NFC-IP1 Initiator gate is setup (Pipe Opened, and *NXP_NFCI_MODE* registry set with at least one baudrate). The NFC-IP1 Initiator mode will be executed during the 'Reader phase' of the Polling Loop.

If NFC-IP1 Target gate is setup (Pipe Opened, and *NXP_NFCT_MODE* registry set with at least one baudrate). The NFC-IP1 Target mode will be executed during the 'Emulation phase' of the Polling Loop.

9.6.5 Multi Secure Element

In the case of Card Emulation over SWP/UICC and NFC-WI enable, PN544 cannot present multiple cards to the external PCD. It is up to the host to manage the Secure Element enabled.

At application level, host can decide to enable or disable each link.

For SWP/UICC, host can use the *NXP_EVT_SWP_SWITCH_MODE* event to disable/enable the link*.*

For NFC-WI, the use of *NXP_EVT_SE_SWITCH_MODE* event is recommended to temporary switch the NFC-WI Secure Element in 'Off mode'.

9.6.6 RF Configuration

9.6.6.1 Card Emulation RF Step

Around 45us after RF ON of a reader, due to an internal reconfiguration, PN544 as card or passive target can impact the reader RF field for around 5us.

The amplitude of the RF decreased will depend on antenna tuning of both sides and of their coupling factor. There is no issue for passing certification.

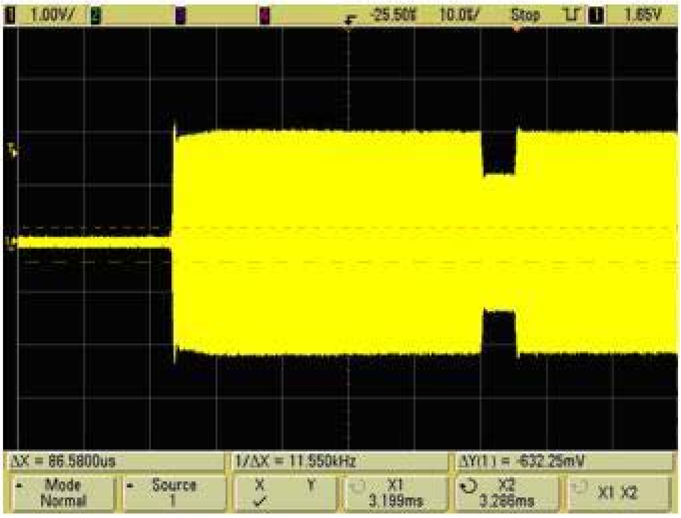


Fig 57. Card Emulation RF step

9.6.6.2 Type B, 212kbps and 424kbps Passive Initiator and Active Initiator/Target

When TVDD is not configured to 2.7V, for VBAT higher than:

- TVDD configured + 0.5V when TXLDO offset is used
- TVDD configured otherwise,

Some time after starting RF field, PN544 can have a lower step in RF amplitude for around 2ms. This is due to a routine which optimizes modulation index around 10% modulation depth. This routine is done with TVDD=2.7V.

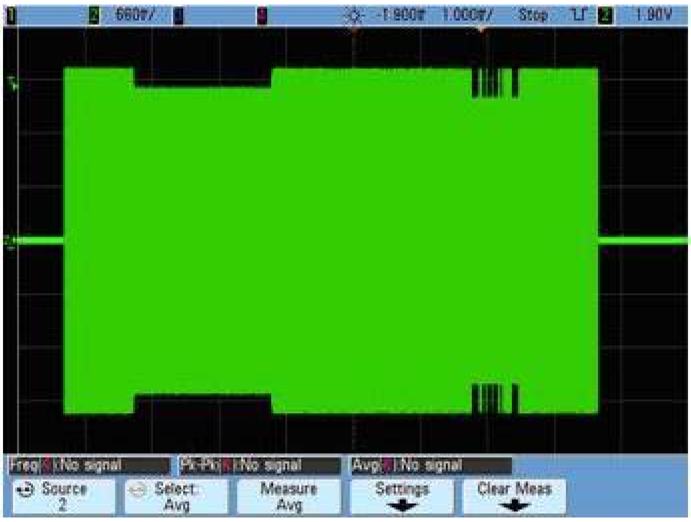


Fig 58. RF Field Lower Step

9.7 NFC-WI

The host could manage the secure element connected over NFC-WI through a dedicated gate “NfcWiMgt”.

Table 72. NfcWiMgt gate

Gate	GID
NfcWiMgt gate	‘A1’

Using this gate the host is able to:

- Manage the secure element connected over NFC-WI

Table 73. NFC-WI Events

Value	Event	Description
‘01’	*NXP_EVT_SE_START_OF_TRANSACTION*	This event is sent to the host to inform that a transaction starts.
‘02’	*NXP_EVT_SE_END_OF_TRANSACTION*	This event is sent to the host to inform that the transaction ends.
‘03’	*NXP_EVT_SE_SWITCH_MODE*	This event is sent by the host to temporary switch NFC-WI mode. **Warning:** PN544 cannot go back to Standby mode if ‘Default mode’ is not set. It means that ‘Wired mode’, ‘Virtual mode’ & ‘Off mode’ will keep PN544 in ActiveBAT.

The *NXP_EVT_SE_SWITCH_MODE* event has parameters as follows

Table 74. NXP_EVT_SE_SWITCH_MODE Event Parameters

Description	Length
Mode	1

- Mode indicates the mode to switch to

Table 75. Mode Description

Value	Description
0x00	‘Wired mode’
0x01	‘Default mode’
0x02	‘Virtual mode’
0x03	‘Off mode’
others	RFU

Note: ‘Default mode’ is the mode set by *NXP_SE_DEFAULTMODE* registry.

Table 76. NFC-WI Registry

<table>
<tr><th>Id</th><th>Name</th><th>Access Rights</th><th>Comment</th><th>Length</th><th>Default</th></tr>
<tr><td>‘01’</td><td><i>NXP_SE_DEFAULTMODE</i></td><td>RW
(EE)</td><td>Indicates the default mode of the SE connected over NFC-WI.
0 -> Off mode
1 -> Virtual mode
Others values are RFU</td><td><i>1</i></td><td><i>0x00</i></td></tr>
<tr><td>‘05’</td><td><i>NXP_SE_EVENTS</i></td><td>RW
(EE)</td><td>Defines which events have to be raised to the host when occurred :
<table>
<tr><td>Bit</td><td>Event</td></tr>
<tr><td>0</td><td>Start of transaction</td></tr>
<tr><td>1</td><td>End of transaction</td></tr>
<tr><td>2 .. 7</td><td>RFU</td></tr>
</table>
0 -> not to be raised
1 -> to be raised</td><td><i>1</i></td><td><i>0x00</i></td></tr>
</table>

Note: Some other NFC-WI settings are available in EEPROM (see **‘SE configuration’** chapter) for *SE_Conf, WI_SE_SigoutSel, WI_SE_SigoutClkSel. NfcWi_SE_SigInDelay, NfcWi_SE_TimeOut_H* and *NfcWi_SE_TimeOut_L.*

There are 3 possible configurations for the NFC-WI Secure Element:

• **Off mode** is used when no communication with the Secure Element is needed, either from the host controller or from an external PCD.

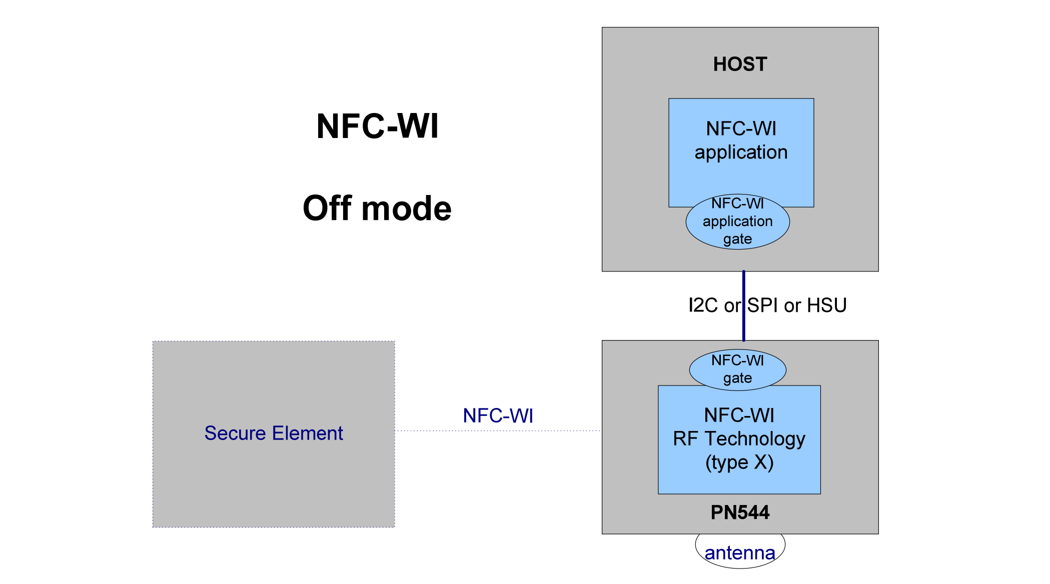


Fig 59. NFC-WI Off Mode

• **Wired mode** is used to communicate with the Secure Element internally.

No RF field is emitted. In this mode, PN544 acts as PCD to access to the Secure Element through the related RF reader gate.

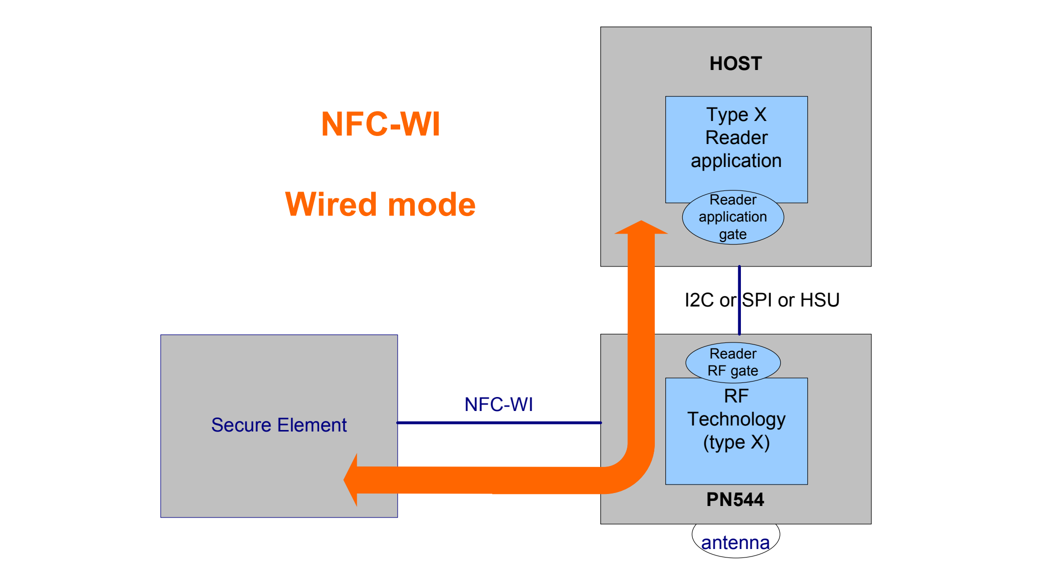


Fig 60. NFC-WI Wired Mode

Note: As soon as NFC-WI is set in WIRED mode, no external tag (through RF) can be detected. Only communication with SE is allowed through the related RF gate.

- **Virtual mode**, Secure Element is seen as a real contact less card.

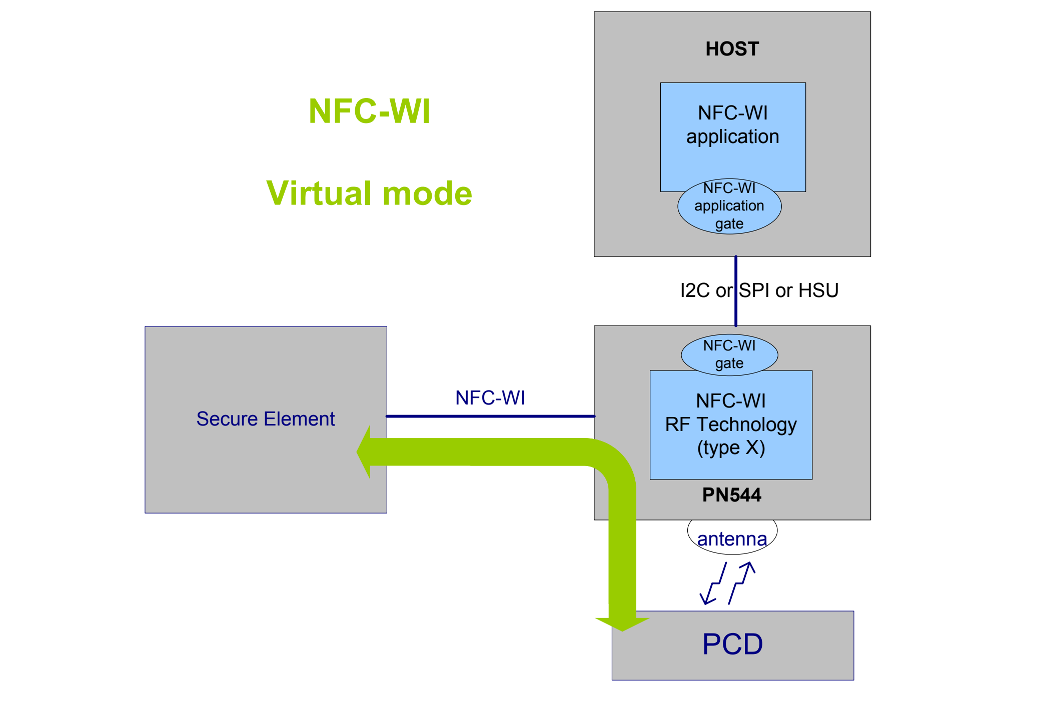


Fig 61. NFC-WI Virtual Mode

Note: NFC-WI Secure Element could be used in PowerByTheField mode. For such mode, Host can configure the *SE_Conf (bit 4-5)* EEPROM area. (See **'SE configuration'** chapter)

9.8 MIFARE PCD

PN544 offers MIFARE PCD functionality through the generic “Type A reader RF gate” of the ETSI HCI specification [1]. Activation is as defined in the specification, and then the MIFARE card (***MIFARE UltraLight*** & ***MIFARE 1K/4K***) can be accessed with the following NXP proprietary command of the “Type A RF reader gate”

Please Note that Mifare documentation uses MSByte first whereas PN544 uses LSByte first convention

9.8.1 NXP_MIFARE_RAW

NOTE: This command is only valid for ***MIFARE UltraLight*** access.

Table 77. NXP_MIFARE_RAW Command

Value	Command	Description
’20’	*NXP_MIFARE_RAW*	This command allows the exchange of raw data between the host and a MIFARE ® card previously activated. Within this command and response, the CRC has to be handled by the host.

This command has the following parameters:

Table 78. NXP_MIFARE_RAW Parameters

Description	Length
TO	1
Status	1
Data	N

- **TO** value specify the timeout to be used with the following formula:
  - Timeout = (256 × 16 / 13.56MHz) × 2 ^ TO (TO = 0 to 14)
- **Status**: bits b0 to b2 indicate information on valid bits in the last data byte (value 0 indicates all are valid, from 1 to 7 it indicated the number of valid bits), other bits are RFU. Only value 0 is handled as no MIFARE command requires not byte align data.
- **Data** is the raw frame to be sent to the card (including CRC)

The response to this command is as follows:

Table 79. NXP_MIFARE_RAW Response parameters

Description	Length
Response code	1
Status	1
Data	N

- **Response code** possible values: any of the ETSI HCI generic response code.
- **Status**: bits b0 to b2 indicate information on valid bits in the last data byte (value 0 indicates all are valid, from 1 to 7 it indicated the number of valid bits), other bits are RFU.
- **Data** is the raw frame received from the card (including CRC).

9.8.2 NXP_MIFARE Commands and Registry

Table 80. NXP_MIFARE_CMD Command

Value	Command	Description
’21’	*NXP_MIFARE_CMD*	This command allows the exchange of data between the host and a MIFARE ® card previously activated. (Refer to [5] for details)

This command has the following parameters:

Table 81. NXP_MIFARE_CMD Parameters

Description	Length
Cmd	1
Addr	1
Data	N

- **Cmd** is the MIFARE ® specific command byte (Refer to [5] for details)[7]

[7] Please keep in mind the different definitions of Littlle and Big Endian in Mifare and PN544 specification

Table 82. Cmd Description

Value	Description
0x60	Authentication A
0x61	Authentication B
0x30	16 bytes reading
0xA0	16 bytes writing
0xA2	4 bytes writing
0xC1	Incrementation
0xC0	Decrementation
0xB0	Transfer
0xC2	Restore
0x38	Read sector[8]
0xA8	Write sector[9]
others	RFU

- **Addr** is the address associated with the MIFARE® command
- **Data** is an array containing either:
  - the data to be sent to the card during a writing operation,
  - or the data to be used during an authentication operation:

  Data0..3 contain the 4 bytes serial number of the card,

  Data4..9 contain the 6 bytes key.

[8] “Read sector” specific command is only relevant for MIFARE 1K/4K tags, usage of this macro command on other MIFARE tags can drive to erratic behavior

[9] “Write sector” specific command is only relevant for MIFARE 1K/4K tags, usage of this macro command on other MIFARE tags can drive to erratic behavior

The response to this command is as follows:

Table 83. NXP_MIFARE_CMD Response parameters

Description	Length
Response code	1
Data	N

- **Response code** possible values: any of the ETSI HCI generic response code and in case of error:

Table 84. Response Code RF Error Description

Value	Response	Description
'10'	WR_RF_ERROR	the target has returned an RF error

- **Data** is an array containing data read from the card in case of a reading command

Table 85. MIFARE Authenticate Command

Value	Command	Description
’22’	*NXP_MIFARE_AUTHENTICATE_CMD*	This command allows authenticate sector of a MIFARE card with key stored in EEPROM (See *153*).

This command has the following parameters:

Table 86. MIFARE Authenticate command

Description	Length
Addr	1
Auth type	1
Key to use	1
UID	4

- Addr is the MIFARE® sector address to authenticate
- Auth type : used to say if command will use key A of key B(see next table)
- Key to use : keyset stored in EEDATA to use for authentication (value 0 to 7 supported, other are RFU)
- UID : Uid of the Mifare card

Table 87. MIFARE Authenticate command – Authentication type value

Value	Description
0x00	Use key A
0x01	Use key B

The response to the *MIFARE* Authenticate command is as follows:

Table 88. NXP_MIFARE_AUTHENTICATE_CMD Command Response

Description	Length
Response code	1

Response code possible values: any of the ETSI HCI generic response code.

Table 89. MIFARE Command

Value	Command	Description
’12’	*NXP_WRA_CONTINUE_ACTIVATION*	The host sends this command to inform the CLF Controller after having received the event EVT_TARGET_DISCOVERED to continue activation in case activation has been stopped after successful SAK response. The response to this command, sent as soon as the activation is finished, indicates the result of the activation procedure.

The response to the *MIFARE* commands is as follows:

Table 90. NXP_WRA_CONTINUE_ACTIVATION Command Response

Description	Length
Response code	1

Response code possible values: any of the ETSI HCI generic response code.

Table 91. MIFARE Registry

Id	Name	Access Rights	Comment	Length	Default
‘10’	*NXP_AUTO_ACTIVATION*	RW (EE)	If set to 0, the activation procedure will stop after Select (SAK has been received). The host could evaluate SAK value and then decide: - to start communicating with the remote card using proprietary commands (see NXP_MIFARE_RAW and NXP_MIFARE_CMD) or - to activate the remote card up to ISO14443-4 level (RATS and PPS) using command NXP_WRA_CONTINUE_ACTIVATION If set to 1, activation follows the flow described in ETSI HCI specification (restrict detection to ISO14443-4 compliant cards).	*1*	*0x01*

0x10

Note: If NXP_AUTO_ACTIVATION is set to 0x00, NXP_WRA_CONTINUE_ACTIVATION is mandatory to fully activate ISO14443-4 cards.

0x32 0x31

It means that NXP_WR_ACTIVATE_ID & NXP_WR_ACTIVATE_NEXT must be followed by NXP_WRA_CONTINUE_ACTIVATION before using NXP_WR_PRESCHECK or WR_XCHG_DATA with ISO14443-4 cards.

0x12 0x30

0x10

In case **NXP_AUTO_ACTIVATION register entry is set to 0**, following flows can be observed:

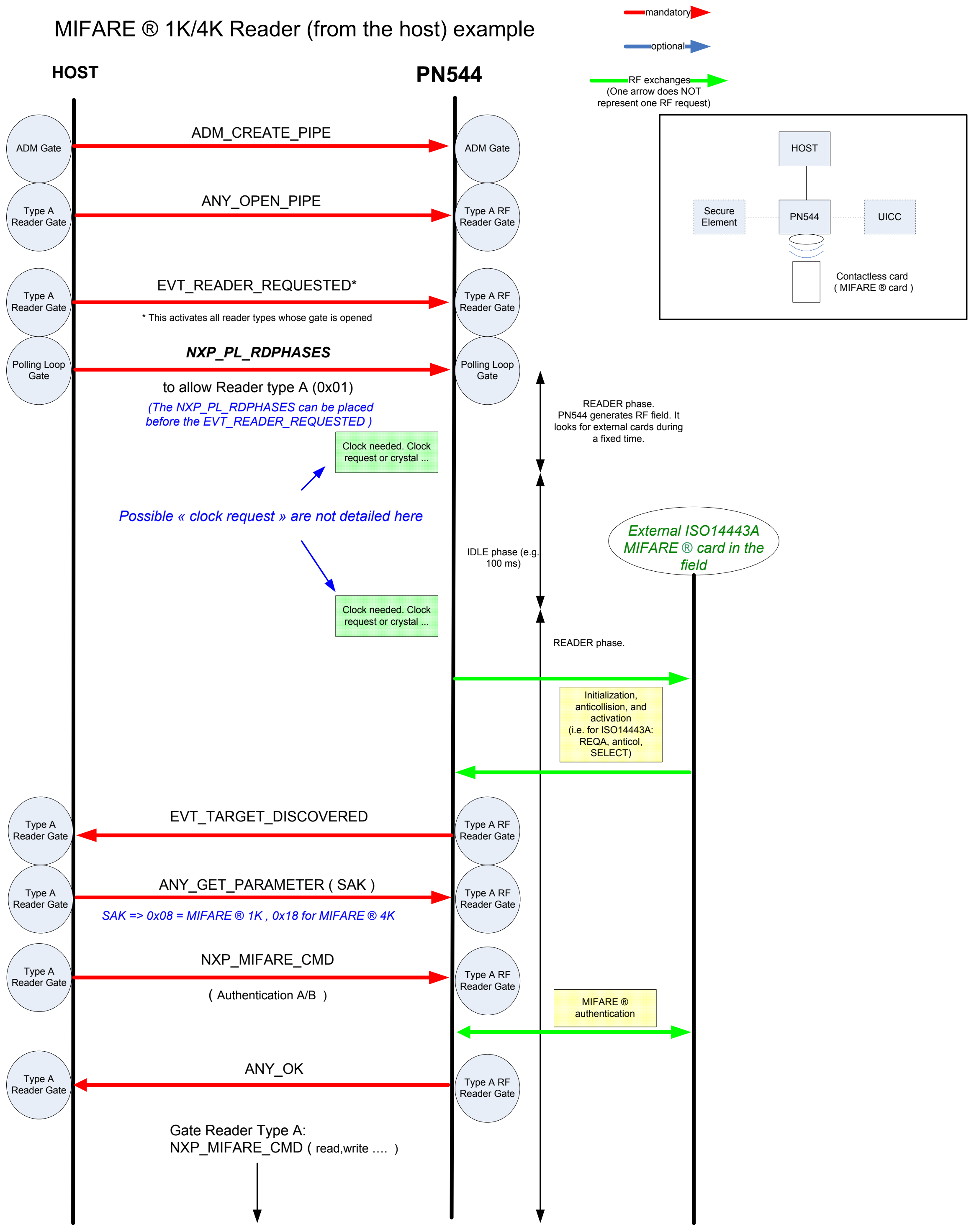

MIFARE ® 1K/4K Reader (from the host) example
mandatory
optional
RF exchanges
(One arrow does NOT represent one RF request)
HOST
PN544
HOST
Secure Element
PN544
UICC
Contactless card (MIFARE ® card)
ADM Gate
ADM_CREATE_PIPE
ADM Gate
Type A Reader Gate
ANY_OPEN_PIPE
Type A RF Reader Gate
Type A Reader Gate
EVT_READER_REQUESTED*
* This activates all reader types whose gate is opened
Type A RF Reader Gate
Polling Loop Gate
NXP_PL_RDPHASES
to allow Reader type A (0x01)
(The NXP_PL_RDPHASES can be placed before the EVT_READER_REQUESTED)
Polling Loop Gate
READER phase. PN544 generates RF field. It looks for external cards during a fixed time.
Clock needed. Clock request or crystal ...
Possible « clock request » are not detailed here
IDLE phase (e.g. 100 ms)
External ISO14443A MIFARE ® card in the field
Clock needed. Clock request or crystal ...
READER phase.
Initialization, anticollision, and activation (i.e. for ISO14443A: REQA, anticol, SELECT)
Type A Reader Gate
EVT_TARGET_DISCOVERED
Type A RF Reader Gate
Type A Reader Gate
ANY_GET_PARAMETER (SAK)
SAK => 0x08 = MIFARE ® 1K , 0x18 for MIFARE ® 4K
Type A RF Reader Gate
Type A Reader Gate
NXP_MIFARE_CMD
(Authentication A/B)
Type A RF Reader Gate
MIFARE ® authentication
Type A Reader Gate
ANY_OK
Type A RF Reader Gate
Gate Reader Type A:
NXP_MIFARE_CMD (read,write ....)

Fig 62. Activation and communication with a MIFARE card

Examples: for a MIFARE 1K (all values are in hex)

- Authentication example :
- 60 02 E2 3F B8 1E FF FF FF FF FF FF

=> authenticate using the keys FF FF FF FF FF FF (value by default for a new card) to the address 02 with UID number E2 3F B8 1E

- Read 16 bytes from the address **02** :
- 30 **02**
-
- Write 16 bytes to the address **02** :

A0 **02** 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10

9.9 FeliCa Reader

PN544 offers FeliCa reader functionality through the generic “Reader RF gates” features of the ETSI HCI specification. This is done with the following proprietary gate:

Table 92. Type F Reader RF gate

Gate	GID
Type F reader RF gate	‘14’

Activation/Deactivation sequence are done using the generic events “EVT_READER_REQUESTED”, “EVT_END_OPERATION”, and “EVT_TARGET_ACTIVATED” described in the “Reader RF gates” and “Reader Gates” of ETSI HCI specifications. Activation sequence is done without any filtering (Polling request is performed with SYSTEM_CODE field set to 0xFFFF). Then the activated FeliCa card can be accessed through the following NXP proprietary commands of the “Type F reader RF gate”:

Table 93. NXP_FELICA_RAW Command

Value	Command	Description
’20’	*NXP_FELICA_RAW*	This command allows the exchange of raw data between the host and a FeliCa card previously activated

The *NXP_FELICA_RAW* command has the following parameters:

Table 94. NXP_FELICA_RAW Parameters

Description	Length
TO	1
Status	1
Data	N

- TO value specify the timeout (in ms) to be used (value 0 means infinite timeout)
- Status: bits b0 to b2 indicate information on valid bits in the last data byte (value 0 indicates all are valid, from 1 to 7 it indicated the number of valid bits).
- Data is the raw frame to be sent to the card.

The response to the *NXP_FELICA_RAW* command is as follows:

Table 95. FeliCa Raw Response parameters

Description	Length
Status	1
Data	N

- Status: bits b0 to b2 indicate information on valid bits in the last data byte (value 0 indicates all are valid, from 1 to 7 it indicated the number of valid bits.
- Data is the raw frame received from the card.

Note: As FeliCa operates with full bytes, Status byte of NXP_FELICA_RAW command value should be 0x00.

Table 96. NXP_FELICA_CMD Command

Value	Command	Description
'21'	*NXP_FELICA_CMD*	This command allows the exchange of data between the host and a FeliCa card previously activated.

This command has the following parameters:

Table 97. NXP_FELICA_CMD Parameters

Description	Length
Cmd	1
Data	N

- Cmd describe the FeliCa command type

Table 98. FeliCa Command Description

Value	Description
0x00	Check
0x01	Update
others	RFU

- Data is an array containing the data to be sent within the command as follows:
  - Check command:

NbrOfServices	Service Code List	NbrOfBlocks	Block List
1 byte	*(2 * NbrOfServices) bytes*	*1 byte*	*(2 * NbrOfBlocks) bytes*

– Update command:

NbrOfServices	Service Code List	NbrofBlocks	Block List	Block Data
1 byte	*(2*NbrOfServices) bytes*	*1 byte*	*(2*NbrOfBlocks) bytes*	*(16*NbrOfBlocks) bytes*

The response to this command is as follows:

Table 99. FeliCa Command Response

Description	Length
Data	N

- Data is an array containing the data to be sent within the response as follows:
  – Check response:

Status Flag 1	Status Flag 2	NbrOfBlocks	Block Data
1 byte	*1 byte*	*1 byte*	*(16 * NbrOfBlocks) bytes*

– Update response:

Status Flag 1	Status Flag 2
1 byte	*1 byte*

Table 100. FeliCa Registry

Id	Name	Access Rights	Comment	Length	Default
‘01’	*NXP_FELICA_SYSTEMCODE*	RO (RAM)	System Code of the FeliCa card.	*2*	*‘0000’*
‘04’	*NXP_ FELICA_CURRENTIDM*	RO (RAM)	IDm of the tag currently communicating with the reader.	*8*	*‘0000000000000000’*
‘05’	*NXP_FELICA_CURRENTPMM*	RO (RAM)	PMm of the tag currently communicating with the reader.	*8*	*NA*

9.10 Jewel/Topaz Reader

PN544 offers Jewel/Topaz reader functionality through the generic “Reader RF gates” features of the ETSI HCI specification. This is done with the following proprietary gate:

Table 101. Jewel reader RF gate

Gate	GID
Jewel reader RF gate	‘15’

Activation/Deactivation sequence are done using the generic events “EVT_READER_REQUESTED”, “EVT_END_OPERATION”, “EVT_TARGET_ACTIVATED” described in the “Reader RF gates” and “Reader Gates” of ETSI HCI specifications. Then the activated Jewel/Topaz card can be accessed through the following NXP proprietary command of the “Jewel reader RF gate”:

Table 102. NXP_JEWEL_RAW_CMD Command

Value	Command	Description
’23’	*NXP_JEWEL_RAW_CMD*	This command is used by the host to use Jewel native command in a RAW Format

The *NXP_JEWEL_RAW_CMD* command has the following parameters:

Table 103. NXP_JEWEL_RAW_CMD Parameters

Description	Length
Cmd	1
Data	N

- Cmd describe the command type
- Data are parameters linked to the command

Table 104. Jewel Command Description For Classic Jewel Card (bit 4 of HRO = 0)

Value	Description	Data
0x00	ReadALL	Byte 1..2 : 0x00 Byte 3..6 : card ID

Value	Description	Data
0x01	Read1	Byte 1 : address Byte 2 : 0x00 Byte 3..6 : card ID
0x02	Read4	Byte 1 : address
0x03	Read8	Byte 1 : address Byte 2..9 : 0x00 Byte 9..12 : card ID
0x53	Write1E	Byte 1 : address Byte 2 : data to write Byte 3..6 : card ID
0x54	Write4E	Byte 1 : address Byte 2..5: data to write Byte 6..9 : card ID
0x55	Write8E	Byte 1 : address Byte 2..9: data to write Byte 9..12 : card ID
0x1A	Write1NE	Byte 1 : address Byte 2 : data to write Byte 3..6 : card ID
0x1B	Write4NE	Byte 1 : address Byte 2..5: data to write Byte 6..9 : card ID
0x1C	Write8NE	Byte 1 : address Byte 2..9: data to write Byte 9..12 : card ID
0x78	RID	Byte 1..6 : 0x00
0x10	Read Seg	Byte 1..9: 0x00 Byte 9..12 : card ID
others	RFU	

Table 105. Jewel Command Description For NDEF Card (bit 4 of HRO = 1)

Value	Description	Data
0x00	ReadALL	Byte 1..2 : 0x00 Byte 3..6 : card ID
0x01	Read1	Byte 1 : address Byte 2 : 0x00 Byte 3..6 : card ID
0x02	Read8	Byte 1 : address Byte 2..9 : 0x00 Byte 9..12 : card ID
0x53	Write1E	Byte 1 : address Byte 2 : data to write Byte 3..6 : card ID
0x54	Write8E	Byte 1 : address Byte 2..9: data to write Byte 9..12 : card ID
0x1A	Write1NE	Byte 1 : address Byte 2 : data to write Byte 3..6 : card ID
0x1B	Write8NE	Byte 1 : address Byte 2..9: data to write Byte 9..12 : card ID
0x78	RID	Byte 1..6 : 0x00
0x10	Read Seg	Byte 1..9: 0x00 Byte 9..12 : card ID
others	RFU	

The response to the *NXP_JEWEL_RAW_CMD* command is as follow:

Table 106. Jewel Raw Command Response

Description	Length
Data	N

- Data contain the data read, in case of write command N=0.

Discovered tag information (UID, Size, HR0 …) can be retrieved using the *NXP_JEWEL_RAW_CMD* with Jewel "RID" command as parameter.

9.11 ISO15693

PN544 offers ISO15693 reader functionality through the generic "Reader RF gates" features of the ETSI HCI specification. This is done with the following proprietary gate:

Table 107. ISO15693 reader RF gate

Gate	GID
ISO15693 reader RF gate	'12'

Activation/Deactivation sequence are done using the generic events "EVT_READER_REQUESTED", "EVT_END_OPERATION", "EVT_TARGET_ACTIVATED" described in the "Reader RF gates" and "Reader Gates" of ETSI HCI specifications. Then the activated ISO15693 card can be accessed through the following NXP proprietary command of the "ISO15693 reader RF gate":

Table 108. Reader ISO15693 Command

Value	Command	Description
'20'	*NXP_ISO15693_CMD*	This command is used by the host to access remote tag.

The *NXP_ISO15693_CMD* command has the following parameters:

Table 109. Reader ISO15693 Command Parameters

Description	Length
Payload	N

- Payload defines general request content as defined in ISO15693-3 specification:

• Flags	• Command code	• Parameters	• Data

The response to the *NXP_ISO15693_CMD* command is as follows:

Table 110. Reader ISO15693 Command Response

Description	Length
Payload	N

- Payload defines general response content as defined in ISO15693-3 specification:

• Flags	• Parameters	• Data

Table 111. ISO15693 Registry

Id	Name	Access Rights	Comment	Length	Default
‘01’	*NXP_ISO15693_INVENTORY*	RO (RAM)	Indicates the VICC parameters as follows: - 1st byte: FLAGS - 2nd byte: DSFID - 3rd to 10th bytes: UID	*10*	*‘00’* *‘00* *‘0000000000000000’*
‘02’	*NXP_ISO15693_AFI*	RW (EE)	Indicates the AFI used by PN544 for detection (during Inventory procedure)	*1*	*‘00’*

9.12 NFCIP-1

PN544 offers Peer to Peer functionality based on NFCIP-1 protocol through the two following proprietary gates:

Table 112. NFC-IP1 gate

Gate	GID
NFCIP1 Initiator RF gate	‘30’
NFCIP1 Target RF gate	‘31’

Table 113. NFC-IP1 Events

Value	Event	Description
‘01’	*NXP_EVT_NFC_SND_DATA*	This event conveys payload from the host to the CLF. In Initiator mode this leads to a DEP_REQ (information PFB). In Target mode this leads to a DEP_RES (information PFB).
‘02’	*NXP_EVT_NFC_ACTIVATED*	This event is sent to the host to notify that NFCIP-1 protocol has been activated. Meaning that, in initiator mode it has activated a remote NFC target, and in target mode it has been activated by a remote NFC initiator
‘03’	*NXP_EVT_NFC_DEACTIVATED*	This event is sent by the NFC target to its host to notify that the connection has been closed. The NFCIP-1link deactivation might have several reasons: - The remote NFC Initiator has explicitly performed a link deactivation - The NFCIP-1 link has detected an error and it was not able to recover the communication on target side
‘04’	*NXP_EVT_NFC_RCV_DATA*	This event conveys payload from the CLF to the host. In Initiator mode this event has been generated due to a DEP_RES. In Target mode this event has been generated because of a DEP_REQ.

Value	Event	Description
‘05’	*NXP_EVT_NFC_CONTINUE_MI*	This event is sent to the host to indicate the remote peer acknowledged the previous Meta-chained packet and then the host can send the next one.

The *NXP_EVT_NFC_ACTIVATED* event has the following parameter:

Table 114. NFCIP-1 Activated Event

Description	Length
Mode	1

- Mode indicates in which mode the current activation as be done
  - 0x00: Passive mode
  - 0x01: Active

The *NXP_EVT_NFC_SND_DATA* event has the following parameter:

Table 115. NXP_EVT_NFC_SND_DATA Parameters

Description	Length
MI	1
Data	N

- MI indicates Multiple Information chaining information (activated if set to 1)
- Data which are received from the peer

The *NXP_EVT_NFC_RCV_DATA* event has the following parameter:

Table 116. NXP_EVT_NFC_RCV_DATA Parameters

Description	Length
RF error indicator	1
MI	1
Data	N

- RF error indicator:

Table 117. RF error indicator Description

Value	Description
0x00	no RF error

Value	Description
0x01	RF error
others	RFU

- MI indicates Multiple Information chaining information (activated if set to 1).
- Data which are received from the peer (in case of RF error, data length is null)

NFCIP-1 RF gates use the same mechanism as ETSI HCI “Reader RF gate” (for initiator gate) and “Card RF gate” (for Target gate) for discovery of remote peers (see following sub-sections).

9.12.1 Initiator

To start discovering a remote NFC target, the host shall send an ETSI HCI generic event “EVT_READER_REQUESTED”. When a remote NFC target has been activated, host receives the proprietary event “NXP_EVT_NFC_ACTIVATED”. Then the host can exchange data with the remote peer using event “NXP_EVT_NFC_SND_DATA” and receiving event “NXP_EVT_NFC_RCV_DATA”. The host can close the communication with remote peer by sending ETSI HCI generic event “EVT_END_OPERATION”.

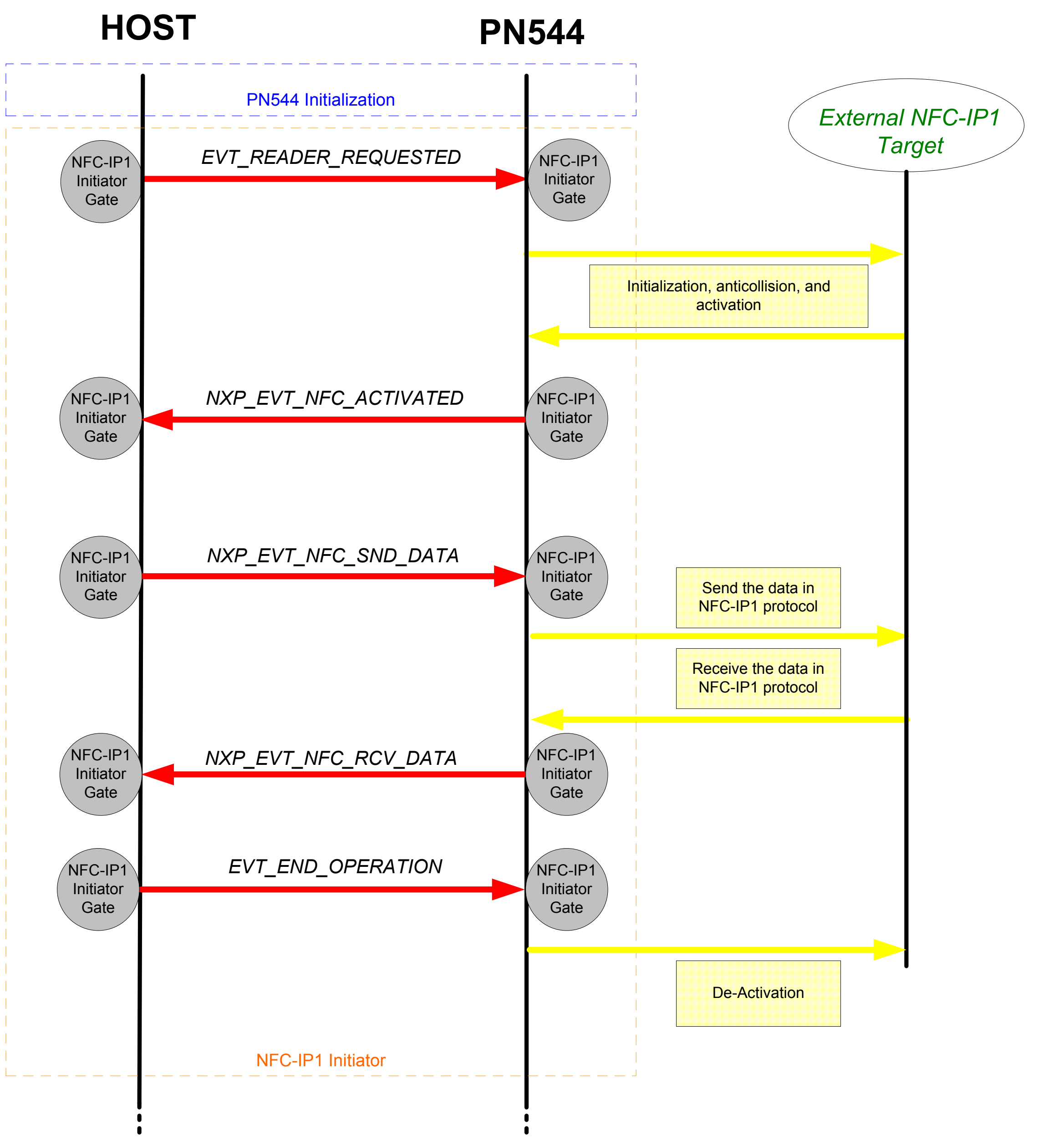


Fig 63. NFC-IP1 Initiator

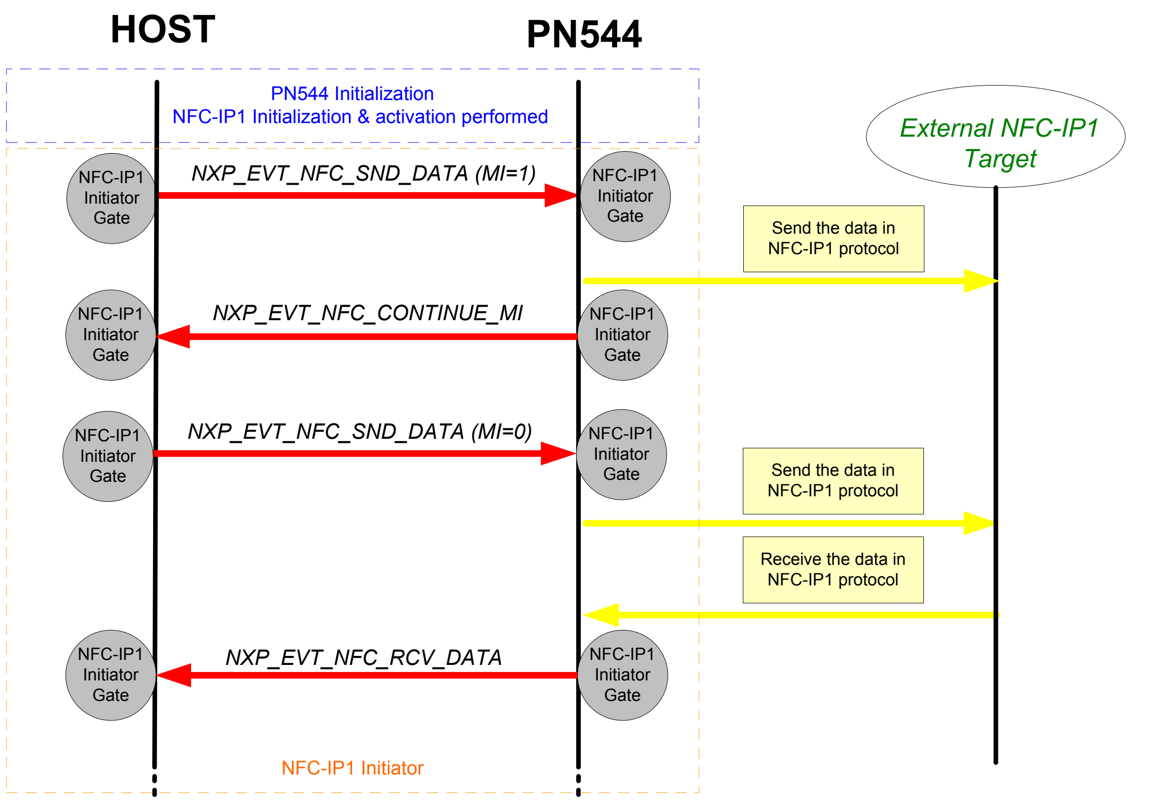


Fig 64. NFC-IP1 Initiator exchange using *NXP_EVT_NFC_CONTINUE_MI*

Note: When using the MI chaining information of the NXP_EVT_NFC_SND_DATA event, the host must insure that the size of data transmitted within this message is lower than the ‘maximum frame length’ supported (can be retrieved in NXP_NFCI_PARAM).

The proprietary ‘NFCIP1 Initiator’ gate has the following command and registry:

Table 118. NFC-IP1 Initiator Command

Value	Command	Description
’12’	*NXP_NFCI_ATTREQUEST*	The host sends this command to check the target is still present in the field (see [4] §12.6.3 - Attention – Target present). The response to this command indicates the result of the Attention procedure.

Value	Command	Description
'13'	*NXP_NFCI_CONTINUE_ACTIVATION*	The host sends this command to allow NFCIP-1 activation when a NFC target has been discovered in Type A Reader RF gate or type F Reader RF gate. Type A: In case the proprietary NXP_AUTO_ACTIVATION registry entry of the Type A Reader RF gate has been set to 0 (stop activation after SAK is received), the EVT_TARGET_DISCOVERED will be sent to the Type A Reader application gate even if the remote target is an NFC device. Type F: In case a pipe has been opened on the Type F Reader RF gate, the activation procedure is stopped after the POLL_REQ and the EVT_TARGET_DISCOVERED will be sent to the Type F Reader application gate even if the remote target is an NFC device. Then, in order to be able to communicate with the remote NFC target, the host has to send the command *NXP_NFCI_CONTINUE_ACTIVATION* to the NFCIP1 Initiator RF gate to indicate to the CLF to continue NFCIP1 activation (ATR/PSL exchange). The response to this command, sent as soon as the activation is finished, indicates the result of the activation procedure.

Here are examples of *NXP_NFCI_CONTINUE_ACTIVATION* use:

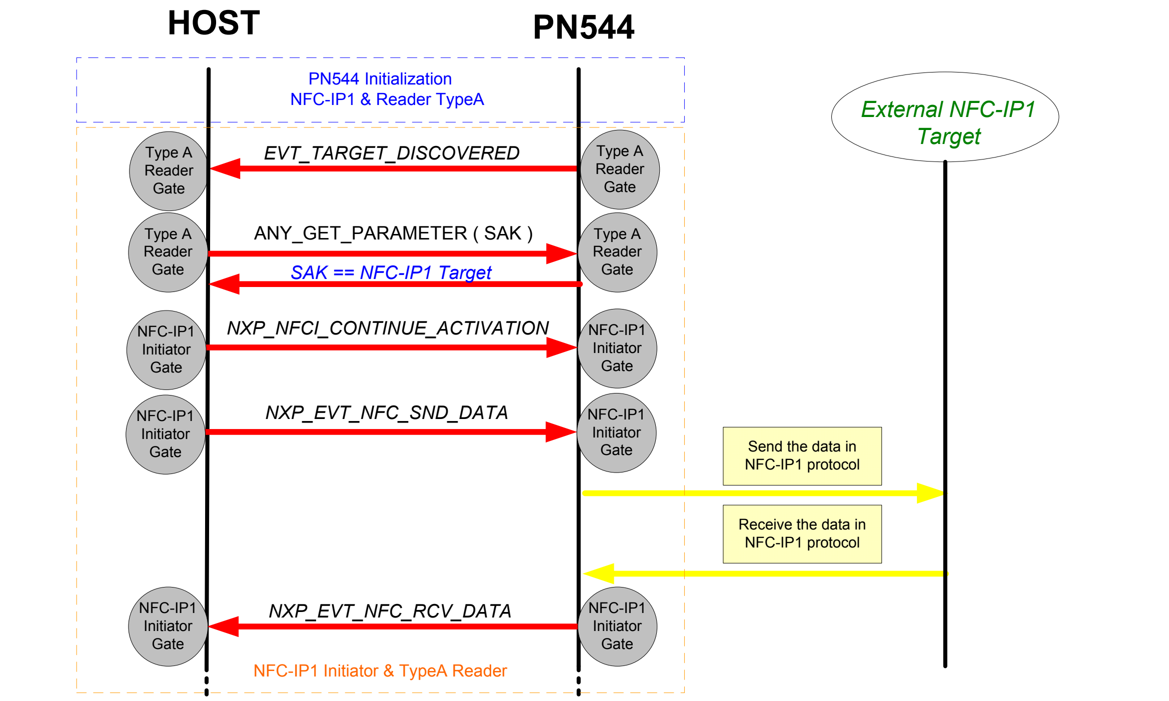


Fig 65. NFC-IP1 Initiator & Type A reader (*NXP_NFCI_CONTINUE_ACTIVATION*)

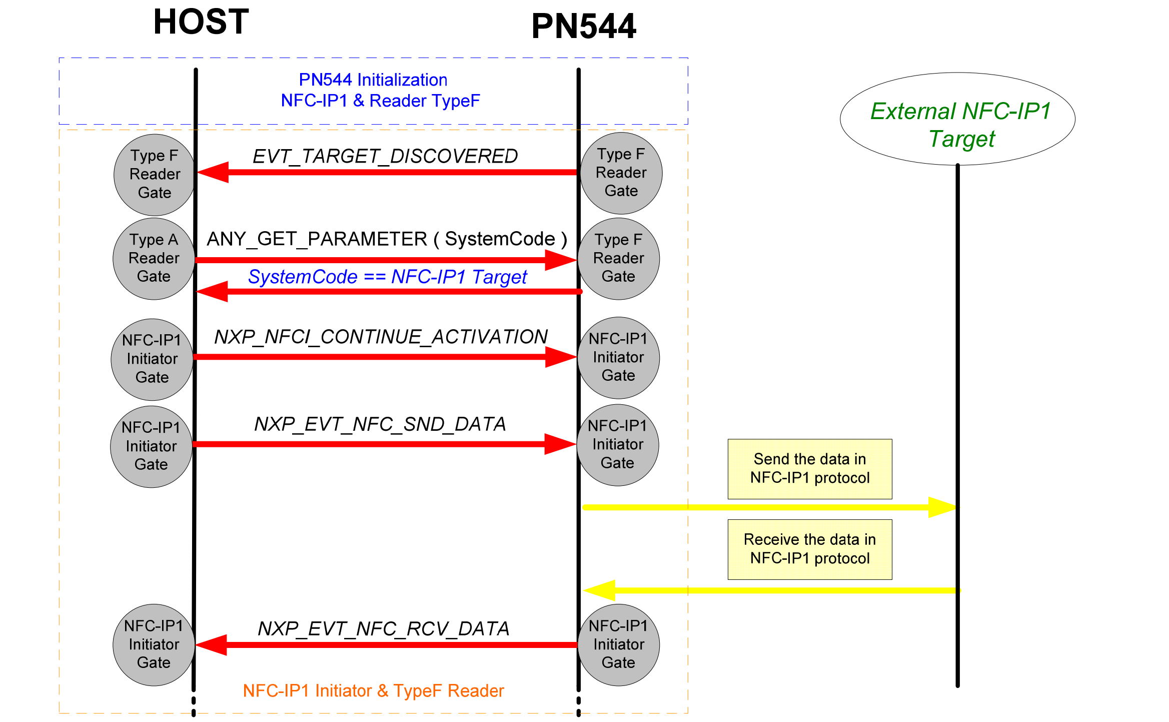


Fig 66. NFC-IP1 Initiator & TypeF reader (*NXP_NFCI_CONTINUE_ACTIVATION*)

Table 119. NFC-IP1 Initiator Registry

<table>
<tr><th>Id</th><th>Name</th><th>Access Rights</th><th>Comment</th><th>Length</th><th>Default</th></tr>
<tr><td>‘01’</td><td><i>NXP_NFCI_MODE</i></td><td>RW
(EE)</td><td>Supported NFCIP-1 Modes.
This value shall be interpreted as a bit mask:<table>
<tr><td>Bit</td><td>Mode</td></tr>
<tr><td>0</td><td>106kbit/s passive</td></tr>
<tr><td>1</td><td>212kbit/s passive</td></tr>
<tr><td>2</td><td>424kbit/s passive</td></tr>
<tr><td>3</td><td>106kbit/s active(*)</td></tr>
<tr><td>4</td><td>212kbit/s active(*)</td></tr>
<tr><td>5</td><td>424kbit/s active(*)</td></tr>
<tr><td>6 ... 7</td><td>RFU</td></tr>
</table>0 -> not supported
1 -> supported
(*) In active only one mode is used in the polling loop sequence (only the one which have the lowest baudrate from the enabled ones)</td><td><i>1</i></td><td><i>0x00</i></td></tr>
<tr><td>‘02’</td><td><i>NXP_NFCI_ATR_REQ</i></td><td>RW
(EE)</td><td>Contains the General Bytes of the ATR_REQ (max size is 48 bytes)</td><td><i>N</i></td><td><i>N=0</i></td></tr>
<tr><td>‘03’</td><td><i>NXP_NFCI_ATR_RES</i></td><td>RO
(RAM)</td><td>Contains the General Bytes of the ATR_RES (max size is 48 bytes)</td><td><i>N</i></td><td><i>N=0</i></td></tr>
</table>

Id	Name	Access Rights	Comment	Length	Default
‘04’	*NXP_NFCI_BRS*	RW (EE)	Specifies the bit rates to be used for DEP exchanges: - Bits 0 to 2: Target to initiator - Bits 3 to 5: Initiator to target Bit rate is coded as follow: 0x00 -> 106 kbps 0x01-> 212 kbps 0x02-> 424 kpbs Others values are RFU - Bit 6: RFU - Bit 7: Indicates if bit rates have to be changed for DEP exchanges (PSL command to be used or not) 0x0 -> bit rates remain unchanged (from bit rate used for activation) 0x1 -> bit rates to be changed according to other bits configuration	*1*	*0x00*
‘06’	*NXP_NFCI_DID*	RW (EE)	Value used as DID (if set to 0, DID is not used).	*1*	*0x00*
‘09’	*NXP_NFCI_STATUS*	RO (RAM)	Contains the current status of the NFCIP-1 link when communication has been set. 0x00 -> data is expected from the host 0x01 -> data is expected from the RF side Others values are RFU	*1*	*0x00*
‘0A’	*NXP_NFCI_NFCID3I*	RO (RAM)	Contains the random NFCID3I conveyed with the ATR_REQ.	*10*	*‘00000000000000000000’*
‘0B’	*NXP_NFCI_NFCID3T*	RO (RAM)	Contains the random NFCID3T conveyed with the ATR_RES.	*10*	*‘00000000000000000000’*

Id	Name	Access Rights	Comment	Length	Default
‘0C’	*NXP_NFCI_PARAM*	RO (RAM)	Contains the current parameters of the NFCIP-1 link when communication has been set. - bits 0 to 2: datarate Target to Initiator - bits 3 to 5: datarate Initiator to Target 0 -> Divisor equal to 1 1 -> Divisor equal to 2 2 -> Divisor equal to 4 Other values -> RFU - bits 6 to 7: maximum frame length 0 -> 64 bytes 1 -> 128 bytes 2 -> 192 bytes 3 -> 256 bytes	*1*	*0x00*

Make sure to manage the maximum frame length received from the target correctly on host side.

PN544 will not take care of this parameter which could lead to a buffer overflow on target side!

9.12.2 Target

When a remote NFC initiator has activated the local NFC target handled in the PN544, the host receives the proprietary event “NXP_EVT_NFC_ACTIVATED”. Then the host can exchange data with the remote peer receiving event “NXP_EVT_NFC_RCV_DATA” and using event “NXP_EVT_NFC_SND_DATA”. When the remote NFC initiator closes the communication, the host receives event “NXP_EVT_NFC_DEACTIVATED”.

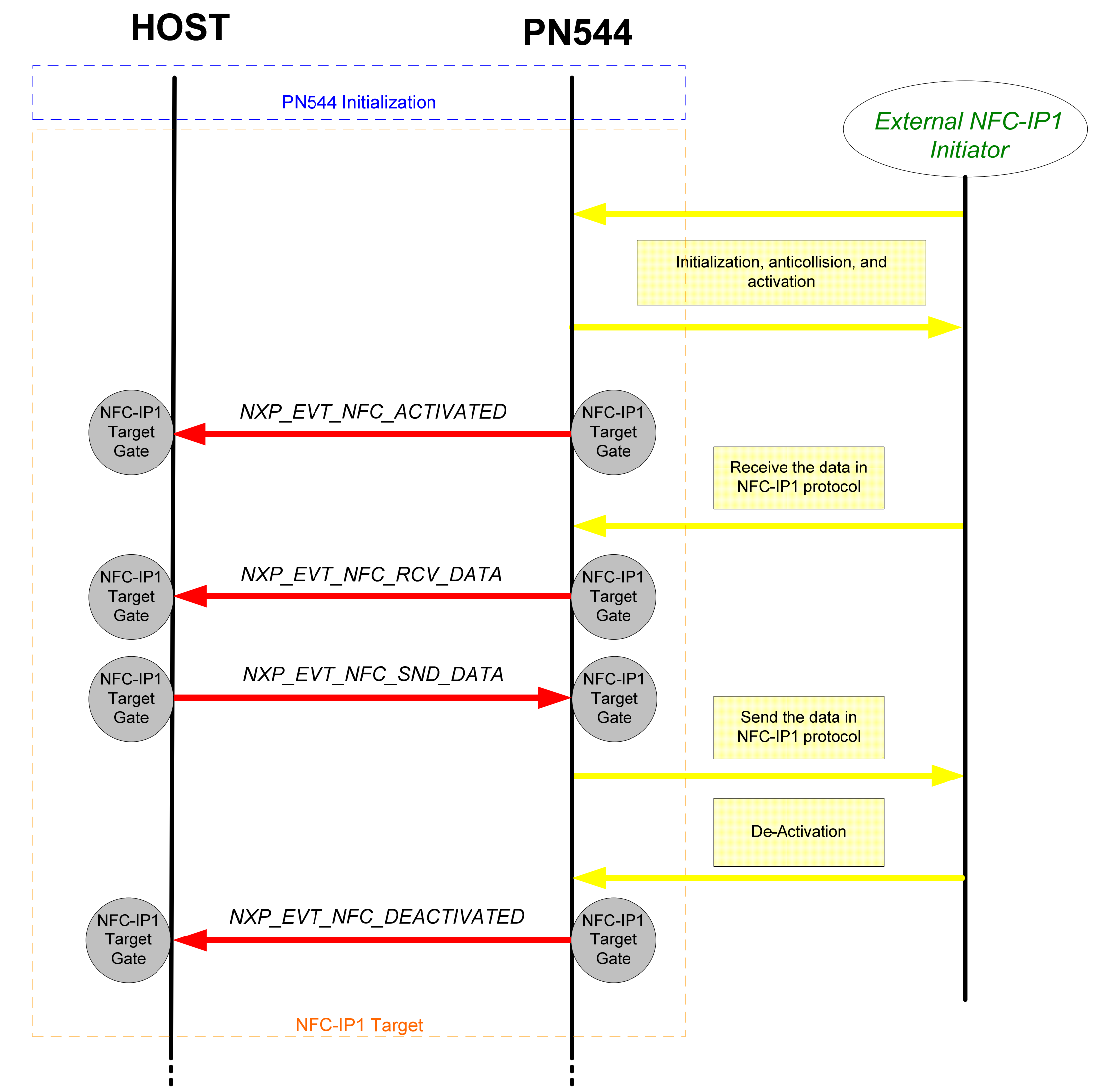


Fig 67. NFC-IP1 Target

Note: In case of ***NFC target active mode***, the clock may be requested (according to *PlClockRequest* configuration) as PN544 shall generate its own field.

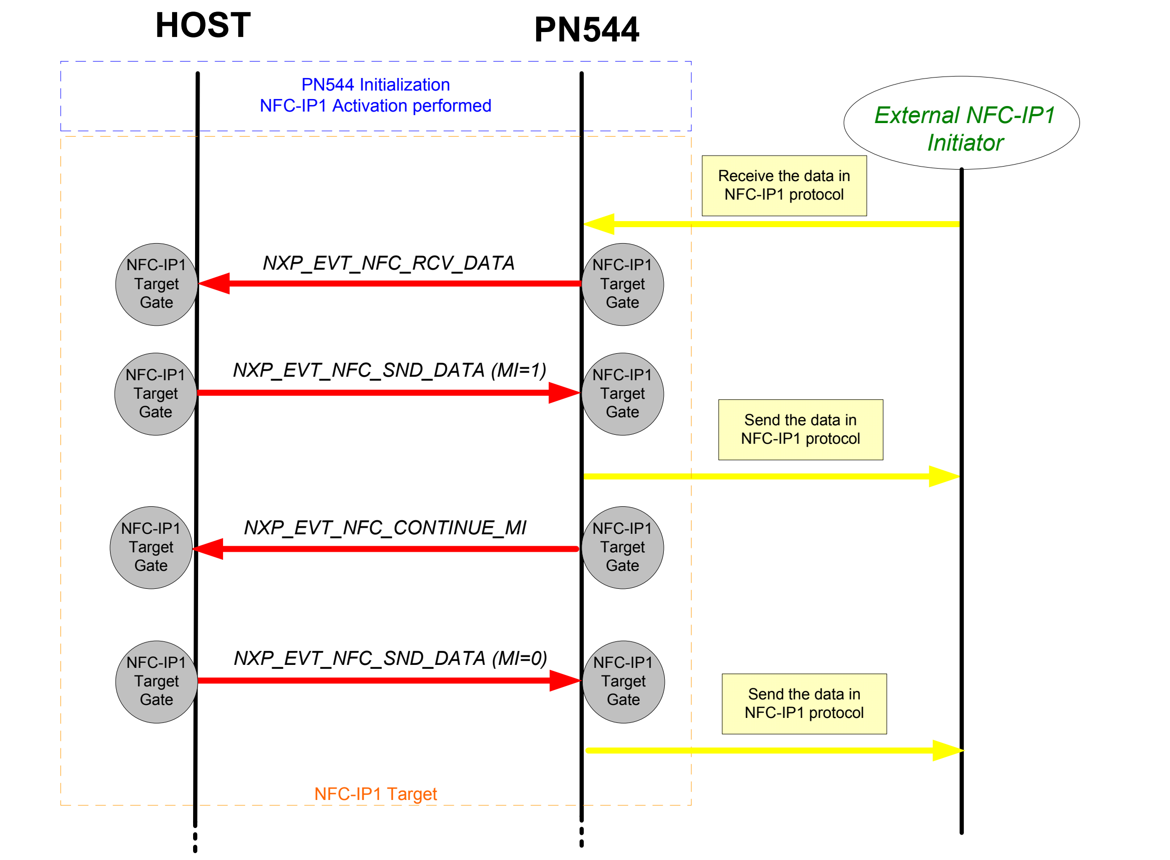


Fig 68. NFC-IP1 TARGET with *NXP_EVT_NFC_CONTINUE_MI*

Note: When using the MI chaining information of the NXP_EVT_NFC_SND_DATA event, the host must insure that the size of data transmitted within this message is lower than the ‘maximum frame length’ supported (can be retrieved in NXP_NFCT_PARAM).

The proprietary ‘NFCIP1 Target’ gate has the following registry:

Table 120. NFC-IP1 Target Registry

<table>
<tr><th>Id</th><th>Name</th><th>Access Rights</th><th>Comment</th><th>Length</th><th>Default</th></tr>
<tr><td>‘01’</td><td><i>NXP_NFCT_MODE</i></td><td>RW
(EE)</td><td>Supported NFCIP-1 Modes.
This value shall be interpreted as a bit mask:<table><tr><td>Bit</td><td>Mode</td></tr><tr><td>0</td><td>106kbit/s passive</td></tr><tr><td>1</td><td>212kbit/s passive</td></tr><tr><td>2</td><td>424kbit/s passive</td></tr><tr><td>3</td><td>Active</td></tr><tr><td>4 ... 7</td><td>RFU</td></tr></table>0 -> not supported
1 -> supported</td><td><i>1</i></td><td><i>0x00</i></td></tr>
<tr><td>‘02’</td><td><i>NXP_NFCT_ATR_REQ</i></td><td>RO
(RAM)</td><td>Contains the General Bytes of the ATR_REQ (max size is 48 bytes)</td><td><i>N</i></td><td><i>N=0</i></td></tr>
<tr><td>‘03’</td><td><i>NXP_NFCT_ATR_RES</i></td><td>RW
(EE)</td><td>Contains the General Bytes of the ATR_RES (max size is 48 bytes)</td><td><i>N</i></td><td><i>N=0</i></td></tr>
<tr><td>‘09’</td><td><i>NXP_NFCT_STATUS</i></td><td>RO
(RAM)</td><td>Contains the current status of the NFCIP-1 link when communication has been set.
0x00 -> data is expected from the host
0x01 -> data is expected from the RF side
Others values are RFU</td><td><i>1</i></td><td><i>0x00</i></td></tr>
<tr><td>‘0A’</td><td><i>NXP_NFCT_NFCID3I</i></td><td>RO
(RAM)</td><td>Contains the random NFCID3I conveyed with the ATR_REQ.</td><td><i>10</i></td><td><i>‘00000000000000000000’</i></td></tr>
<tr><td>‘0B’</td><td><i>NXP_NFCT_NFCID3T</i></td><td>RO
(RAM)</td><td>Contains the random NFCID3T conveyed with the ATR_RES.</td><td><i>10</i></td><td><i>‘00000000000000000000’</i></td></tr>
</table>

Id	Name	Access Rights	Comment	Length	Default
‘0C’	*NXP_NFCT_PARAM*	RO (RAM)	Contains the current parameters of the NFCIP-1 link when communication has been set. - bits 0 to 2: datarate Target to Initiator - bits 3 to 5: datarate Initiator to Target 0 -> Divisor equal to 1 1 -> Divisor equal to 2 2 -> Divisor equal to 4 Other values -> RFU - bits 6 to 7: maximum frame length 0 -> 64 bytes 1 -> 128 bytes 2 -> 192 bytes 3 -> 256 bytes	*1*	*0x00*
‘0D’	*NXP_NFCT_MERGE*	RW (EE)	Indicates if the NFCIP-1 target feature must be merged with Type A RF card feature in order to present only one type A target (set of the related bit in SAK to reflect the ISO18092 compliancy). 0x00 -> disabled 0x01 -> enabled Others values are RFU	*1*	*0x00*

9.13 Reader RF gates – additional commands

PN544 implement reader feature defined in ETSI HCI specification. However, it provides the following additional commands:

Table 121. Reader RF Additional Commands

Value	Command	Description
'30'	*NXP_WR_PRESCHECK*	The host sends this command to check the remote card is still present in the field. The response to this command indicates the result of the Presence Check procedure. This command is only valid for ISO14443-4 compliant card, otherwise error ANY_E_CMD_NOT_SUPPORTED is returned as response.
'31'	*NXP_WR_ACTIVATE_NEXT*	The host sends this command to activate next remote card (in case of several cards in the field). The response to this command indicates the result of the activation procedure of next card (when the response is sent, next remote card has been activated). Parameter in the response indicates if there is still other cards in the field (previous activated cards will not be visible). By successive use of this command, host is able to successively activate all cards within a technology. If this command is sent when no more other card is present in the field (no collision detected) this command will return an error.
'32'	*NXP_WR_ACTIVATE_ID*	The host sends this command to select a previously detected remote card (by means of *NXP_WR_ACTIVATE_NEXT* command) using the ID of the card.
'33'	*NXP_WR_DISPATCH_TO_UICC*	The host sends this command to dispatch the remote card to the UICC (reactivation of the card is implicitly done before the dispatch). The response to this command indicates the result of the dispatch procedure. After this command has been processed, no card is activated, if the host wants to use a card (e.g. for presence check purpose) it need to reactivate it via *NXP_WR_ACTIVATE_ID* command.

The response to the *NXP_WR_ACTIVATE_NEXT* command is as follows:

Table 122. NXP_WR_ACTIVATE_NEXT Response

Description	Length
Status	1

- Status field indicates if the activation has been successful

Table 123. NXP_WR_ACTIVATE_NEXT Response Description

Value	Description
0x00	Single target in the field
0x03	Several targets in the field

The *NXP_WR_ACTIVATE_ID* command has the following parameters:

Table 124. NXP_WR_ACTIVATE_ID Command

Description	Length
ID	N

- ID value specify ID of the remote card to activate:

  - Type A Reader RF gate: UID

  - Type B Reader RF gate: PUPI

  - Type F Reader RF gate: IDm

  (Jewel RF reader gate does not allow multiple activation)

The *NXP_WR_DISPATCH_TO_UICC* command has the following parameters:

Table 125. NXP_WR_DISPATCH_TO_UICC Command

Description	Length
ID	N

- ID value specify ID of the remote card to activate:

  - Type A Reader RF gate: UID

  - Type B Reader RF gate: PUPI

The response to the *NXP_WR_DISPATCH_TO_UICC* command is as follows:

Table 126. NXP_WR_DISPATCH_TO_UICC Command Response

Description	Length
ErrorCode	1

- ErrorCode field indicates reason of the dispatch failure test (optional: ErrorCode field is only present in case of failure)
- - 0x01: indicates error during card reactivation
- - 0x02: indicates the UICC is not interested in such card

Table 127. Reader RF Additional Event

Value	Event	Description
'35'	NXP_EVT_RELEASE_TARGET	This event allows the host to release the current activated target (as it is done using EVT_END_OPERATION) without stopping the detection request (previously started by use of EVT_READER_REQUESTED). This event trigs the restart of the polling loop from the next phase.

The *NXP_EVT_RELEASE_TARGET* event has parameters as follows

Table 128. Release target Parameter

Description	Length
Option	1

- Option parameter indicates whether the polling loop shall continue from current technology or repoll in the current technology. This parameter is optional (if not present the "Continue" behavior is applied).
  - 0x00: Continue
  - 0x01: Repoll

9.14 Type A PICC

This paragraph explains in which ways the PN544 can behave like a Type A card.

9.14.1 Type A PICC over NFC WI (SMX as a Type A card)

The Type A card emulation can be located in the secure element connected to PN544 over NFC WI link.

Refer to paragraph NFC WI.

9.14.2 Type A PICC over SWP (UICC as a Type A card)

The Type A card emulation can be located in the UICC connected to PN544 over SWP link.

Refer to paragraph SWP.

9.14.3 Type A PICC over host link (I2C, SPI, HSU) (host as a Type A card)

The Type A card emulation can be located in the host connected to PN544 over I2C, SPI or HSU link.

The host is not allowed to transmit a UID. The UID in that case is a single size UID, whose first byte is 0x08 and the other three bytes are randomly generated by the PN544.

9.14.4 Multiple Type A PICC

PN544 cannot emulate two type A cards at the same time.

It is not possible to emulate at the same time a Type A card in the UICC and in the host. (The Create Pipe, on an already in use card RF gate, will return an error). The host decides to allow or not the UICC to emulate a Type A card. If it allows the UICC emulate a Type A card, then it cannot itself emulate a Type A card.

It is not possible to emulate at the same time a Type A card in the UICC or host and in the SmartMX.

However the host can decide to dynamically enable/disable each Secure Element.

9.15 Type A PCD

This paragraph explains how to realize a PN544 and its host(s) can be a Type A reader.

9.15.1 Type A PCD over SWP (UICC as a Type A reader)

The UICC can run a Type A reader application, as described in HCI specification.

Refer to paragraph SWP.

9.15.2 Type A PCD over host link (I2C, SPI, HSU) (Host as a Type A reader)

The host processor can run a Type A reader application, as described in HCI specification.

Refer to paragraph Polling Loop (host reader example).

9.15.3 Meaning of additional commands of Reader RF gates for Type A

Refer to Reader RF gates additional commands chapter for commands description.

9.15.3.1 Several Type A cards in the RF field: use NXP_WR_ACTIVATE_NEXT, NXP_WR_ACTIVATE_ID

When there is several Type A cards in the field, the PN544 can send *NXP_WR_ACTIVATE_NEXT* to put in HALT the first card detected and activate the other card in the field (Fig 69).

To reactivate a chosen card, its UID is used, in NXP_WR_ACTIVATE_ID command (Fig 70)

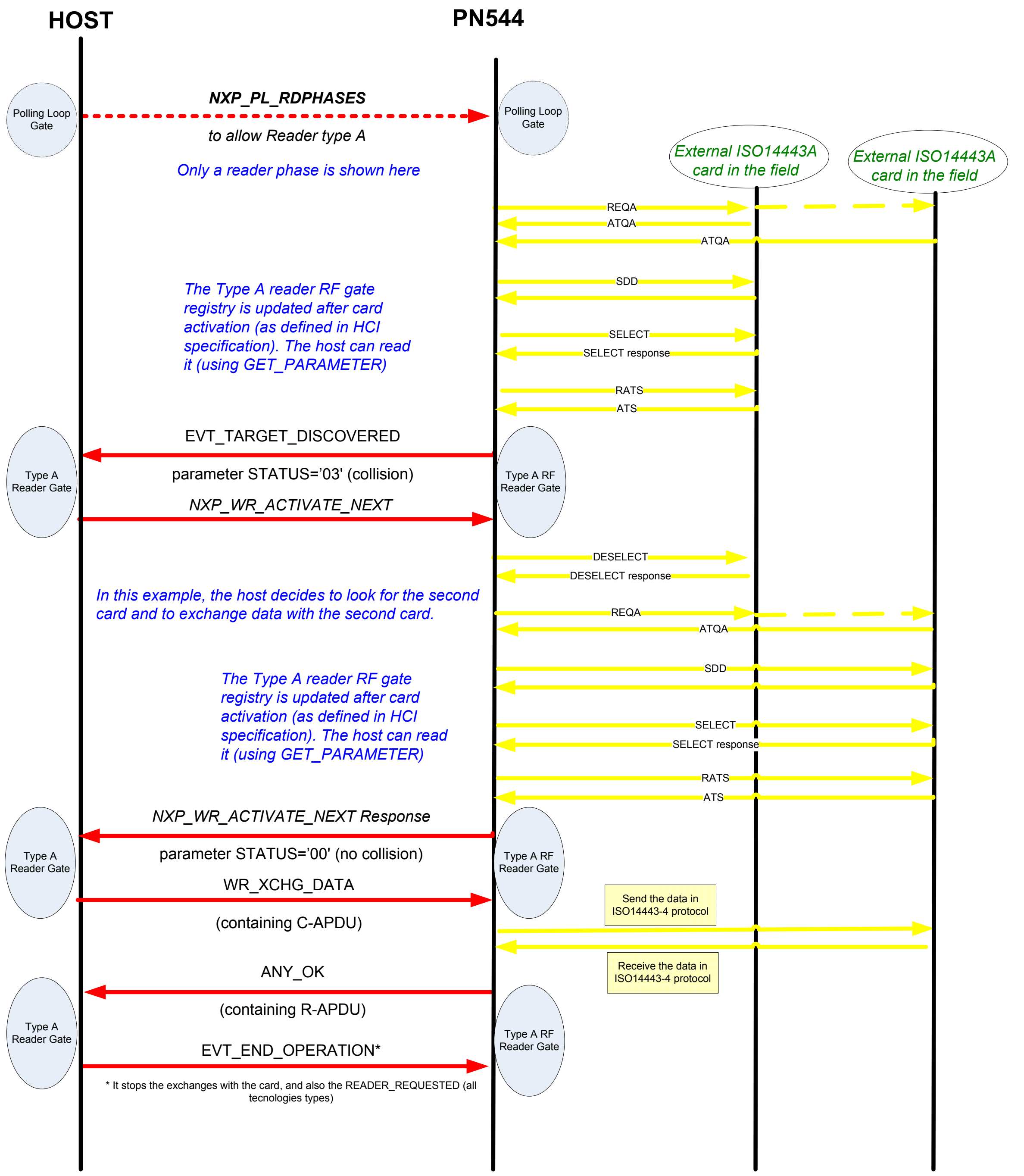

HOST
PN544
Polling Loop Gate
NXP_PL_RDPHASES
to allow Reader type A
Only a reader phase is shown here
Polling Loop Gate
External ISO14443A card in the field
External ISO14443A card in the field
REQA
ATQA
ATQA
SDD
The Type A reader RF gate registry is updated after card activation (as defined in HCI specification). The host can read it (using GET_PARAMETER)
SELECT
SELECT response
RATS
ATS
EVT_TARGET_DISCOVERED
Type A Reader Gate
parameter STATUS='03' (collision)
Type A RF Reader Gate
NXP_WR_ACTIVATE_NEXT
DESELECT
DESELECT response
In this example, the host decides to look for the second card and to exchange data with the second card.
REQA
ATQA
SDD
The Type A reader RF gate registry is updated after card activation (as defined in HCI specification). The host can read it (using GET_PARAMETER)
SELECT
SELECT response
RATS
ATS
NXP_WR_ACTIVATE_NEXT Response
Type A Reader Gate
parameter STATUS='00' (no collision)
Type A RF Reader Gate
WR_XCHG_DATA
Send the data in ISO14443-4 protocol
(containing C-APDU)
Receive the data in ISO14443-4 protocol
ANY_OK
Type A Reader Gate
(containing R-APDU)
Type A RF Reader Gate
EVT_END_OPERATION*
* It stops the exchanges with the card, and also the READER_REQUESTED (all tecnologies types)

Fig 69. Host as a Type A reader, two Type A cards in the field.

Several cards can be put in HALT. The parameter of EVT_TARGET_DISCOVERED indicates if there is still a collision (i.e. if there are still several cards not activated).

To reactivate a chosen card, its UID is used, in NXP_WR_ACTIVATE_ID command.

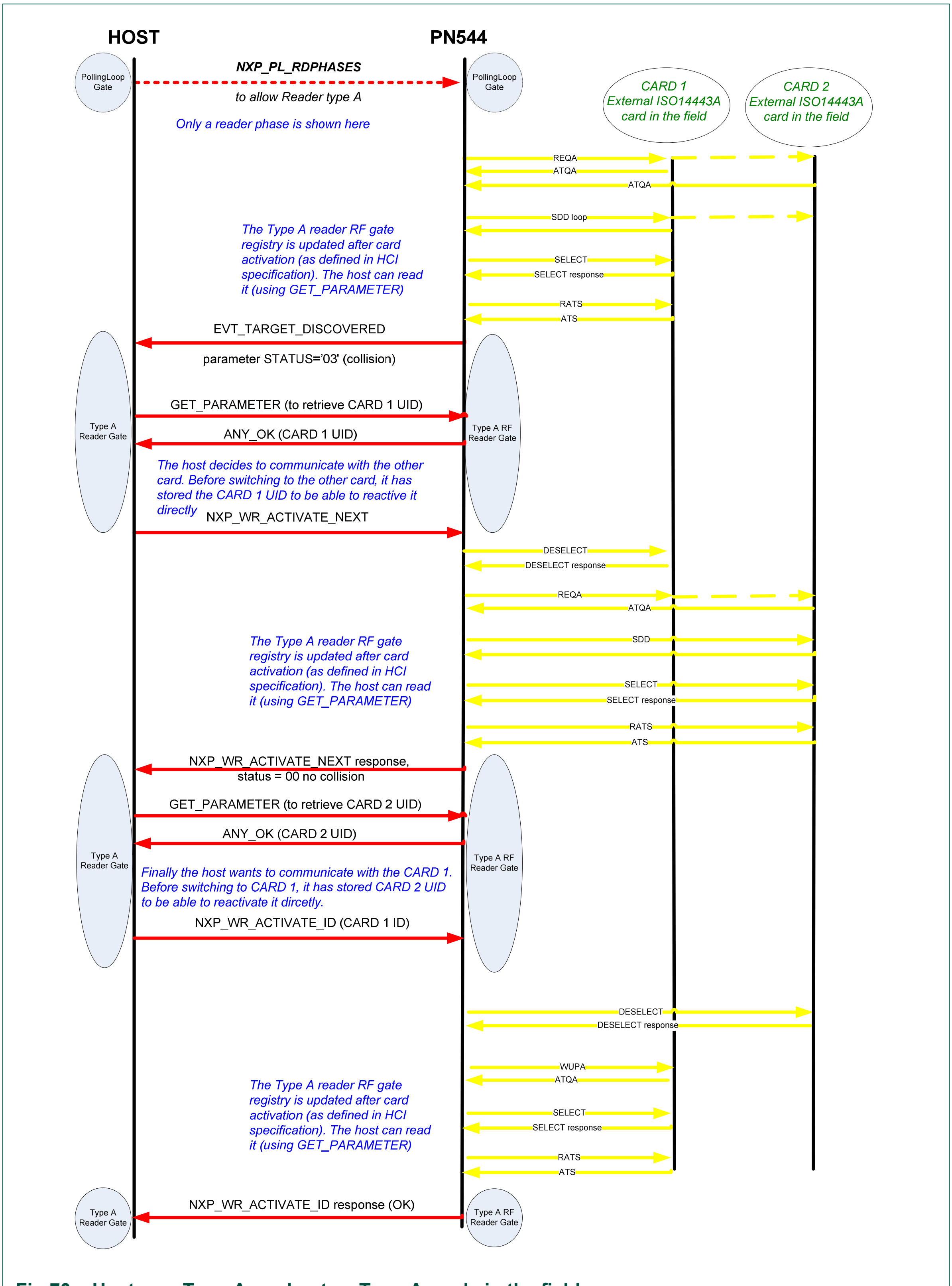

HOST
PN544
PollingLoop Gate
NXP_PL_RDPHASES
to allow Reader type A
Only a reader phase is shown here
PollingLoop Gate
CARD 1 External ISO14443A card in the field
CARD 2 External ISO14443A card in the field
REQA
ATQA
ATQA
SDD loop
The Type A reader RF gate registry is updated after card activation (as defined in HCI specification). The host can read it (using GET_PARAMETER)
SELECT
SELECT response
RATS
ATS
EVT_TARGET_DISCOVERED
parameter STATUS='03' (collision)
Type A Reader Gate
GET_PARAMETER (to retrieve CARD 1 UID)
ANY_OK (CARD 1 UID)
Type A RF Reader Gate
The host decides to communicate with the other card. Before switching to the other card, it has stored the CARD 1 UID to be able to reactive it directly
NXP_WR_ACTIVATE_NEXT
DESELECT
DESELECT response
REQA
ATQA
The Type A reader RF gate registry is updated after card activation (as defined in HCI specification). The host can read it (using GET_PARAMETER)
SDD
SELECT
SELECT response
RATS
ATS
NXP_WR_ACTIVATE_NEXT response, status = 00 no collision
Type A Reader Gate
GET_PARAMETER (to retrieve CARD 2 UID)
ANY_OK (CARD 2 UID)
Type A RF Reader Gate
Finally the host wants to communicate with the CARD 1. Before switching to CARD 1, it has stored CARD 2 UID to be able to reactivate it dircetly.
NXP_WR_ACTIVATE_ID (CARD 1 ID)
DESELECT
DESELECT response
WUPA
ATQA
The Type A reader RF gate registry is updated after card activation (as defined in HCI specification). The host can read it (using GET_PARAMETER)
SELECT
SELECT response
RATS
ATS
NXP_WR_ACTIVATE_ID response (OK)
Type A Reader Gate
Type A RF Reader Gate

Fig 70. Host as a Type A reader, two Type A cards in the field

9.15.3.2 Presence check of ISO14443A card: use NXP_WR_PRESCHECK

When the reader wants to check the presence of the ISO14443 card in the field, it shall send the command NXP_WR_PRESCHECK. In ISO14443, the PN544 will send a R(NAK) block to the card. It expect R(ACK) or last I block in response, as defined in ISO14443-4.

9.15.4 Handling of multiple Type A readers

Both the UICC and the host can activate a reader session at the same time.

The next figure shows a simplified view of the system.

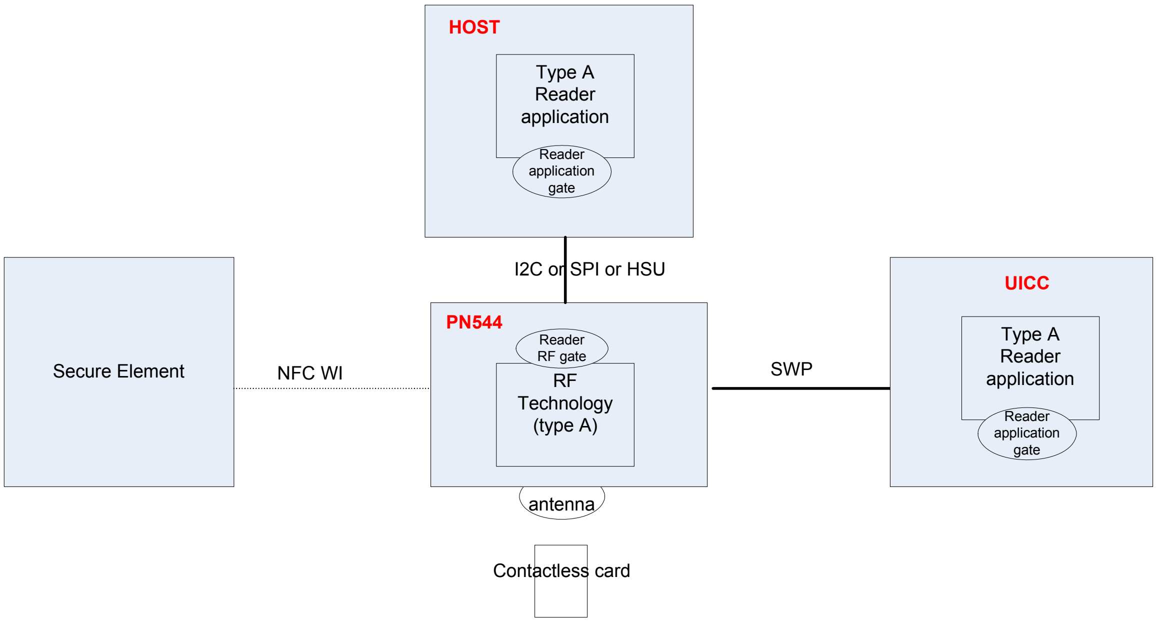


Fig 71. Simplified system view

Both the host and the UICC must configure their pipes as defined in HCI specification. The host must in addition set some NXP proprietary parameters to enable the UICC as a reader (refer to SWP paragraph). The next figure shows those exchanges between the host, the PN544 and the UICC.

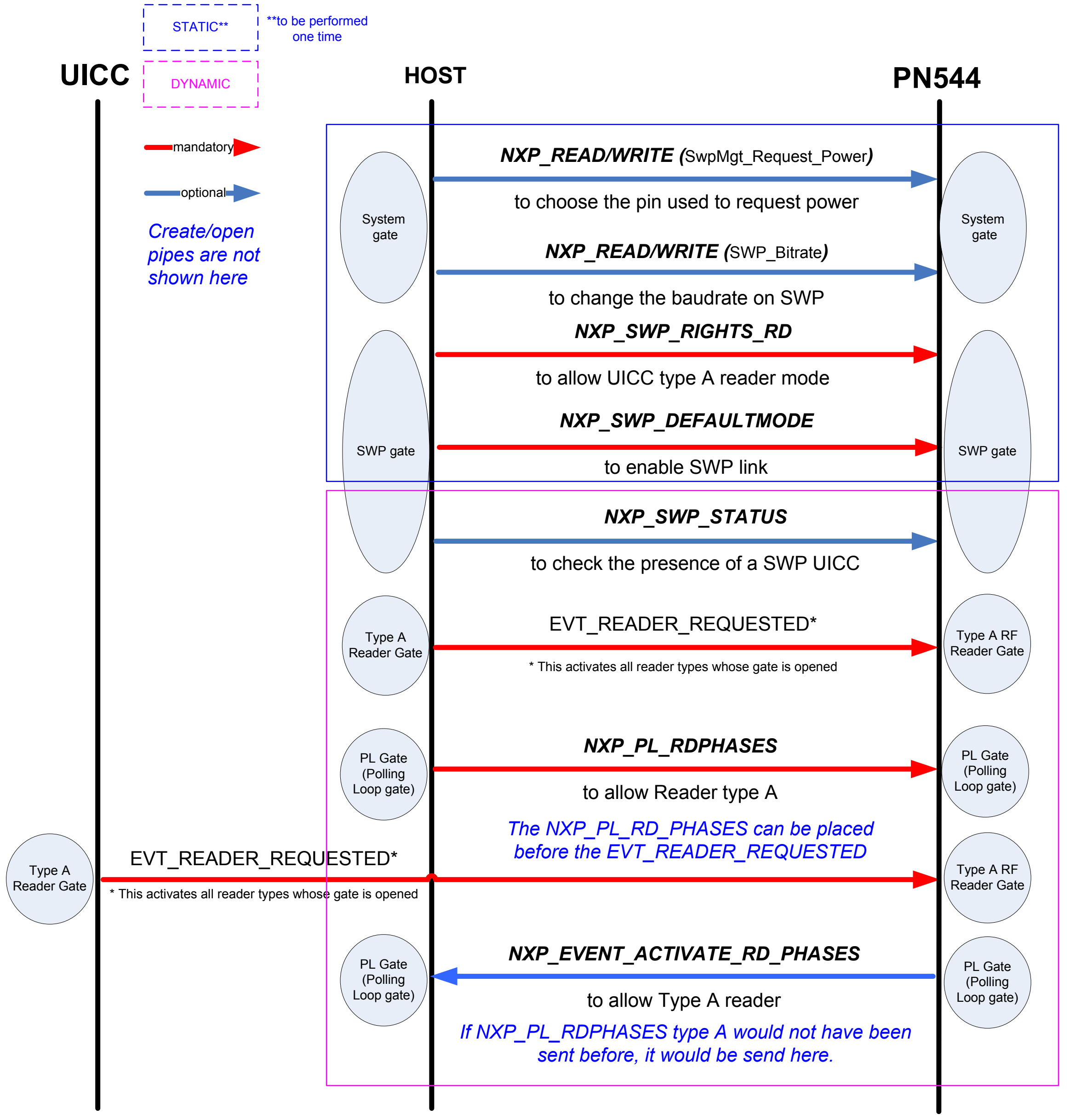

STATIC**
**to be performed one time
DYNAMIC
UICC
HOST
PN544
mandatory
optional
Create/open pipes are not shown here
System gate
System gate
NXP_READ/WRITE (SwpMgt_Request_Power)
to choose the pin used to request power
NXP_READ/WRITE (SWP_Bitrate)
to change the baudrate on SWP
NXP_SWP_RIGHTS_RD
to allow UICC type A reader mode
SWP gate
SWP gate
NXP_SWP_DEFAULTMODE
to enable SWP link
NXP_SWP_STATUS
to check the presence of a SWP UICC
Type A Reader Gate
Type A RF Reader Gate
EVT_READER_REQUESTED*
* This activates all reader types whose gate is opened
PL Gate (Polling Loop gate)
PL Gate (Polling Loop gate)
NXP_PL_RDPHASES
to allow Reader type A
The NXP_PL_RD_PHASES can be placed before the EVT_READER_REQUESTED
Type A Reader Gate
EVT_READER_REQUESTED*
* This activates all reader types whose gate is opened
Type A RF Reader Gate
PL Gate (Polling Loop gate)
PL Gate (Polling Loop gate)
NXP_EVENT_ACTIVATE_RD_PHASES
to allow Type A reader
If NXP_PL_RDPHASES type A would not have been sent before, it would be send here.

Fig 72. Initialization of PN544

The next figure shows what happen when a card is present in the external field.

The host is the first to receive the information that a card is present in the RF field. The UICC will be able to communicate with the card only after the host has sent *NXP_WR_DISPATCH_TO_UICC*.

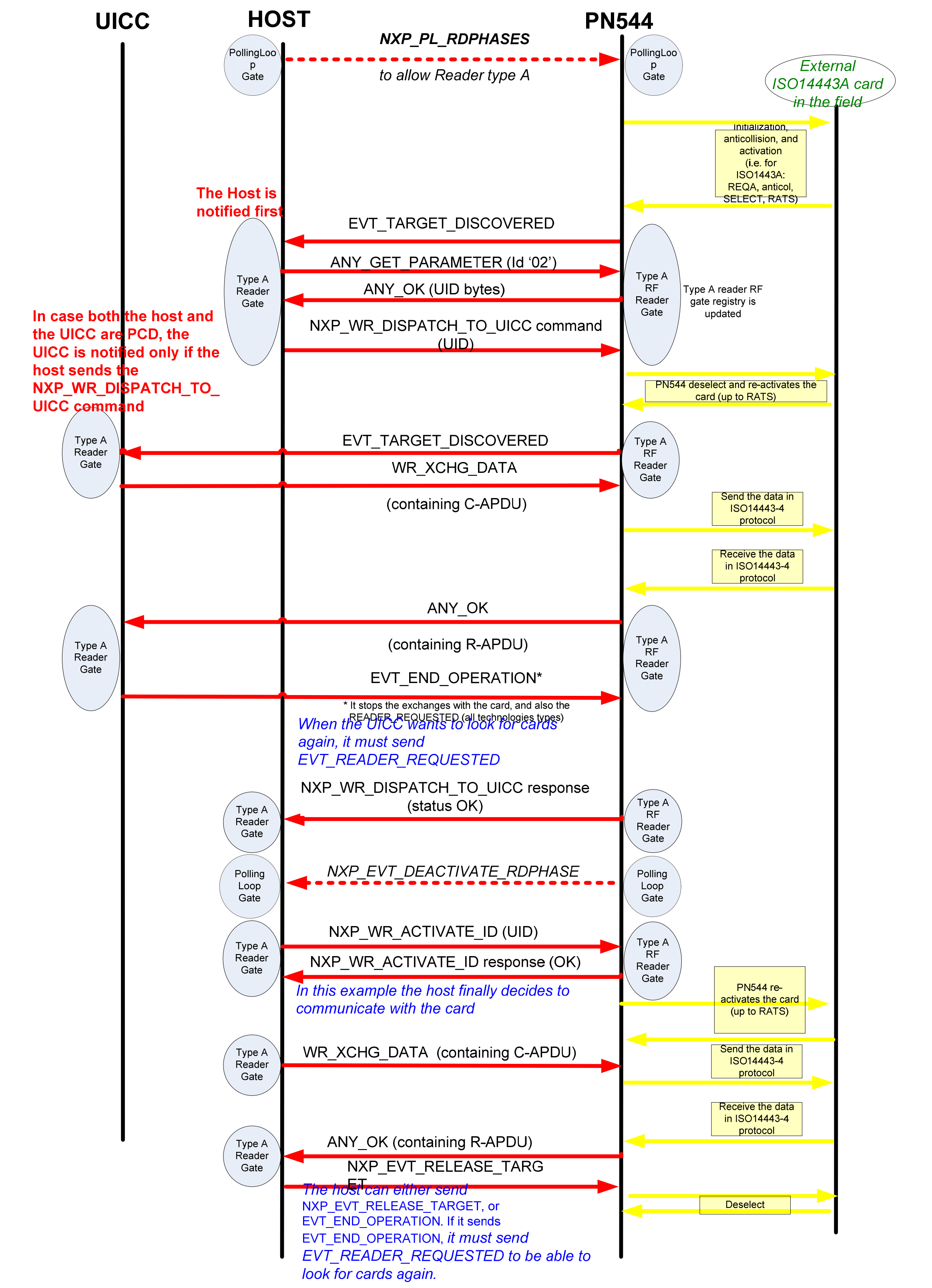

UICC
HOST
PN544
NXP_PL_RDPHASES
to allow Reader type A
PollingLoop Gate
PollingLoop Gate
External ISO14443A card in the field
Initialization, anticollision, and activation (i.e. for ISO1443A: REQA, anticol, SELECT, RATS)
The Host is notified first
EVT_TARGET_DISCOVERED
ANY_GET_PARAMETER (Id '02')
ANY_OK (UID bytes)
Type A Reader Gate
Type A RF Reader Gate
Type A reader RF gate registry is updated
NXP_WR_DISPATCH_TO_UICC command (UID)
In case both the host and the UICC are PCD, the UICC is notified only if the host sends the NXP_WR_DISPATCH_TO_UICC command
PN544 deselect and re-activates the card (up to RATS)
Type A Reader Gate
EVT_TARGET_DISCOVERED
Type A RF Reader Gate
WR_XCHG_DATA
(containing C-APDU)
Send the data in ISO14443-4 protocol
Receive the data in ISO14443-4 protocol
ANY_OK
Type A Reader Gate
(containing R-APDU)
Type A RF Reader Gate
EVT_END_OPERATION*
* It stops the exchanges with the card, and also the READER_REQUESTED (all technologies types)
When the UICC wants to look for cards again, it must send EVT_READER_REQUESTED
NXP_WR_DISPATCH_TO_UICC response (status OK)
Type A Reader Gate
Type A RF Reader Gate
NXP_EVT_DEACTIVATE_RDPHASE
Polling Loop Gate
Polling Loop Gate
NXP_WR_ACTIVATE_ID (UID)
NXP_WR_ACTIVATE_ID response (OK)
Type A Reader Gate
Type A RF Reader Gate
PN544 re-activates the card (up to RATS)
In this example the host finally decides to communicate with the card
WR_XCHG_DATA (containing C-APDU)
Type A Reader Gate
Send the data in ISO14443-4 protocol
Receive the data in ISO14443-4 protocol
ANY_OK (containing R-APDU)
Type A Reader Gate
NXP_EVT_RELEASE_TARGET
Deselect
The host can either send NXP_EVT_RELEASE_TARGET, or EVT_END_OPERATION. If it sends EVT_END_OPERATION, it must send EVT_READER_REQUESTED to be able to look for cards again.

Fig 73. Communication with a card

9.16 Type B PICC

This paragraph explains in which ways the PN544 can behave like a Type B card.

9.16.1 Type B PICC over SWP (UICC as a Type B card)

The Type B card emulation can be located in the UICC connected to PN544 over SWP link.

Refer to paragraph SWP.

9.16.2 Type B PICC over host link (I2C, SPI, HSU) (host as a Type B card)

The Type B card emulation can be located in the host connected to PN544 over I2C, SPI or HSU link.

The host is not allowed to transmit a PUPI. The PUPI in that case is randomly generated by the PN544.

9.16.3 Multiple Type B PICC

PN544 cannot emulate two type B cards at the same time.

It is not possible to emulate at the time a Type B card in the UICC and in the host. (The Create Pipe, on an already in use card RF gate, will return an error). The host decides to allow or not the UICC to emulate a Type B card. If it allows the UICC emulate a Type B card, then it cannot itself emulate a Type B card.

Note: Refer to '***Type A PICC***' flowcharts to have an overview of system exchanges. (Same HCI events/registry/commands are required on ***Type B Card RF Gate***)

Note: In Type B PICC mode, the HIGHER_LAYER_RESPONSE registry has a maximum size of 31 bytes. PN544 does not support HIGHER_LAYER_RESPONSE registry size between 32 and 252 bytes.

No visible impact is foreseen at end user level.

ETSI/HCI test specification Rev2.4 proposes a test with 10 bytes.

Type B PCD

This paragraph explains how to realize a PN544 and its host(s) can be a Type B reader.

WARNING: Type B PCD does not support multiple card detection (Anti-collision not supported). Thus only one Type B card can be detected by PN544.

9.16.4 Type B PCD over SWP (UICC as a Type B reader)

The UICC can run a Type B reader application, as described in HCI specification.

Refer to paragraph SWP.

9.16.5 Type B PCD over host link (I2C, SPI, HSU) (Host as a Type B reader)

The host processor can run a Type B reader application, as described in HCI specification.

The method used is fixed in ROM. The method used is the ***probabilistic*** approach.

9.16.6 Meaning of additional commands of Reader RF gates for Type B

Refer to Reader RF gates additional commands chapter for commands description.

9.16.6.1 Several Type B cards in the RF field: use NXP_WR_ACTIVATE_NEXT

Not Applicable (Type B Anti-collision not supported).

9.16.6.2 Presence check of ISO14443B card: use NXP_WR_PRESCHECK

When the reader wants to check the presence of the ISO14443 card in the field, it shall send the command NXP_WR_PRESCHECK. In ISO14443, the PN544 will send a R(NAK) block to the card. It expect R(ACK) or last I block in response, as defined in ISO14443-4.

9.16.7 Handling of multiple Type B readers

Both the UICC and the host can activate a reader session at the same time.

Note: Refer to '***Type A PCD***' flowcharts to have an overview of system exchanges. (Same HCI events/registry/commands are required on ***Type B Reader RF Gate***)

9.17 GPIO(s)

The host can change the **G**eneral **P**urpose **I**nput **O**utput pins configuration

- * Either by accessing directly the hardware registers (in that case the configuration will be lost after a power down)
- * Or by setting some EEPROM areas. The EEPROM values will be loaded in the chip at power up.
-

There are 8 GPIOs named GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, GPIO5, GPIO6 and GPIO7. Some of them can have a special function:

GPIO1 is Clock acknowledge when Clock acknowledge by pin is used (see chapter "***Acknowledge through CLKACK pin***").

GPIO2 is Clock request when Clock acknowledge by pin is used or Power request when Power request is used (see chapter "***Request through CLKREQ pin***" and parameter SwpMgt_Request_Power in chapter "***SWP configuration***").

GPIO3 is Power request when Power request is used (see parameter SwpMgt_Request_Power in chapter "***SWP configuration***").

GPIO4 is used at boot to possibly enter the Download mode (see paragraph "Download").

GPIO5 has to be kept HIGH during PN544 boot.

The default configuration of the GPIOs port is described in [7].

The PN544 NXP_READ and NXP_WRITE commands (see $ HCI Proprietary gate 'PN544Mgt') can be used to change the default configuration of the GPIOs (directly in the hardware registers or in EEPROM).

9.17.1 GPIO Direct Hardware register access

See chapter "***PN544 register access / GPIO settings***"

9.17.2 GPIO EEPROM settings access

See ***'0*** chapter.

9.17.3 Examples

In the following descriptions, "permanently" means that the change is applied in EEPROM. Therefore the new value is only applied at next power up.

“Temporarily” means that the change is applied directly in the register of the hardware, therefore the new value is immediately applied, but will be lost after power down (at next power up, the EEPROM values will be applied)

9.17.3.1 Output configuration

To configure permanently GPIO5 in output

- Set *0x9893* GPIO_Config_PDIR bit 5 to 1.
- The value is stored in EEPROM; it will be loaded at next power up of the IC.
- In order not to change the other GPIO configuration, it is advised to first read the GPIO_Config_PDIR byte, to apply then a new value which changes only the desired bit.

To configure temporarily GPIO5 in output

- Set *0xF821* PDIR bit 5 to 1. It writes directly into the hardware register.
- In order not to change the other GPIO configuration, it is advised to first read the PDIR byte, to apply then a new value which changes only the desired bit.

9.17.3.2 Read/write an output pin

To set with immediate effect GPIO5 to 1 (GPIO5 is considered in this example configured in output)

- Set 0xF82B POUT bit 5 to 1
- In order not to change the other output GPIOs values, it is advised to first read the POUT register, to apply then a new value which changes only the desired bit.

To set with immediate effect GPIO5 to 0 (GPIO5 is considered in this example configured in output)

- Set 0xF82B POUT bit 5 to 0
- In order not to change the other output GPIOs values, it is advised to first read the POUT register, to apply then a new value which changes only the desired bit.

To set GPIO5 to 1 at start up (GPIO5 is considered in this example configured in output)

- Set 0x9890 GPIO_Config_POUT bit 5 to 1

To set GPIO5 to 0 at start up (GPIO5 is considered in this example configured in output)

- Set 0x9890 GPIO_Config_POUT bit 5 to 0
-

To read an output pin value, e.g. GPIO5 considered configured in output

- Read 0xF82B POUT. GPIO5 value is bit 5 of the register.

9.17.3.3 Input configuration

To configure permanently GPIO5 in input

- Set *0x9893* GPIO_Config_PDIR bit 5 to 0.
- Set *0x9899* GPIO_Config_PEN bit 5 to 1.
- The value is stored in EEPROM; it will be loaded at next power up of the IC.
- In order not to change the other GPIO configuration, it is advised to first read the GPIO_Config_PDIR and GPIO_Config_PEN bytes, to apply then new values which change only the desired bit.

To configure temporarily GPIO5 in input

- Set *0xF821* PDIR bit 5 to 0. It writes directly into the hardware register.
- Set *0xF829* PEN bit 5 to 1. It writes directly into the hardware register
- In order not to change the other GPIO configuration, it is advised to first read the PDIR and PEN bytes, to apply then new values which change only the desired bit.

9.17.3.4 Read an input pin (write is not possible)

To read an input pin value, e.g. GPIO5 considered configured in input

- Read *0xF82A* PIN. GPIO5 value is bit 5 of the register.

9.17.3.5 Connect internal Pull up or pull down to the GPIO

To connect permanently a pull up to e.g GPIO5 (the configuration will be kept even after a reset)

- Set 0x9894 GPIO_Config_UPUD bit 5 to 1.

To connect temporarily a pull up to e.g GPIO5 (the configuration will be lost after a reset)

- Set *0xF823* UPUD bit 5 to 1.

To connect permanently a pull down to e.g GPIO5 (the configuration will be kept even after a reset)

- Set 0x9895 GPIO_Config_ DPUD bit 5 to 1.

To connect temporarily a pull down to e.g GPIO5 (the configuration will be lost after a reset)

- Set *0xF822* DPUD bit 5 to 1.

9.17.3.6 Inverse polarity

To inverse temporarily the polarity of a GPIO port, the corresponding bit to 1 in *0xF820* PINV register.

To inverse permanently the polarity of a GPIO port, the corresponding bit to 1 in *0x9892* GPIO_Config_PINV register.

9.18 Download

This chapter describes the EEPROM download of PN544.

The Download mode use a specific protocol (different from LLC/HCI) described in [8]. The only way to leave the Download mode is to perform a reset (through download mode reset command or HW reset).

9.18.1 Switch in ‘Download’ mode

To switch in this mode, the *GPIO4* has to be set to *PVDD* and PN544 must be reset (using VEN).

As soon as PN544 has switched in ‘Download’ mode, it can not leave it without a specific command (Reset command of this mode) or a Hardware Reset.

To perform this SW update, NXP will provide binary files for EEPROM update. On Host controller side, binary file content needs to be written to PN544 EEPROM using the Download protocol as follows:

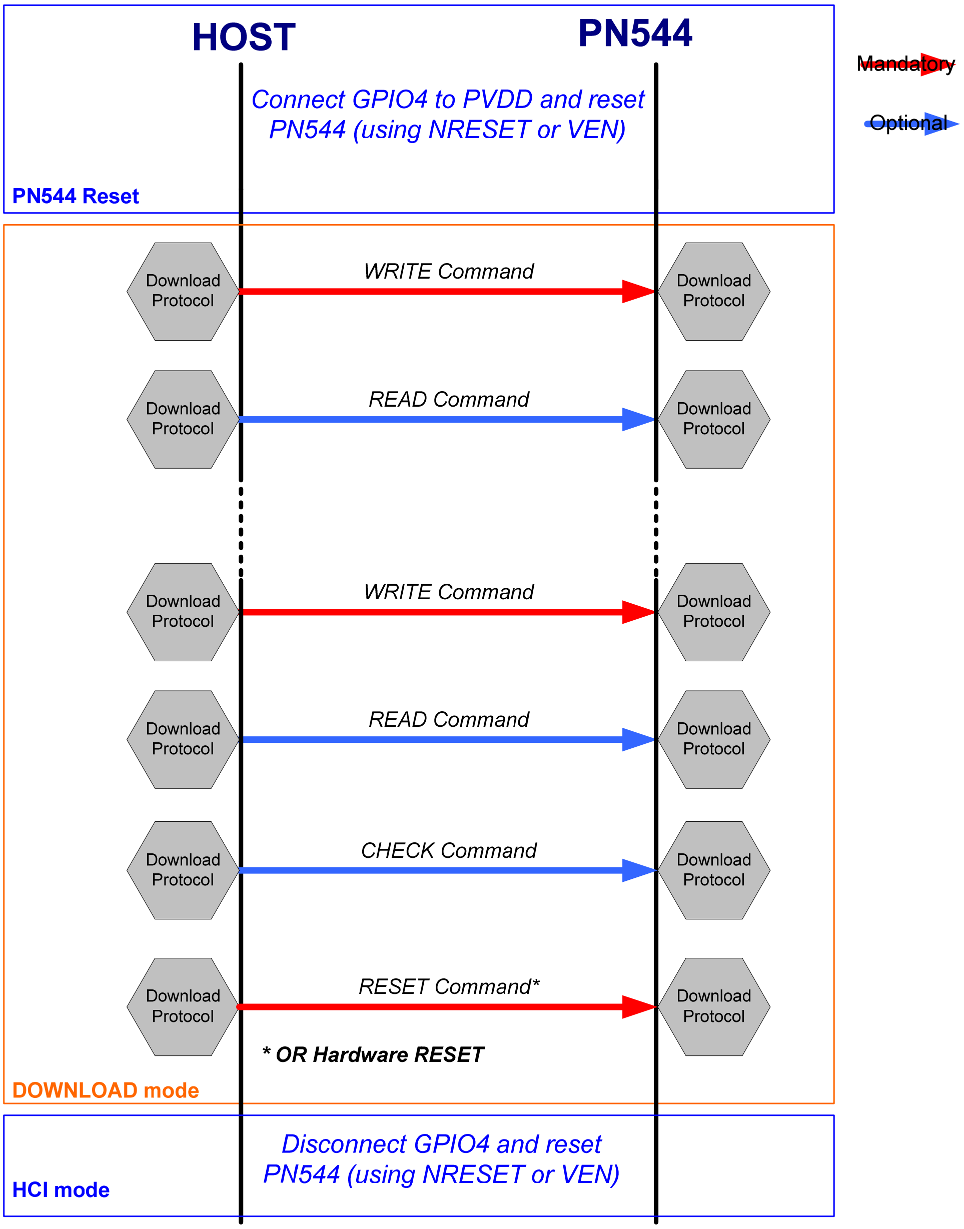


Fig 74. Switch to Download mode

Note: *WRITE*, *READ*, *CHECK* & *RESET* command are part of the Download protocol documented in the dedicated Application Note [8].

WRITE command is mandatory to use,

READ & *CHECK* commands are optional to cross check the consistency of the EEPROM.

RESET command is used to go out of the Download mode; this command can be replaced by a Hardware reset

9.19 PN544 Debug Mode

This chapter describes the debug functionality of PN544 (i.e. Output signal on GPIO).

To setup the PN544 in debug mode, the following register has to be used:

Table 129. Debug_Interface Register

Address	*Name*	*Comment*	*Default*
0xF830	*Debug_Interface*	*Debug_Interface Number* *(default value 0x00 = No debug mode)*	0x00

Using the *Debug_Interface* register, GPIO as output (See ***GPIO chapter***) and setting EEPROM Data area (See ***'0** chapter*), the following debug mode can be used.

9.19.1 SWP digitized

To allow outputting SWP RX signal digitized to GPIO7, the following actions are needed:

-1- Set *ANAIRQ_Conf - bit5* to 1 (i.e. 0x30).

-2- Reset PN544.

-3- Set GPIO7 direction to Output.

-4- Set *Debug_Interface* to 0x75.

Then on GPIO7, the SWP RX signal (S2 from UICC to PN544) digitized can be seen.

To disable this setup:

-1- Set *ANAIRQ_Conf - bit5* to 0 (i.e. 0x10).

-2- Reset PN544.

-3- Set GPIO7 to its default configuration (depending of application use).

9.19.2 PLL Lock & System clock

PLL Lock and System clock signals can be output to GPIO7 and GPIO5.

To allow it, the following actions are needed:

-1- Set GPIO7 and GPIO5 direction to Output.

-2- Set *Debug_Interface* to 0x68.

Then on GPIO7, the PLL Lock state can be seen (1 = lock, 0 = unlock).

On GPIO5, the PN544 System Clock is output, 27.12 MHz.

To disable this setup:

-1- Perform a Hardware Reset (VEN pin use).

-2- Set GPIO7 and GPIO5 direction to their default configuration (depending of application use).

9.19.3 RF Level Detector

To output RF Level Detector state to GPIO7, the following actions are needed:

-1- Set GPIO7 direction to Output.

-2- Set *Debug_Interface* to 0x44.

Then on GPIO7, the RF Level Detector is seen (1 = RF detected, 0 = No RF detected).

For further details on this signal, refer to [11] & [7].

To disable this setup:

-1- Perform a Hardware Reset (VEN pin use).

-2- Set GPIO7 direction to its default configuration (depending of application use).

9.19.4 RF envelope TX

To output RF TX envelope to GPIO6, the following actions are needed:

-1- Set GPIO6 direction to Output.

-2- Set *WI_SE_SigoutSel* to 0x02 (TX envelope).

-3- Enable a Reader phase using HCI commands.

-4- Set *Debug_Interface* to 0x14.

For further details on this signal, refer to [11].

To disable this setup:

-1- Perform a Hardware Reset (VEN pin use).

-2- Set *WI_SE_SigoutSel* to its default value (0x03).

-3- Set GPIO6 direction to its default configuration (depending of application use).

9.19.5 RF Signal RX

To output RF RX signals state to GPIO(s), the following actions are needed:

-1- Set GPIO(s) direction to Output.

-2- Set *Debug_Interface* to 0x01.

For further details on this signal, refer to [11].

To disable this setup:

-1- Perform a Hardware Reset (VEN pin use).

-2- Set GPIO(s) direction to its default configuration (depending of application use).

9.20 PN544 Configuration

PN544 can be customized thanks to the data EEPROM. The following settings can be accessed using *NXP_READ* and *NXP_WRITE* commands (Refer to ***'PN544 System Management'*** chapter).

9.20.1 SWP configuration

Table 130. SWP configuration in EEPROM

Address	*Name*	*Comment*	*Default*
0x9C01	SWP_Bitrate	Indicates the bitrates to be used on SWP 'according to UICC capability). Supported bitrates are: 0x02 -> 212 kbit/s 0x04-> 424 kbit/s 0x08-> 848 kbit/s 0x10-> Maximum bitrate. Others values are RFU	0x08
0x9C04	SWP_MiscConf	- Bit0: Bit1: force the CRC to be OK for each frame received - Bit3: if equal to 1 the RNR will be sent after ClearAllPipe command (and other long processing command) instead of RR - Bit4: if equal to 1 the soft reset in case of too many errors on RX is activated - Bit5: if equal to 1 the soft reset in case of too many errors on TX is activated - Others Bits are RFU	0x30
0x9EB4	SwpMgt_Request_Power	Indicates if the power needs to be requested before a communication to the UICC can be started in case the voltage on PMUVcc is already there but the PMU is switched to low power mode. - 0x00 -> no request - 0x01 -> request through CLKREQ pin (GPIO pin 2 set to high when clock requested, Hiz unless, whatever GPIO default configuration) - 0x02 -> request through PWR_REQUEST pin (GPIO pin 3 set to high when clock requested, Hiz unless, whatever GPIO default configuration) - Others values are RFU	0x00
0x9C00	SWP_CurrentThreshold	0x02 -> 240 µA 0x12-> 260 µA 0x22-> 300 µA 0x32-> 330 µA *Others values are RFU*	*0x22*

Address	*Name*	*Comment*	*Default*
0x9F0A	SWP_PBTF_RFLD	SWP PowerByTheField RF Level Detector sensitivity: Value to be set from 0x00 to 0x07. 0x00: Maximum sensitivity (to detect Low RF field). 0x07: Minimum sensitivity.	*0x05*
0x9ED7	*UICC_GateList*	Registry entry related to the Gate list (IdentityManagement gate) of the Uicc. Can be used to define UICC rights. For example, to allow Card type B and Reader Type A feature, related gate Ids 0x21 and 0x13 must be present (at any place, list order having no impact) in this list.	*0x04*
0x9ED8			0x05
0x9ED9			0xFF
0x9EDA			0xFF
0x9EDB			0xFF
0x9EDC			0xFF
0x9EDD			0xFF
0x9EDE			0xFF
0x9EDF			0xFF
0x9EE0			0xFF
0x9EE1			0xFF
0x9EE2			0xFF
0x9EE3			0xFF
0x9EE4			0xFF
0x9EE5			0xFF
0x9EE6			0xFF
0x9EE7			0xFF
0x9EE8			0xFF

Address	*Name*	*Comment*	*Default*
0x9EE9			0xFF
0x9EEA			0xFF
0x9C02	*SWP_SyncID0*	Reference value for checking the UICC identity during ACT SYNC_ID verification sequence (SWP activation). Can be used to inhibit the UICC.	0x13
0x9C03	*SWP_SyncID1*		0x13
0x9EA2	*UICC_AdminSessionId*	Registry entry related to SessionIdentity (Administration gate) of the UICC. Can be used to force UICC to re-create its context.	0xFF
0x9EA3			0xFF
0x9EA4			0xFF
0x9EA5			0xFF
0x9EA6			0xFF
0x9EA7			0xFF
0x9EA8			0xFF
0x9EA9			0xFF

Table 131. SWP configuration register(*)

Address	*Name*	*Comment*	*Default*
0xF841	SWIO_HighZ	Register to set SWP SWIO line in HighZ mode. -Bit 0 : If set to 1, the SWP Pad is in Power Down mode (HighZ). -Bit 1-7 : RFU (keep unchanged)	0x*10*

(*) Register, volatile memory, valid until a power off/up sequence is performed or until Hardware Reset (VEN pin use).

9.20.2 SE configuration

Table 132. SE Configuration in EEPROM

SWP-SIM卡模拟配置

Address	Name	Comment	Default
0x9800	*SE_Conf*	*Defines the properties of Secure Elements (UICC and NFC-WI device)* *-Bit 0 : NFC-WI device connected to PN54x* 0 *-Bit 1 : UICC/SIM card connected to PN54x* 1 *-Bit 2 : Powering of UICC enabled* 1 *-Bit 3 : RFU (keep unchanged)* 1 *-Bits 4 - 5 : External Power Selection for PbtF-mode* *00 ... SIM/UICC* 0 *01 ... SE* 0 *10-11 ... No device* *-Bit 6 : RFU (keep unchanged)* 0 *-Bit 7 : RFU (keep unchanged)* 0	0x3F
0x9E75	*WI_SE_SigoutSel*	*Defines which signal is put to SigOut when in virtual mode (in wired mode this parameter is fixed to 0x02 -> TX envelope)* *0x00 -> Low* *0x01 -> High* *0x02 -> TX envelope* *0x03 -> MILLER enveloppe (ISO14443A @106kbit/s)* *0x04 -> MANCHESTER envelope (FeliCa)* *0x05 -> NRZ envelope* *Others values are RFU*	0x03
0x9E76	*WI_SE_SigoutClkSel*	*Defines how the 13.56MHz clock is combined to the SigOut signal.* *- 0x00 -> Plain signal on SigOut, the clock is not combined* *- 0x20 -> Clock is XORed to the SigOut signal (ISO14443B, FeliCa)* *- 0x40 -> Clock is ANDed to the SigOut signal (ISO14443A @106kbit/s)* *- Others values are RFU*	0x40
0x9B69	*NfcWi_SE_SigInDelay*	*Indicates whether the PN544 apply a delay when transmitting data from Sigin to RF or not* *- 0x00 -> No delay inserted* *- 0x01 -> Delay inserted*	0x01
0x9B4A	*NfcWi_SE_TimeOut_H*	*Timeout (in milliseconds) value of inactivity on Sigin line to detect EndOfTransaction. If set to 0, it means that timeout is not used to detect the end of the transaction.*	0x00

Address	*Name*	*Comment*	*Default*
0x9B4B	*NfcWi_SE_TimeOut_L*		0x00

9.20.3 HW configuration

Table 133. HW configuration in EEPROM

Address	Name	Comment	Default
0x9810	HW_Conf	TX-LDO configuration -Bits 0-1 : TVDD Supply select for TX-LDO 00 ... 3.0 V 01 ... 3.0 V 10 ... 2.7 V 11 ... 3.3 V -Bit 2 : Enables offset for TX-LDO, if set to 1 -Bit 3 : Power down TX-LDO in Card-emulation mode, if set to 1 -Bit 4 : RFU (keep unchanged) -Bit 5 : Enable TX-LDO limiter, if set to 1 CLOCK configuration -Bit 7 : 0 ... Use internal FracNPLL for clock generation 1 ... Use external crystal for clock generation	*0xBC*
0x9801	ANAIRQ_Conf	Analog-IRQ Configuration -Bits 0-1 : RFU (keep unchanged) -Bit 2 : Enable/Disable PMU-Vcc IRQ -Bit 3 : RFU -Bit 4 : Enable/Disable RF-Modulation Tuning -Bit 5 : Enable/Disable SWP-Debug during boot-process -Bits 6-7 : RFU	*0x17*
0x9F9D	TX_Current_Check	TX Overcurrent Protection Set to 0x00 to disable the TX over current protection	*0x02*
0x9805	AST-TX1pass	Antenna Self Test: RFLD pass threshold on TX1 (1st parameter)	*0x03*
0x9806	AST-TX2pass	Antenna Self Test: RFLD pass threshold on TX2 (2nd parameter)	*0x05*
0x9807	AST-Current	Antenna Self Test: Current pass threshold (3rd parameter)	*0x0E*

Address	Name	Comment	Default
0x9808	AST-ANT1-ANT2	Antenna Self Test: default PBF (4th parameter)	*0xFF*
0x989D	AST-GSP2	Antenna Self Test: GSP for TX2	*0x27*
0x989E	AST_Current	Antenna Self Test: Current start threshold	*0x3F*
0x989F	AST-GSP1	Antenna Self Test: GSP for TX1	*0x2D*
0x9890	GPIO_Config_POUT	Gpio default configuration: POUT (Output Port Latch) Read or write the output port value. Each bit in the configuration register represents the GPIO pin with the corresponding number (default value 0x00: output pins are set to 0)	*0x00*
0x9891	GPIO_Config_PIN	Gpio default configuration: PIN (Input Port Value) Each bit in the configuration register represents the GPIO pin with the corresponding number Read the input port value	*0x00*
0x9892	GPIO_Config_PINV	Gpio default configuration: PINV (Port Invert) Port inversion register Each bit in the configuration register represents the GPIO pin with the corresponding number	*0x00*
0x9893	GPIO_Config_PDIR	Gpio default configuration: PDIR (Port Direction) Each bit in the configuration register represents the GPIO pin with the corresponding number If set to 1, the corresponding GPIO is configured in output (whatever PEN value). If set to 0 and if the corresponding PEN bit is set to 1, the corresponding GPIO is configured in input.	*0x00*
0x9894	GPIO_Config_UPUD	Gpio default configuration: UPUD (Pull up enable register) Each bit in the configuration register represents the GPIO pin with the corresponding number Set to 1, an internal pull up is connected to the corresponding GPIO.	*0x03*
0x9895	GPIO_Config_DPUD	Gpio default configuration: DPUD (Pull Down Enable Register) Each bit in the configuration register represents the GPIO pin with the corresponding number Set to 1, an internal pull down is connected to the corresponding GPIO	*0x00*

Address	Name	Comment	Default
0x9898	IRQ_Config	- Bit 5 : 0 ... IRQ logical level is active HIGH (Default value) 1 ... IRQ logical level is active LOW - Other Bits : RFU (keep unchanged)	*0x02*
0x9899	GPIO_Config_PEN	Gpio default configuration: PEN (Port Enable) Each bit in the configuration register represents the GPIO pin with the corresponding number If set to 1, and if the corresponding PDIR is set to 0, the corresponding GPIO is configured in input	*0x00*
0x9E71	PlClockRequest	Indicates how the clock is requested to the host by the PN544. - 0x00 -> no request - 0x01 -> request through CLKREQ pin (GPIO pin 2 set to high when clock requested, HighZ unless, whatever GPIO default configuration) - 0x02 -> request through NXP_EVT_CLK_REQUEST event - Others values are RFU	*0x00*
0x9E72	PlClockAck	Indicates how the clock request is acknowledged by the host. Clock acknowledgment is only valid in case of Clock request mechanism enabled (see NXP_PlClockRequest). - 0x00 -> acknowledged thanks to timeout (see PlClockTimeout) - 0x01 -> acknowledged through CLKACK pin (GPIO pin 1) - 0x02 -> acknowledged through NXP_EVT_CLK_ACK event - Others values are RFU	*0x00*
0x9E6F	PlClockTimeout	Indicates the timeout value to be used for clock request acknowledgment (up to 255ms in 1ms steps). Recommended value is 10 ms or less.	*0x0A*
0x9E74	PlRfLowPower	- Bits 0 to 2: Sensitivity of the RF Low Power Mode (value between 0 and 6, 0 -> sensitivity maximal, 6 -> sensitivity minimal) - Bits 3 to 6 are RFU - Bit 7: use of the RF Low Power Mode (0 -> not used, 1 -> used)	*0x00*

Address	Name	Comment	Default
0x9809	FRAC_ClkSel	Input Clock selection for FracNPLL -00h ... No input clock -01h ... 13.0 MHz -02h ... 19.2 MHz -03h ... 26.0 MHz -04h ... 38.4 MHz -05h ... custom Other values are RFU	*0x00*
0x980A	FRAC4_DIV	Customizable FracNPLL set	*0x04*
0x980B	FRAC4_OOF0		*0xCF*
0x980C	FRAC4_OOF1		*0xF3*
0x980D	FRAC4_OOF2		*0x00*
0x980E	FRAC4_CAL0		*0x4C*
0x980F	FRAC4_CAL1		*0x54*
0x9F12	PlMgtPassiveGTA	Guard time (in ms) to be used in detection type A phase	*0x06*
0x9F13	PlMgtPassiveGTB	Guard time (in ms) to be used in detection type B phase	*0x06*
0x9F14	PlMgtPassiveGTF	Guard time (in ms) to be used in detection type F phases 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	*0x14*
0x9F15	PlMgtActiveGTA	Guard time (in ms) to be used in detection NFC active phase	*0x06*
0x9F16	PlMgtVicinityGT	Guard time (in ms) to be used in detection ISO15693 phase 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	*0x01*
0x9F19	TO_Before_SDTBY_MSB	Timeout used to wait after last host/Uicc communication before going into standby (from 0 to 3.145s in 48µs step)	*0x60*
0x9F1A	TO_Before_SDTBY_LSB		*0x00*

Address	Name	Comment	Default
0x9EAA	PWR_STATUS	Indicates PN544 power modes used: 0x00 -> PN544 stays in active bat mode (except when generating RF field) 0x01 -> PN544 goes in standby when possible otherwise stays in active bat mode 0x02 -> RFU. 0x03 -> RFU.	*0x00*
0x9C0C	Host_TIC_MSB_RX	T_{IC}, inter-character timeout in Reception (in 3µs step) (When Host is writing to PN544) (0x0000 means T_{IC} mechanism disabled, 0x0096 = 0,45ms)	*0x00*
0x9C0D	Host_TIC_LSB_RX		*0x96*
0x9C12	Host_TIC_MSB_TX	T_{IC}, inter-character timeout in Transmission (in 3µs step) (When Host is reading from PN544) (0x0000 means T_{IC} mechanism disabled, 0x203A = 25ms)	*0x20*
0x9C13	Host_TIC_LSB_TX		*0x3A*
0x9C27	*AckHostTimeout_MSB*	PN544 Ack Timeout on Host link. (in 1ms step) (0x0005 = 5ms)	*0x00*
0x9C28	*AckHostTimeout_LSB*		*0x05*
0x9C31	*GuardHostTimeoutMSB*	Guard Host Timeout on Host link. (in 1ms step) (0x0000 means resend mechanism disabled, 0x0032 = 50ms)	*0x00*
0x9C32	*GuardHostTimeoutLSB*		*0x32*
0x9C19	*Host_RX_Retry*	Maximum consecutive retries on host interface RX path before deactivating (soft reset)	*0x0A*
0x9C1A	*Host_TX_Retry*	Maximum consecutive retries on host interface TX path before deactivating (soft reset)	*0x0A*
0x9C1B	*SWP_RX_Retry*	Maximum consecutive retries on SWP RX path before deactivating (soft reset)	*0x0A*
0x9C1C	*SWP_TX_Retry*	Maximum consecutive retries on SWP TX path before deactivating (soft reset)	*0x0A*
0x9C18	SWP_Act_Retry	Maximum consecutive retries to activate SWP connection	*0x03*
0x9C41	IFSLEW	De/Activate IF0..3 pins slew rate control, default: activated Set to 0x00 to deactivate it	*0x0F*
0x98A2	NFCT_TO	WT value to compute RWT (See [4]) (PN544 timeout will be half as specified)	*0x09*
0x98A3	NFCT_PPt	Define NAD handling on target side Set to 0x30 to disable NAD usage	*0x31*

Address	Name	Comment	Default
0x98A4	NFCT_RTOX	RTOX value to compute RWTINT (See [4]) (PN544 timeout will be half as specified)	*0x07*
0x98DE	NFCI_ATR_TO_MSB	Initiator Activation timeout (in 1ms step) (Recommended Max value is 30 ms).	*0x00*
0x98DF	NFCI_ATR_TO_LSB		*0x1A*
0x9F2C	TGINIT_GUARD_TO_MSB	Target Activation timeout (from 0 to 3.145s in 48µs step)	*0x51*
0x9F2D	TGINIT_GUARD_TO_LSB		*0x61*
0x9F35	Card_Emulation_TO_MSB	Time for which PN544 stays in Card Emulation mode after leaving RF field (from 0 to 3.145s in 48µs step)	*0x04*
0x9F36	Card_Emulation_TO_LSB		*0x11*
0x9F38 *To* *0x9F3D*	MIF_KEYSET0_KEY_A	Key A of Keyset 0 for MIFARE authenticate command	*0*
0x9F3E *To* *0x9F43*	MIF_KEYSET0_KEY_B	Key B of Keyset 0 for MIFARE authenticate command	*0*
0x9F44 *To* *0x9F49*	MIF_KEYSET1_KEY_A	Key A of Keyset 1 for MIFARE authenticate command	*0*
0x9F4A *To* *0x9F4F*	MIF_KEYSET1_KEY_B	Key B of Keyset 1 for MIFARE authenticate command	*0*
0x9F50 *To* *0x9F55*	MIF_KEYSET2_KEY_A	Key A of Keyset 2 for MIFARE authenticate command	*0*
0x9F56 *To* *0x9F5B*	MIF_KEYSET2_KEY_B	Key B of Keyset 2 for MIFARE authenticate command	*0*
0x9F5C *To* *0x9F61*	MIF_KEYSET3_KEY_A	Key A of Keyset 3 for MIFARE authenticate command	*0*

Address	Name	Comment	Default
0x9F62 *To* *0x9F67*	MIF_KEYSET3_KEY_B	Key B of Keyset 3 for MIFARE authenticate command	*0*
0x9F68 *To* *0x9F6D*	MIF_KEYSET4_KEY_A	Key A of Keyset 4 for MIFARE authenticate command	*0*
0x9F6E *To* *0x9F73*	MIF_KEYSET4_KEY_B	Key B of Keyset 4 for MIFARE authenticate command	*0*
0x9F74 *To* *0x9F79*	MIF_KEYSET5_KEY_A	Key A of Keyset 5 for MIFARE authenticate command	*0*
0x9F7A *To* *0x9F7F*	MIF_KEYSET5_KEY_B	Key B of Keyset 5 for MIFARE authenticate command	*0*
0x9F80 *To* *0x9F85*	MIF_KEYSET6_KEY_A	Key A of Keyset 6 for MIFARE authenticate command	*0*
0x9F86 *To* *0x9F8B*	MIF_KEYSET6_KEY_B	Key B of Keyset 6 for MIFARE authenticate command	*0*
0x9F8C *To* *0x9F91*	MIF_KEYSET7_KEY_A	Key A of Keyset 7 for MIFARE authenticate command	*0*
0x9F92 *To* *0x9F97*	MIF_KEYSET7_KEY_B	Key B of Keyset 7 for MIFARE authenticate command	*0*
0x9F98	Retry ReqB Timeout	timeout to send a another reqB on RF error or collision (unit in ms)	*0x01*
0x9F99	max retry of Req B	number of reqB re done	*0x03*
0x9F9A	FeRd Request code	Request code used for Felica poll command in passive 212 phase	*0x01*
0x9F9B	FeRd System code MSB	System code used for Felica poll command in passive 212 phase	*0xFFFF*

Address	Name	Comment	Default
0x9F9C	FeRd System code LSB		
0x9F9E	NfcT-RFOFF-TO_MSB	In NFC target active, Time the target wait after command reception has been ended and RF OFF before reinitializing.	*0x1570*
0x9F9F	NfcT-RFOFF-TO_LSB	To avoid active target in front of passive reader (in 5.5µs step)	
0x9F35	Card_Emulation_TO_MSB	Time for which PN544 stays in CE after leaving RF field. (in 48µs step)	*0x0411*
0x9F36	Card_Emulation_TO_LSB		
0x9EB1	FlashBuildVersion	EEPROM code identifier for FW build revision.	*0x5C*

9.20.4 RF configuration

Table 134. RF settings in EEPROM

Address	Name	Comment	Default
0x9976	ANATXCWGSN	- bits 4 to 7: Set_Conductance_NMOS_LoadNoMod in PCD mode - Others bits are reserved	*0xFF*
0x9977	ANATXCWGSPON	Set_Conductance_PMOS_LoadNoMod in PCD mode	*0xFF*
0x9979	ANATXMODGSN	- bits 4 to 7: Set_Conductance_NMOS_LoadMod in PCD mode - Others bits are reserved	*0xFF*
0x997A	ANATXMODGSPON	Set_Conductance_PMOS_LoadMod in PCD mode	*0x3E*
0x992F	ANT	Switch_ANT1+ANT2_to_GND	*0x12*
0x989E	AntennaLdoCon1	Value used for antenna self test scenario 3: start value for current measurement	*0x3F*
0x989F	AntennaCwGsPOn	Value used for antenna self test CWGSPON register	*0x20*
0x9F0F	PipeLenght	Data length for pipelining	*0x1C*
0x9F10	Pipelining-Mode	Indicates SWP pipelining mode: - Bit 0 : Set to 1 to activate pipelining RF to SWP in reader mode - Bit 1 : Set to 1 to activate pipelining RF to SWP in card mode - Bit 2 : Set to 1 to activate pipelining SWP to RF in reader mode - Bit 3 : Set to 1 to activate pipelining SWP to RF in card mode - Bit 4 : Set to 1 to activate pipelining RF to host in reader mode - Bit 5 : Set to 1 to activate pipelining RF to host in card mode - Bits 6 and 7 : RFU	*0x03*

Note: for details on RF settings, refer to [11].

9.21 PN544 register access

PN544 HW registers can be accessed using *NXP_READ* and *NXP_WRITE* commands.

9.21.1 GPIO settings

For a detailed description please refer to the ‘***GPIO chapter***’.

Table 135. GPIO register set

Address	Name	Comment
0xF820	PINV	Port Inversion register Inverse the polarity
0xF821	PDIR	Port Direction Register If set to 1, the corresponding GPIO is configured in output (whatever PEN value). If set to 0 and if the corresponding PEN bit is set to 1, the corresponding GPIO is configured in input.
0xF822	DPUD	Pull Down Enable register Set to 1, an internal pull down is connected to the corresponding GPIO.
0xF823	UPUD	Pull Up Enable Register Set to 1, an internal pull up is connected to the corresponding GPIO.
0xF82A	PIN	Input Port Register Read the input port value.
0xF82B	POUT	Output Port Register Read or write the output port value.
0xF829	PEN	Signal Input Enable If set to 1, and if the corresponding PDIR is set to 0, the corresponding GPIO is configured in input

Note: Each bit in the configuration register represents the GPIO pin with the corresponding number (i.e. bit0 = GPIO0)

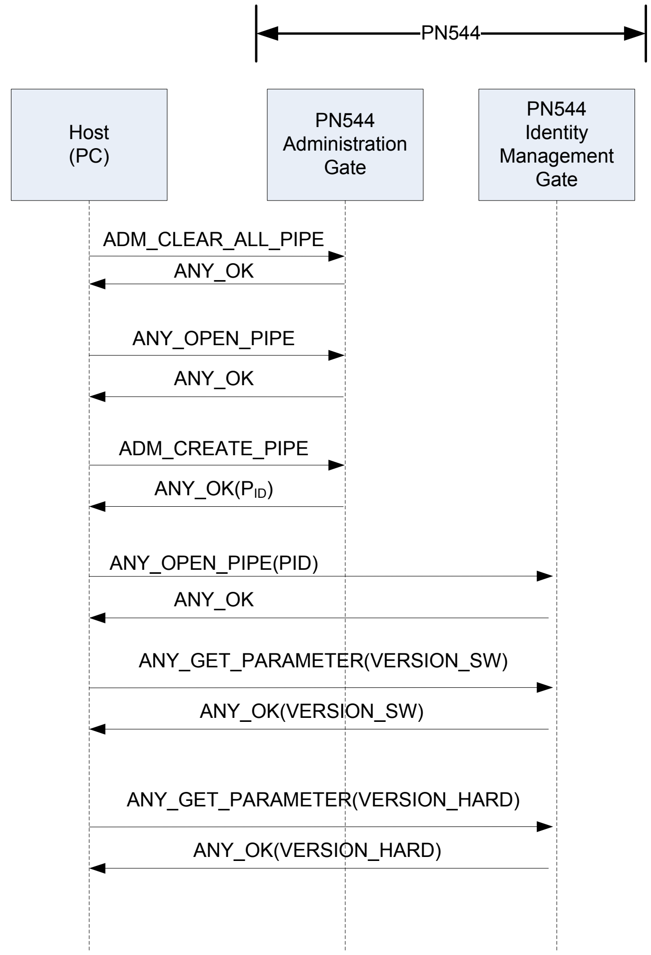
10. Practical Approach on PN544 HCI

For debug of your system we give a first short idea how to verify your host interface connection of PN544 as well as the validation of your chip itself by reading out the version numbers.

10.1 Reading PN544 Version Register

As host we understand here a PC based solution (SCR Tester) with a PN544 Demo Board attached.

Please refer to ETSI HCI specification [1] chapter 8 (HCI procedures) as well for more information.



(3) PN544 identifiers

Fig 75. Read PN544 versions

10.2 PN544 Software & Hardware Version

For information, “VERSION_SW”, “VERSION_HW” and “MODEL_ID”, registry entries defined in the ETSI HCI specification indicates PN544 version as follows:

Table 136. VERSION_SW description

1st byte		2nd byte	3rd byte
7 4	3 0	7 0	7 0
RomLib	Patch	FlashLib Major	FlashLib Minor

E.g. First release will be coded as:

- 10.1.0 (RomLib = 1, no Patch and FlashLib = 1.0)

See [10] for SW Version detail.

Table 137. VERSION_HW description

1st byte		2nd byte		3rd byte
7 5	4 0	7 6	5 0	7 0
Derivate	HW version	#MPW	Software	BSID version

E.g. First releases will be coded as:

- MCB_Rxxx_OESD_T1.dat = ‘00’ ‘00’ ‘00’
- MCB_Rxxx_OESD_T2.dat = ‘00’ ‘40’ ‘00’
- MCB_Rxxx_NESD_T3.dat = ‘00’ ‘80’ ‘00’
- MCB_Rxxx_NESD_T4.dat = ‘00’ ‘C0’ ‘00’

Table 138. MODEL_ID description

NXP IC	ID
PN544 (standalone)	’00’

In addition to Identity management gate, PN544 provides the following additional registry entry NXP_FULL_VERSION_SW:

Table 139. NXP_FULL_VERSION_SW description

Id	Name	Access Rights	Comment	Length	Default
‘10’	*NXP_FULL_VERSION_SW*	RO	Version of the software as followed: - RomLibVersion - RomLibBuildVersion - eedata_SM_CustomerProjectVersion - eedata_SM_RomBuildVersion - PatchCode_CustomerProjectVersion - PatchCode_Version - FlashLibVersionMajor - FlashLibVersionMinor - FlashLibBuildVersion - eedata_UM_CustomerProjectVersion - eedata_UM_RomBuildVersion	*11*	*NA*

10.3 PN544 Versioning and Configuration Management

Every patch or firmware update provided by NXP will automatically change the above registers of PN544 to guarantee full traceability and support.

11. Legal information

11.1 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.

11.2 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.

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 accepts 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 national authorities.

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.

11.3 Licenses

ICs with DPA Countermeasures functionality

LICENSED DPA COUNTERMEASURES™

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

11.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

11.5 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.

12. List of figures

13. List of tables

14. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2010-06-16
Document identifier: 191812

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2010-06-16
Document identifier: 191812

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 2010-06-16
Document identifier: 191812