

AN<10449_1>

PN532 application note

Rev. <01.00> — 07 December 2006

Application note

Document information

Info	Content
Keywords	NFC, ISO/IEC 18092, the PN532A3HN/C104
Abstract	The PN532 Application note. How to use the PN532 as a reader, as a card, or in a NFC peer-to-peer communication.

Revision history

Rev	Date	Description
1.0	2006 Dec 14	First release
<0.1>	<2006-09-25>	creation

Contact information

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

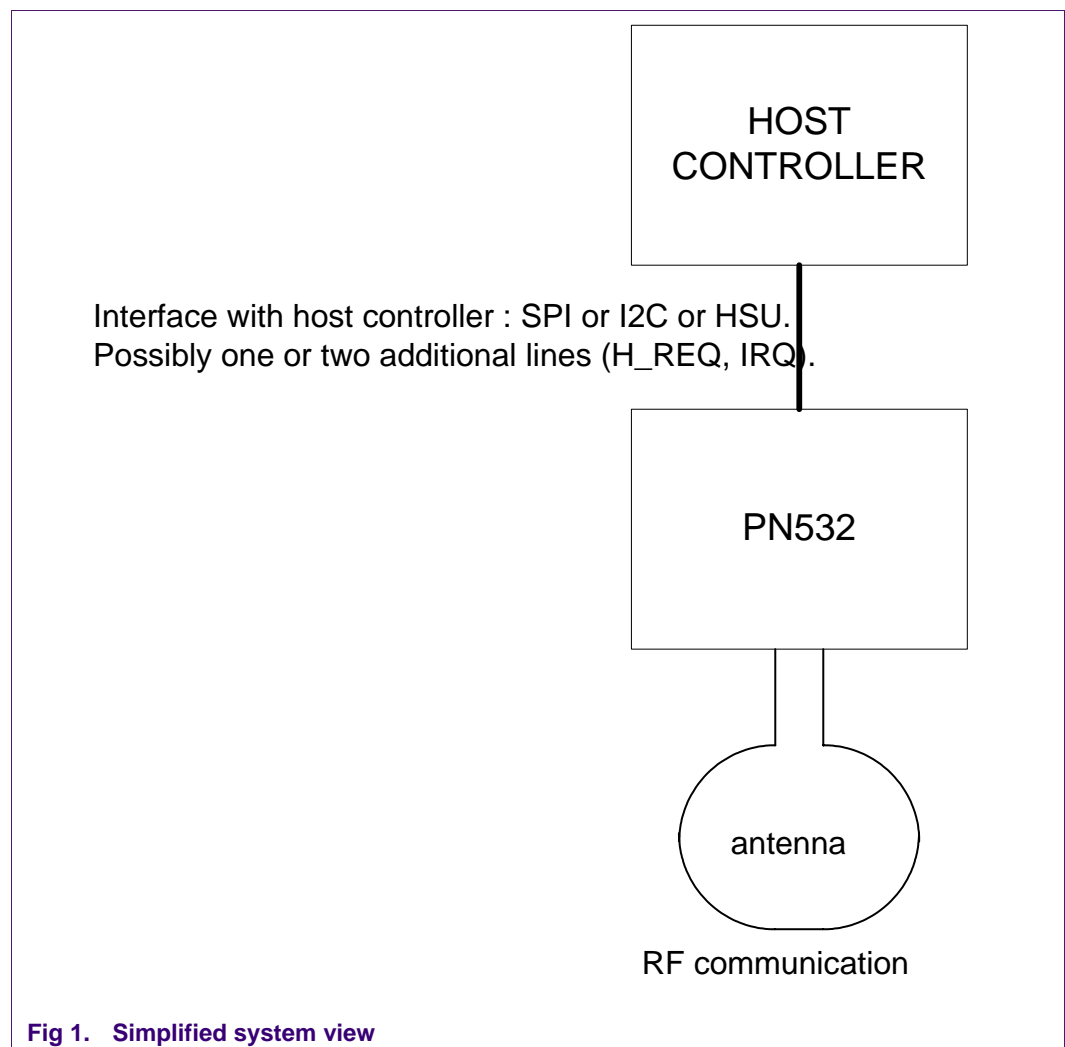
For sales office addresses, please send an email to: sales.addresses@www.nxp.com

1. Introduction

The PN532 is a highly integrated transmission module for contactless communication at 13.56 MHz including microcontroller functionality based on a 80C51 core with 40 Kbytes of ROM and 1 Kbyte of RAM.

The PN532 combines a modulation and demodulation concept completely integrated for different kinds of contactless communication methods and protocols at 13.56 MHz (particularly Near Field Communication NFC), with an easy-to-use firmware for the different supported modes and the required host interfaces.

The PN532 includes a switch to power an external SAM connected to S2C interface. It is controlled by the embedded firmware.



This document intends to allow the customer getting quickly started with the PN532. It summarizes commands needed to use the PN532 as a reader, as a card, or in a NFC peer-to-peer communication. It gives an overview on possible interfaces with the host controller.

Detailed description of the PN532 firmware can be found in the PN532 User manual (cf. References table below).

Full description of the PN532 IC can be found in the PN532 Datasheet.

References

<i>Ref.number</i>	<i>Document name</i>	
1	PN532 user manual UM0502-06	
2	PN532 Product Datasheet	115430.pdf
3	NFC Transmission Module Antenna and RF Design Guide	100720.pdf
4	DESfire cards specification	M075031.pdf
5	Mifare cards specification	http://www.nxp.com/products/identification/datasheets/
6	ISO/IEC 14443 specification (T=CL)	ISO/IEC 14443-3 specification ISO/IEC 14443-4 specification
7	NFCIP-1 specification	ISO/IEC 18092 or ECMA340 specification

Glossary

NFC	Near Field Communication
HSU	High Speed UART
SMX	Philips SmartMX (Memory Extension)
PCR	Power, Clock and Reset controller
SAM	Secure Access Module
MINT	Multiple Interfaces
PMU	Power Management Unit
DEP	Data Exchange Protocol. (see reference 7)

The PN532 is the successor of the PN531. Main new functionalities are:

- ISO/IEC 14443 type B reader
- ISO/IEC 14443 card emulation
- I2C wake up
- AutoPoll function
- Up to 265 bytes sent in one frame from the Host controller to the PN532
- Handshake mode implemented for SPI, I2C and HSU
- Power management improvement
- Integrated LDO (2.7V to 5.4V power supply)
- Antenna self test
-

Pin correspondence: In the documents the following correspondence can be used in the names of the pins:

PN532 Datasheet name	Application Note name
P70_IRQ	IRQ
P32_INT0	H_REQ
P50_SCL	SCL
HSU_RX	T_RX
HSU_TX	T_TX

IC correspondence: In the documents the following correspondence can be used in the names of the IC:

Commercial Name	Application Note name
PN5320A3HN/C104 and PN5321A3HN/C104	PN532
PN5320A3HN/C104	PN5320
P5CN072	SMX

2. Interfaces with the host controller

2.1 PN532 block diagram

The PN532 is based on an 8051 core, with 40 Kbytes of ROM and 1Kbyte of RAM. The chip contains a contactless UART, a contactless front end, a “PCR” block that controls clocks and power.

It can be connected to the host controller in I2C, SPI or HSU (High Speed UART). One or two more lines (IRQ and H_REQ) can be added. The interface is selectable using I0 and I1 pins.

A SAM companion chip can be attached using S2C interface.

A part of the IC can be powered directly from a mobile battery (VBAT between 2.7V and 5.4V). The Pad power supply (PVDD) must be between 1.6V and 3.6V.

The SAM power supply SVDD is provided by the PN532.

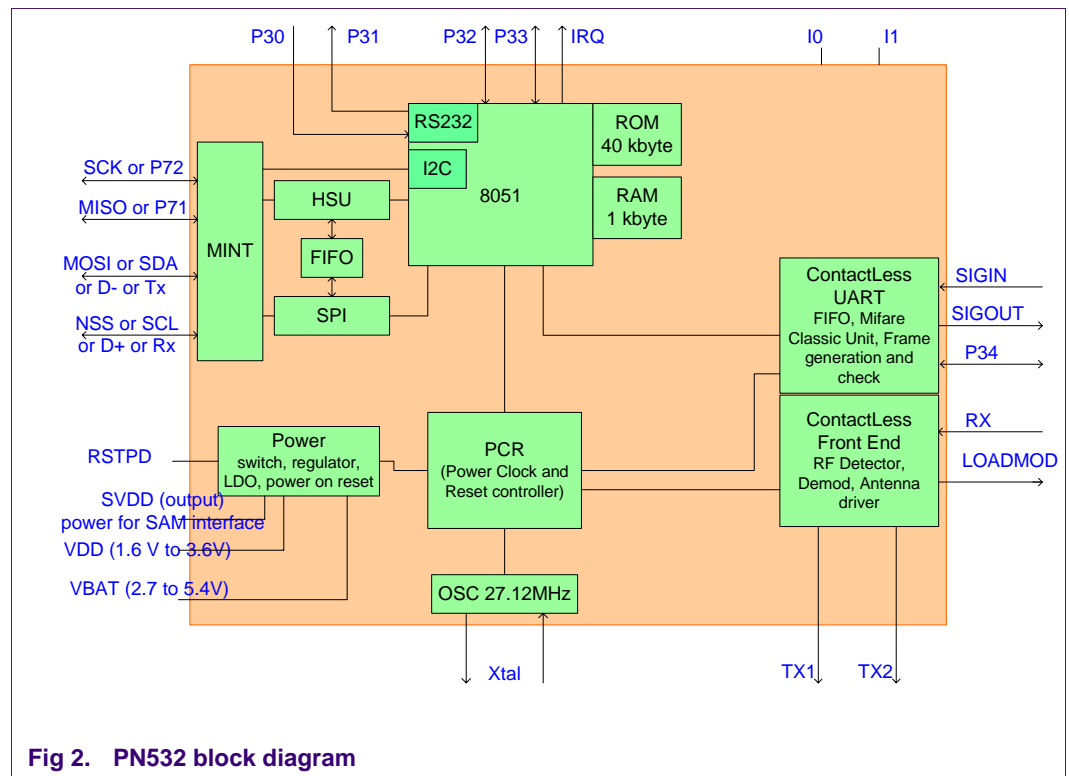


Fig 2. PN532 block diagram

Three interfaces are available: I2C, SPI and HSU (high speed UART). The interface is selectable by hardware (pin I0 and I1).

	Interface Selection Pin	
	I0 (pin #16)	I1 (pin #17)
HSU	0 Ground	0 Ground
I2C	1 DVDD	0 Ground
SPI	0 Ground	1 DVDD

The embedded software manages the communication with the host controller (I2C, SPI, or HSU interface, protocol on the host link) and the communication on the RF side.

2.2 PN532 host link protocol

The protocol used on host link is fully described in the PN532 User manual (cf. References table on page 4)

2.2.1 Standard frame

A basic exchange consists in a command frame sent by the host controller to the PN532, an ACK frame sent by the PN532 as soon as the command is correctly received, and a response frame, read by the host controller (polling mechanism or use of IRQ).

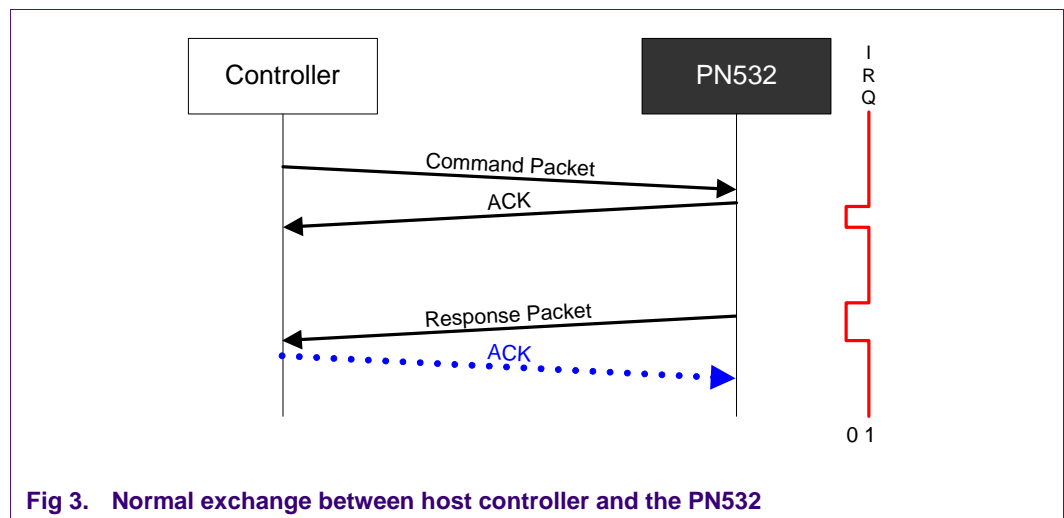


Fig 3. Normal exchange between host controller and the PN532

Command and response frame structure is shown in figure 3.

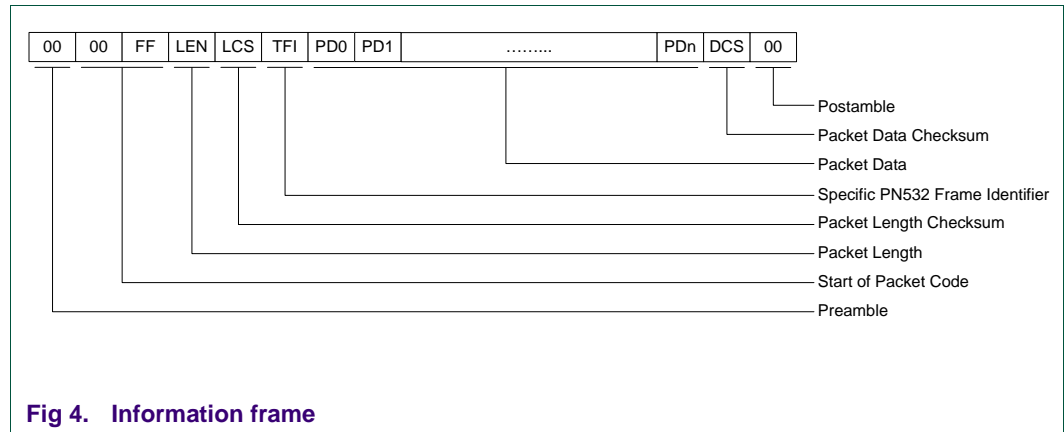


Fig 4. Information frame

- **PREAMBLE** 1 byte
- **START CODE** 2 bytes (00h and FFh)
- **LEN** 1 byte indicating the number of bytes in the data field (TFI and PD0 to PDn)
- **LCS** 1 Packet Length Checksum LCS byte that satisfies the relation:
Lower byte of [LEN + LCS] = 00h
- **TFI** 1 byte the PN532 Frame Identifier, the value of this byte depends on the way of the message
 - D4h in case of a frame from the system controller to the the PN532
 - D5h in case of a frame from the the PN532 to the system controller
- **DATA** LEN-1 bytes of Packet Data Information
The first byte PD0 is the Command Code
- **DCS** 1 Data Checksum DCS byte that satisfies the relation:
Lower byte of [TFI + PD0 + PD1 + ... + PDn + DCS] = 00h
- **POSTAMBLE** 1 byte

ACK frame is described in figure 4.

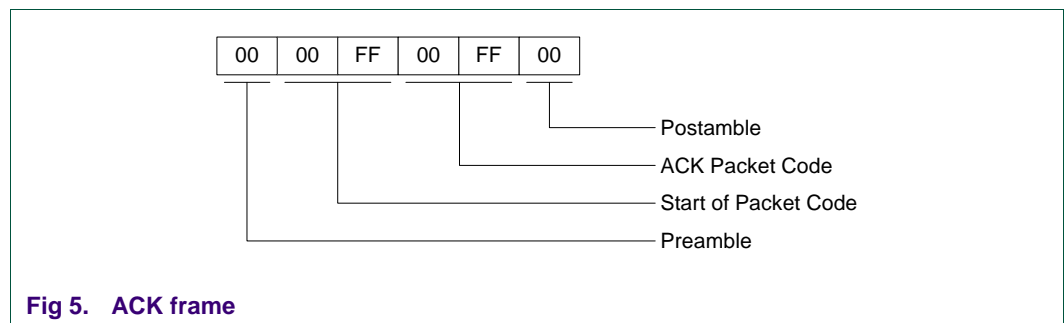


Fig 5. ACK frame

List of available commands (PD0 byte) is provided in paragraph 3.2 on page 25.

2.2.2 Extended frame

The information frame has an extended definition allowing exchanging more data between the host controller and the the PN532 (theoretically up to 64kB). In the firmware implementation of the the PN532, the maximum length of the packet data is limited to 264 bytes (265 bytes with TFI included).

The structure of this frame is the following:

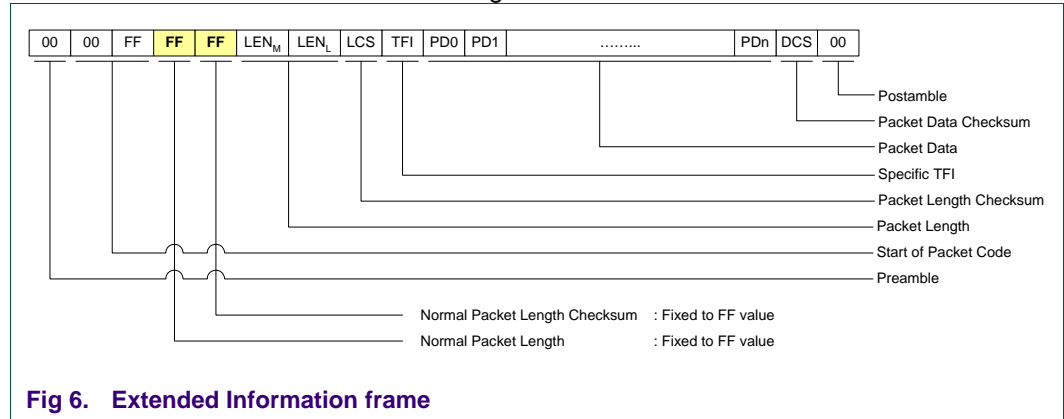


Fig 6. Extended Information frame

The normal **LEN** and **LCS** fields are fixed to the **0xFF** value, which is normally considered as an erroneous frame, due to the fact that the checksum does not fit.

The real length is then coded in the two following bytes **LEN_M** (MSByte) and **LEN_L** (LSByte) with:

LENGTH = **LEN_M** x 256 + **LEN_L** coding the number of bytes in the data field (TFI and PD0 to PDn)

- **LCS** 1 Packet Length Checksum LCS byte that satisfies the relation:
Lower byte of [**LEN_M** + **LEN_L** + **LCS**] = 0x00,
- **DATA** **LENGTH**-1 bytes of Packet Data Information
The first byte PD0 is the Command Code.

The host controller, for sending frame whose length is less than 255 bytes, can also use this type of frame.

But, the the PN532 always uses the suitable type of frame, depending on the length (Normal Information Frame for frame <= 255 bytes and Extended Information Frame for frame > 255 bytes).

2.2.3 Additional lines (IRQ only or IRQ and H_REQ)

Once the interface with the host controller is chosen, the choice of using one or two additional lines (H_REQ and IRQ) has also to be considered.

IRQ indicates to the host controller to know when a response from the PN532 is ready. To enable IRQ, command SetParameters has to be used

Handshake mode means the choice of using one or two additional lines (IRQ, with a behavior different from the previous paragraph, and optionally H_REQ). To select Handshake mode, a hardware configuration is required:

Handshake mode is particularly useful for managing the PN532 power down and wake up, and virtual card mode.

The implementation of IRQ behavior and Handshake mode is fully described in the user manual (for each interface – I2C – SPI or HSU)

P31 is used to select handshake or standard mode:

Handshake	Selection Pin P31 (pin #26)
Handshake is selected	0 Ground
Normal mode	1 PVDD

This mode allows:

- the controller to wake up the PN532 before sending a command
- The PN532 to warn the host controller when a transaction happened in virtual card mode.

Remarks:

- **IRQ pin behaviour is different between handshake mode and standard mode using IRQ.**
- **in this mode, it is not possible to use SetParameters to enable IRQ**

In the following pages of this document, mandatory bytes of the command frames (preamble, start code, length, LCS, TFI, DCS and postamble) are not represented. The focus is put on command code (PD0), command parameters and data bytes.

2.3 Typical application diagram

See also application schematics of NXP demoboard, provided in section 4 on page 64.

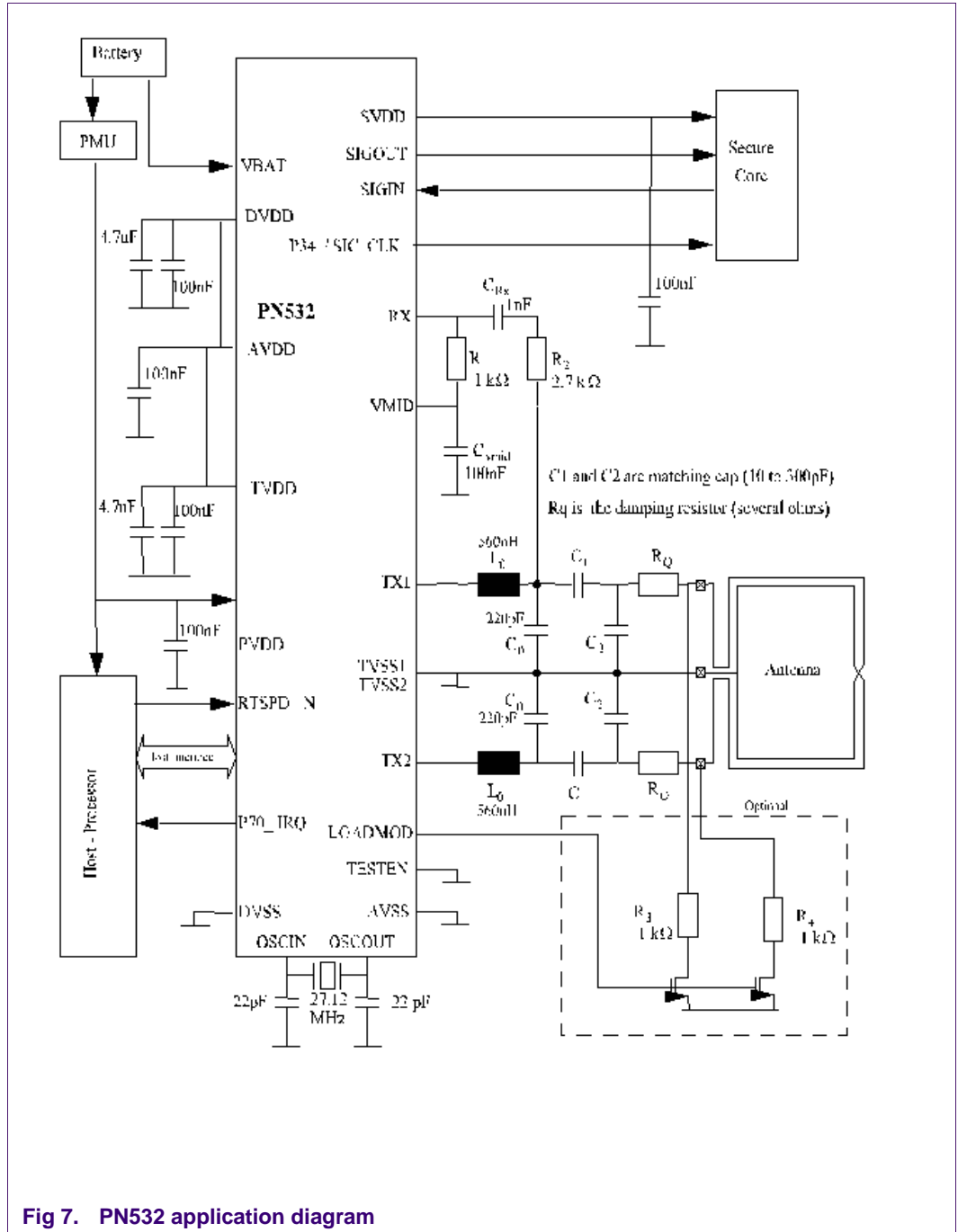


Fig 7. PN532 application diagram

Default pin configuration

The default pin configuration is not changed by the PN532 v1.4 firmware. Consequently, the default pin configuration is as described in the PN532 datasheet.

Pin	Configuration	Additional information
I0 I1	Input	Connect directly to DVDD or to GND (no need of external resistor)
PVDD	Power pin	Externally supplied regulated voltage, 1.6V to 3.6V
RSTPD_N	Input	Connect directly to PVDD (no need of external resistor) RSTPD_N must never be above PVDD
P30 / UART_RX P31 / UART_TX P32_INT0 P33_INT1 P34 / SIC_CLK P35	Quasi bi directional	No need of external resistor. When connected to the P5CN072, to be used in Virtual Card mode, P34 / SIC_CLK shall be connected to IO2 (pin 26)
P70_IRQ	Quasi bi directional	No need of external resistor. In the Application Note P70_IRQ will be written as IRQ when used as interrupt line.
MISO / P71	In I2C mode: Quasi bi directional	No need of external resistor.
	In SPI: Push pull	No need of external resistor
	In HSU: Quasi bi directional	No need of external resistor
SCK / P72	In I2C mode: Quasi bi directional	No need of external resistor.
	In SPI: Input	No need of external resistor.
	In HSU: Quasi bi directional	No need of external resistor
MOSI / SDA / HSU_TX	In I2C mode: Open drain	Use pull up
	In SPI: Input	No need of external resistor
	In HSU: Push pull	No need of external resistor
NSS / P50_SCL / HSU_RX	In I2C mode: Open drain	Use pull up
	In SPI: Input	No need of external resistor
	In HSU: Input	No need of external resistor

2.4 How to interface with the PN532?

2.4.1 SPI interface

The PN532 is slave. Functional operation is guaranteed up to 1.2 MHz. A Status byte (Bit 0 of Status byte) indicates if the PN532 is ready to give a response or not. First byte sent on MOSI by the host controller indicates which operation will be performed:

xxxx xx10 : Read (by the host) Status byte

xxxx xx01 : Write data (transmission from the host to the PN532)

xxxx xx11 : Read data (transmission from the PN532 to the host)

After having sent a command, the host controller must wait for bit 0 of Status byte equals 1 before reading the data from the PN532.

Bytes are transmitted LSB first.

2.4.1.1 Sequence to use communicate in SPI

```
// Send a command
// =====
write "write data" i.e. xxxx xx01 LSB first
write a certain number of data bytes

// Wait for ACK of the PN532
// =====
write "read status" ie xxxx xx10
read one byte. Response of the PN532 is ready only if this byte is 01h.

write "read data" ie xxxx xx11
read 6 bytes of ACK frame (00 00 FF 00 FF 00)

// Wait for response of the PN532
// =====
write "read status" ie xxxx xx10
read one byte. Response of the PN532 is ready only if this byte is 01h.

write "read data" ie xxxx xx11
read a certain number of bytes (length at offset 4 in the frame)
```

SPI bus can be shared between the PN532 and another device.

2.4.1.2 SPI mode selection

In the PN532, CKPOL and CKPHA are configurable with P3.0 and P3.3

SPI mode		P30	P33_INT1
CKPHA = 1, CKPOL = 1 Data is sampled on the second edge of SCK SCK is active low	3	0	0
CKPHA = 0, CKPOL = 1 Data is sampled on the first edge of SCK SCK is active low	2	1	0
CKPHA = 1, CKPOL = 0 Data is sampled on the second edge of SCK SCK is active high	1	0	1
CKPHA = 0, CKPOL = 0 Data is sampled on the first edge of SCK SCK is active high	0	1	1

2.4.2 I2C interface

I2C clock speed is up to 400 kHz.

The PN532 is a slave. Its address is 48h.

In case the PN532 doesn't acknowledge its address, the host controller shall retry (it shall retransmit the slave address).

First byte read by the host controller indicates if the PN532 response is available ("Ready byte"). The host controller must wait for first byte is 01h before reading the data.

The host controller must send a Stop bit after reading Ready byte if Ready = 00h. If Ready = 01h, the host must go on reading data. If ACK expected, the host must read 6 bytes since ACK frame length is 6. If response to command expected, the host must decode the length byte (byte 4 of incoming data) to know how many bytes must be read before generating the Stop bit.

2.4.2.1 Handshake mode in I2C

See reference 1 for a detailed description.

Four lines are connected between the host controller and the PN532: SCL, SDA, H_REQ and IRQ.

However, the H_REQ line is optional. It is possible not to use it. Indeed, the PN532 can be waken up either by a pulse on H_REQ or by its own slave address (0x48)

Remark: in handshake mode, IRQ line is not set low by the PN532 when ACK frame is ready (different behaviour than standard mode and use of IRQ).

The PN532 is not in power down, the host controller initiates an exchange.

H_REQ can be used but is not mandatory.

- **H_REQ used**

H_REQ usage is described in reference 1.

- **H_REQ not used**

The figure below shows an exchange without H_REQ. IRQ line is asserted by the PN532 when the response to the command is ready.

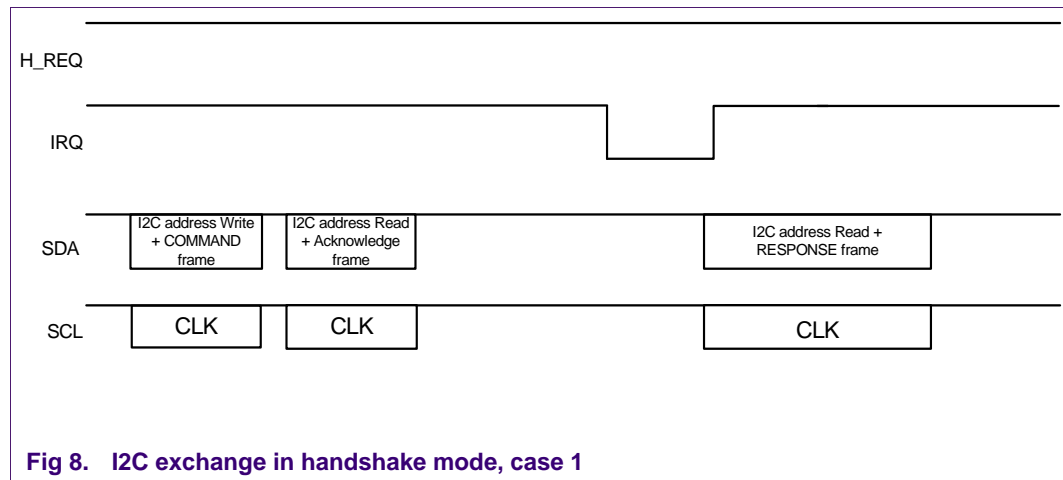


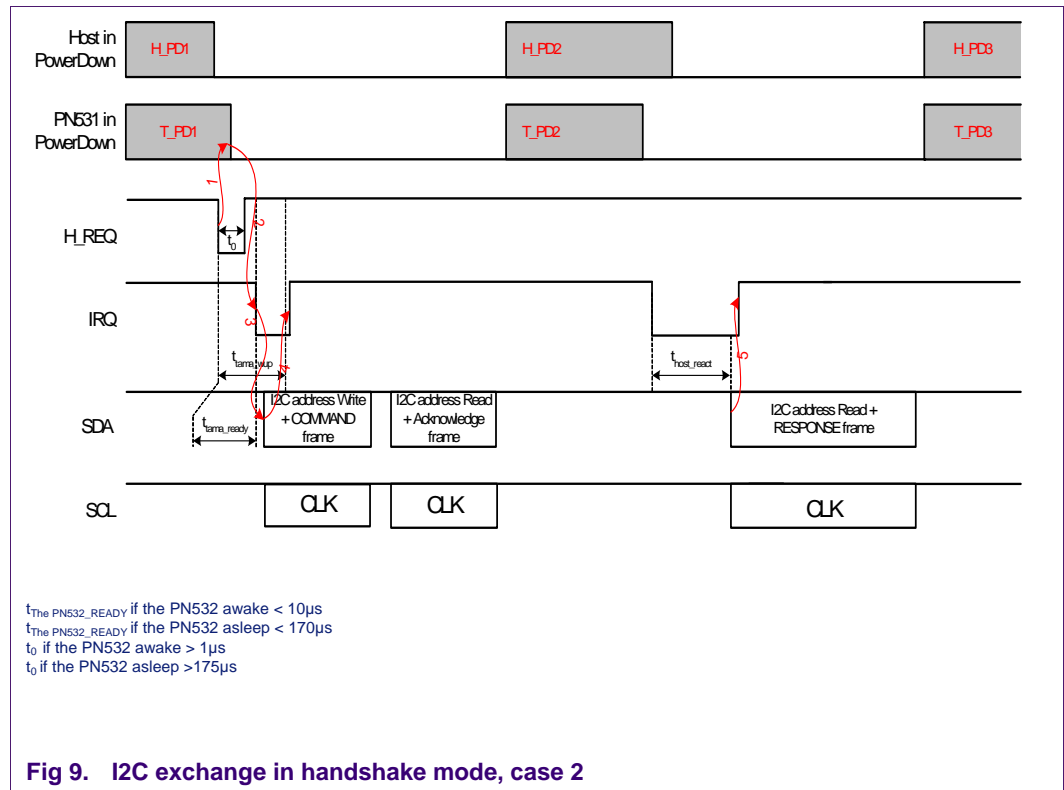
Fig 8. I2C exchange in handshake mode, case 1

The PN532 may be in power down, the host controller initiates an exchange.

H_REQ can be used or not.

- H_REQ used

If the PN532 is in power down, the host controller shall generate a pulse on H_REQ to wake it up, and wait for IRQ negative edge before sending a command. the PN532 will also generate set IRQ low to indicate that the response is ready.



• **H_REQ not used**

The H_REQ line is optional. It is possible not to use it. Indeed, the PN532 can be waken up either by a pulse on H_REQ or by its own slave address (0x48)

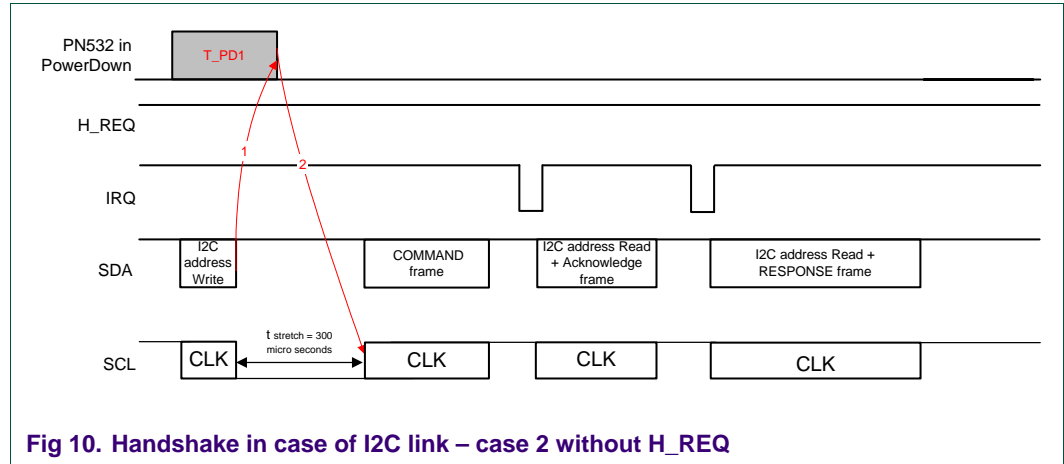


Fig 10. Handshake in case of I2C link – case 2 without H_REQ

The PN532 is in power down, after TgInitAsTarget command.

After TgInitAsTarget command, the PN532 automatically goes into power down (if there is no external field). It saves power until an initiator is detected. The initiator wakes up the PN532 and the PN532 is configured as a target. the PN532 will assert IRQ line to indicate to the host controller that response to TgInitAsTarget is ready.

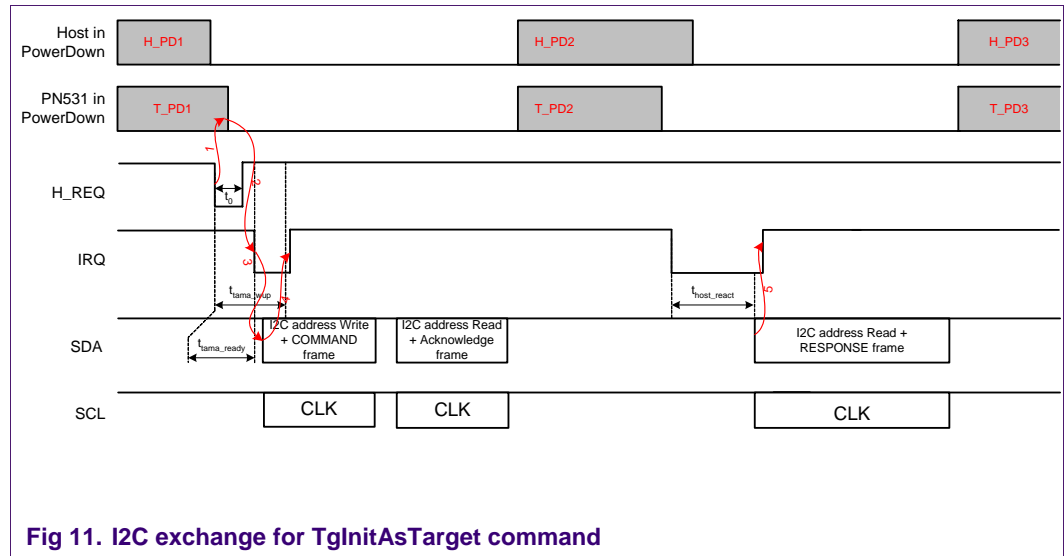


Fig 11. I2C exchange for TgInitAsTarget command

Virtual card mode.

The PN532 informs the host controller that something happened.

The host controller is possibly in power down. the PN532 is woken up by an external event, for example RF detection in virtual card mode. the PN532 is automatically switched to standard mode. IRQ is asserted by the PN532 to wake up the host controller. The host controller shall send a GetGeneralStatus command to get information about what happened (see GetGeneralStatus command description in User manual (cf. References table on page 4)).

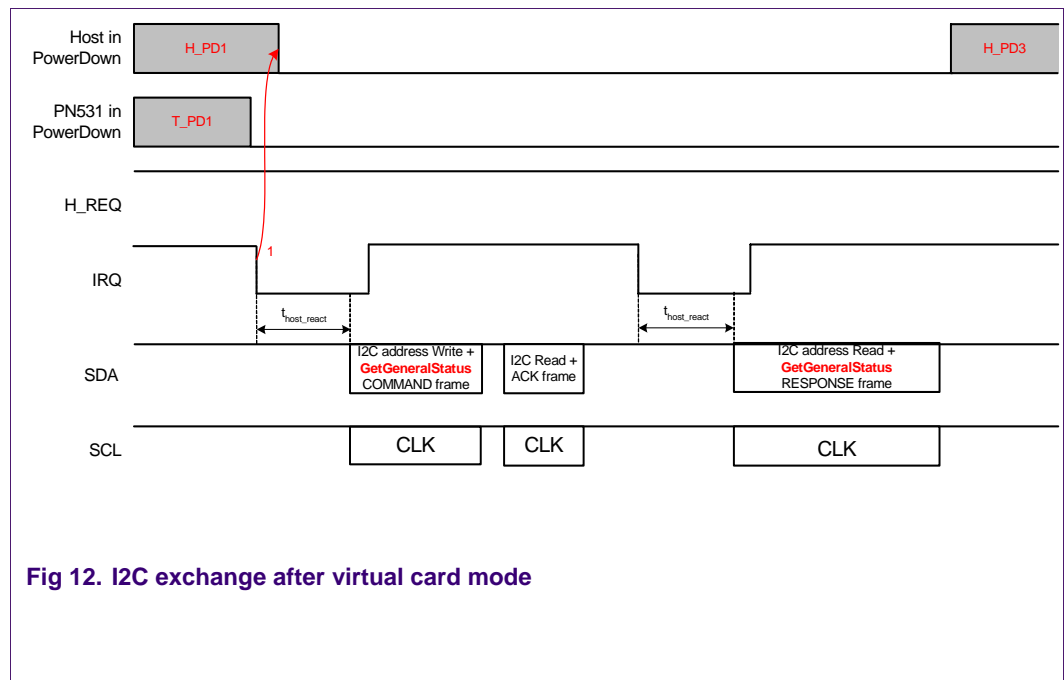


Fig 12. I2C exchange after virtual card mode

2.4.3 Recommended operating conditions

It is recommended to use the handshake mode, described in paragraph 2.4.2.1, in order to reduce the traffic on the I2C interface, and to improve the overall data throughput.

If the handshake mode is not used, the host controller must poll for the response. It is recommended to use a polling frequency lower than 100 Hz.

2.4.4 HSU interface

2.4.4.1 Baudrate

Default baudrate is 115200 bauds.

The PN532 HSU is up to 1288 kbauds. The HSU speed is changeable with SetSerialBaudrate command (command code 10h, parameter = baudrate).

00h	01h	02h	03h	04h	05h	06h	07h	08h
9.6 kbauds	19.2 kbauds	38.4 kbauds	57.6 kbauds	115.2 kbauds	230.4 kbauds	460.8 kbauds	921.6 kbauds	1288 kbauds

The host controller must send an ACK frame after reception of SetSerialBaudrate response. the PN532 will switch to the new baudrate only after reception of this ACK frame.

How to change HSU speed in SCRTester?

To activate sending of ACK frame after reception of a command, use “.config(10,01)” in SCRTester.

Example of script file:

```
.config(10,01);           // SCRTester is forced to send the optional ACK
10 06;                   // SetSerialBaudrate (460.8 kbauds)
.config(00, 460800);     // Change baudrate of SCRTester to 460.8 kbaud
.config(10, 00);        // Deactivate the optional ACK option
```

2.4.4.2 Handshake mode in HSU

Four lines are connected between the host controller and the PN532: HSU_RX (T_RX) in the figures), HSU_TX (T_TX in the figures), H_REQ and IRQ.

Remark: in handshake mode, IRQ line is **not** set low by the PN532 when ACK frame is ready (different behaviour than standard mode + use of IRQ).

The PN532 is not in power down, the host controller initiates an exchange.

No handshake mechanism is needed.

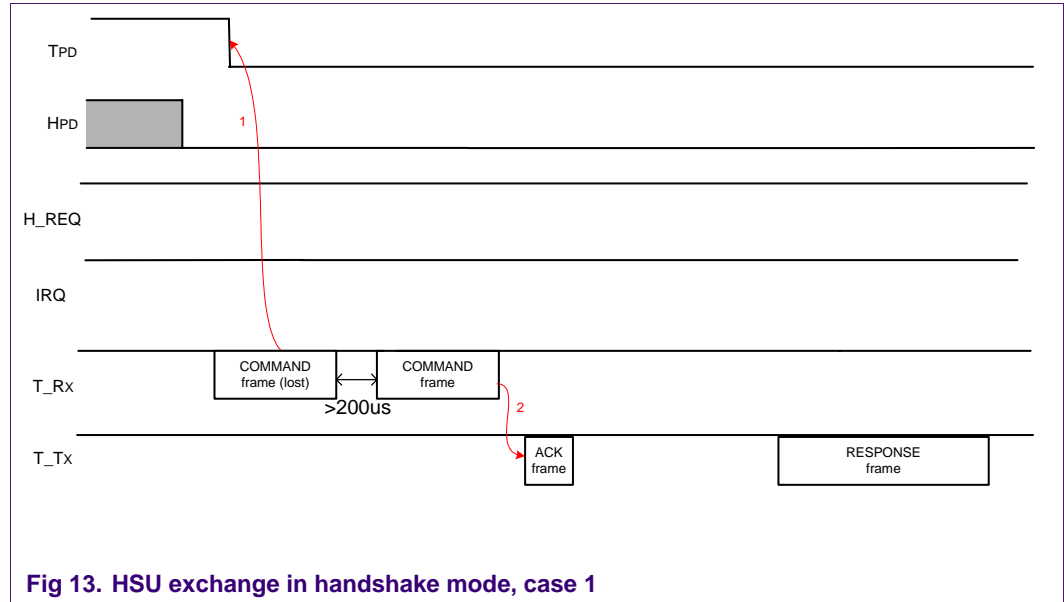


Fig 13. HSU exchange in handshake mode, case 1

The PN532 may be in power down, the host controller initiates an exchange.

The use of H_REQ is optional. If the PN532 is in power down, the host controller shall either generate a pulse on H_REQ or send a dummy command, to wake it up; and wait at least 200 μs before sending a command.

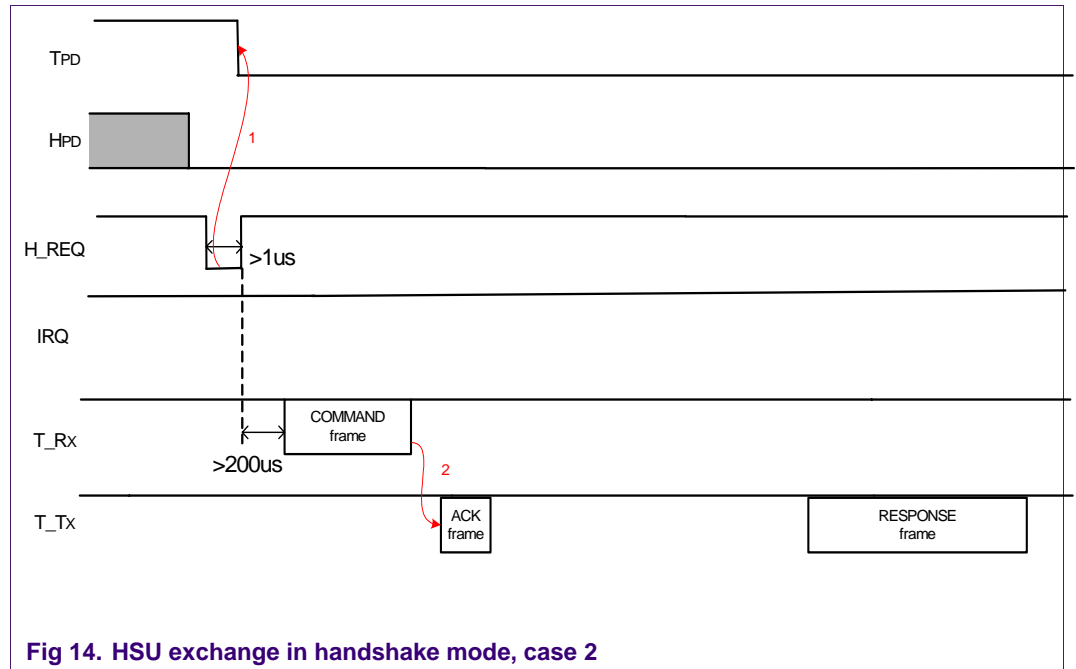


Fig 14. HSU exchange in handshake mode, case 2

The PN532 is in power down, after TglnitAsTarget command.

After TglnitAsTarget command, the PN532 automatically goes into power down (if there is no external field). It saves power until an initiator is detected. The initiator wakes up the PN532 and the PN532 is configured as a target. the PN532 will assert IRQ line to indicate to the host controller that response to TglnitAsTarget is ready. The host acknowledges with a pulse on H_REQ line. If H_REQ is not used, (no H_REQ falling edge is detected during 10 ms), the the PN532 will send its response frame, supposing that the host controller has been awoken. See reference 1.

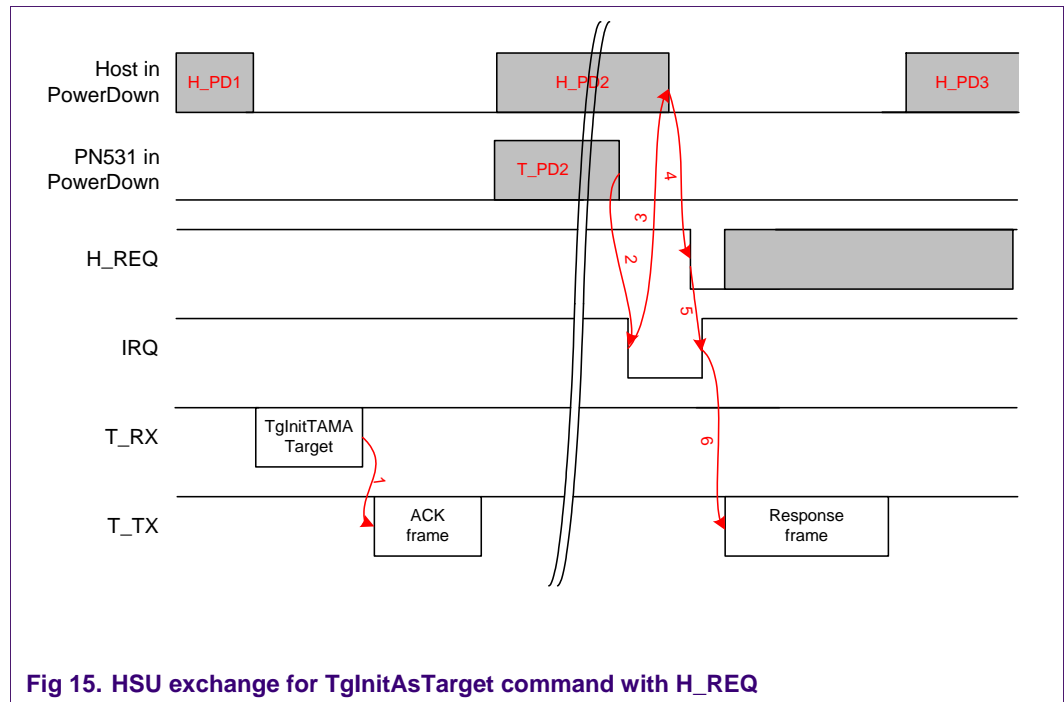


Fig 15. HSU exchange for TglnitAsTarget command with H_REQ

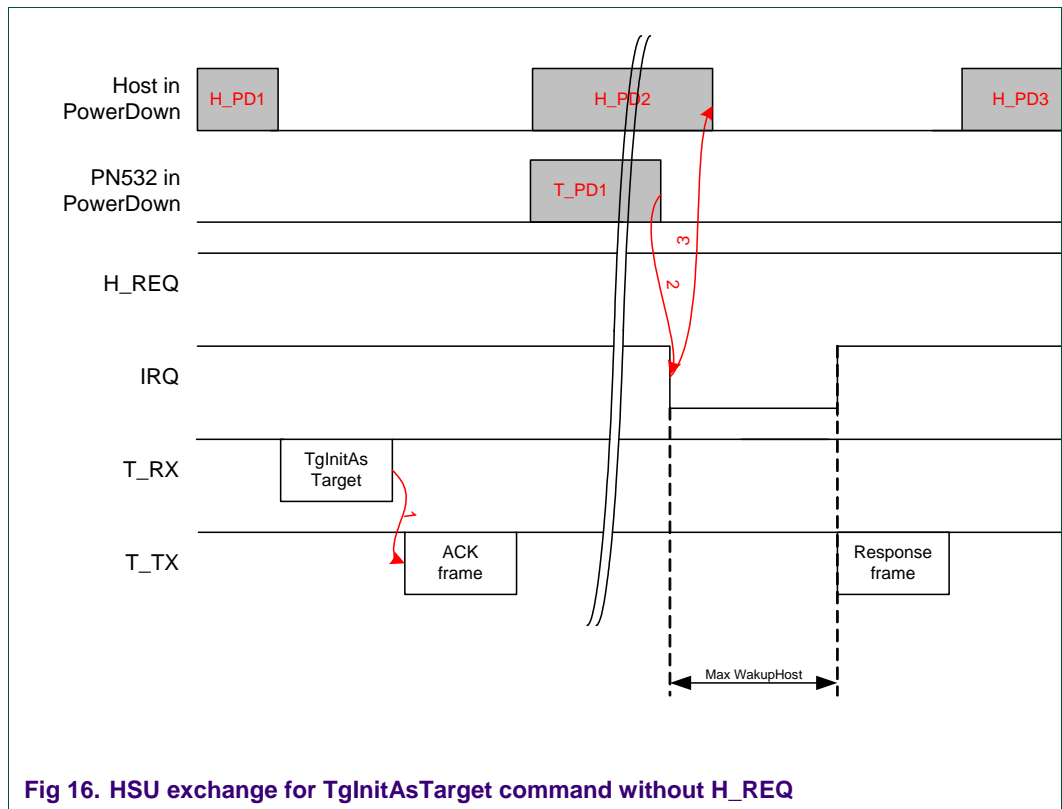


Fig 16. HSU exchange for TgInitAsTarget command without H_REQ

Virtual card mode.

The PN532 informs the host controller that something happened.

The host controller is possibly in power down. the PN532 is woken up by an external event, for example RF detection in virtual card mode. the PN532 is automatically switched to standard mode. IRQ is asserted by the PN532 to wake up the host controller (1). The host controller optionally acknowledges with a falling edge on H_REQ (2), and shall send a GetGeneralStatus command to get information about what happened and to release IRQ line (3 and 4)(see GetGeneralStatus command description in reference 1).

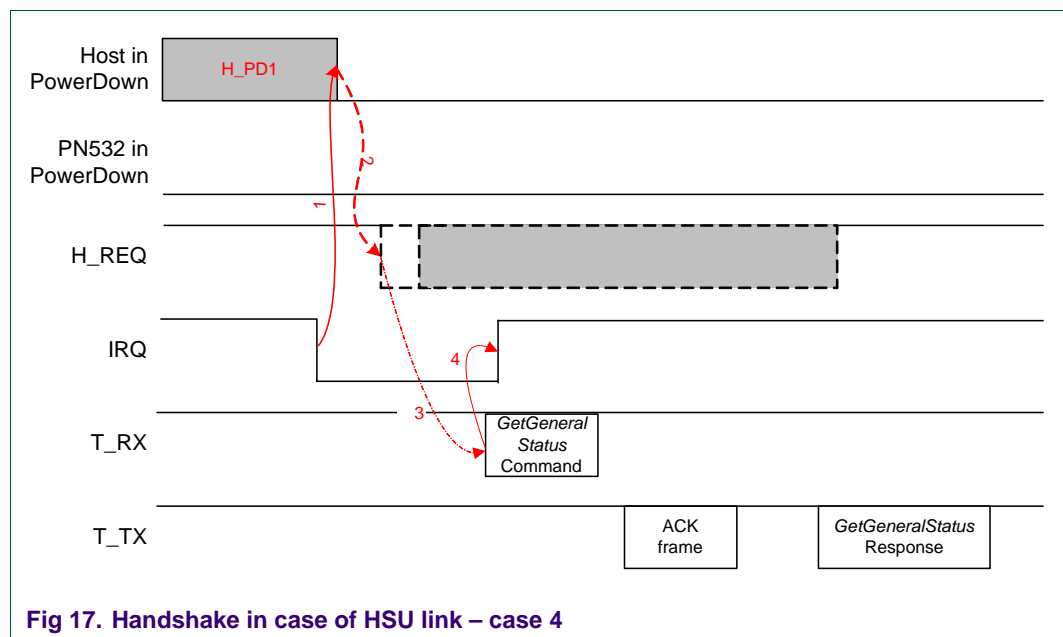


Fig 17. Handshake in case of HSU link – case 4

3. The PN532 commands

3.1 Global view

The PN532 can be access using directly the firmware API described in reference 1 and in the following pages (interface B in the figure 17). Or an upper software layer can be used (NXP can provide this layer called Hardware Abstraction Layer (HAL) – HAL is the interface A in the figure 17).

Note: PN51x, PNxxx, RCxxx represents other NXP NFC products. PN53x represents PN531 or the PN532 product.

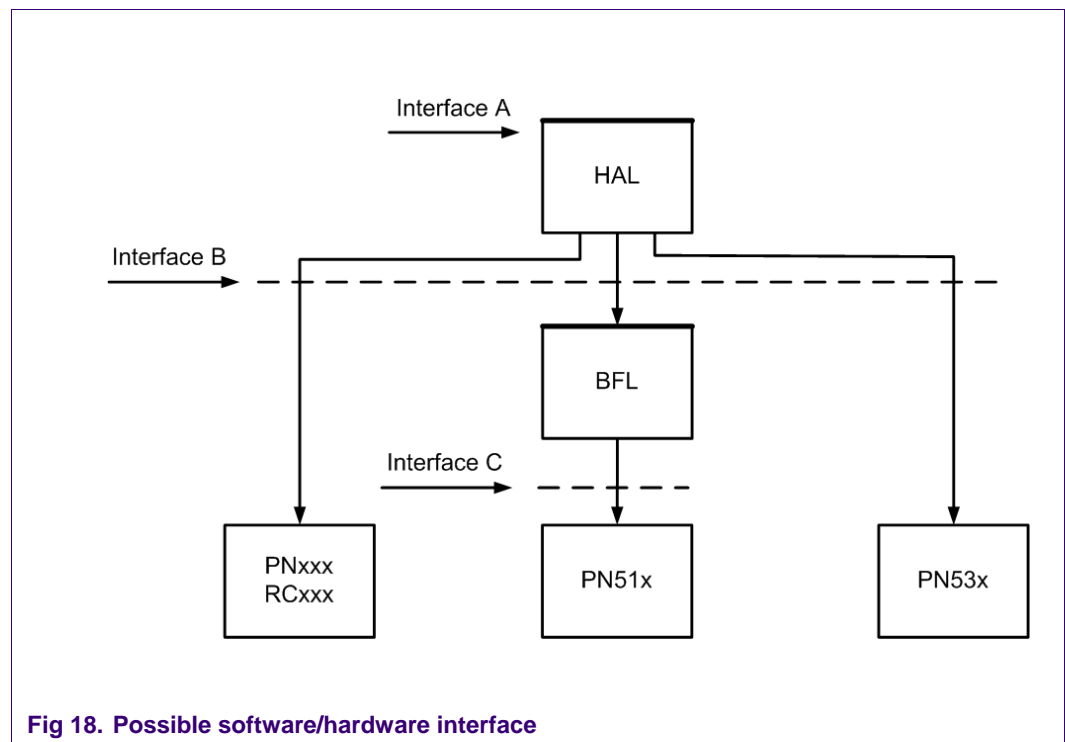


Fig 18. Possible software/hardware interface

The next paragraph described the “interface B”, i.e. the firmware commands.

3.2 The PN532 embedded software API: commands and errors lists

3.2.1 Commands codes

Command	The PN532 as Initiator	The PN532 as Target	Command Code
M i s c e l l a n e o u s			
Diagnose	X	X	0x00
GetFirmwareVersion	X	X	0x02
GetGeneralStatus	X	X	0x04
ReadRegister	X	X	0x06
WriteRegister	X	X	0x08
ReadGPIO	X	X	0x0C
WriteGPIO	X	X	0x0E
SetSerialBaudRate	X	X	0x10
SetParameters	X	X	0x12
SAMConfiguration	X	X	0x14
PowerDown	X	X	0x16
R F c o m m u n i c a t i o n			
RFConfiguration	X	X	0x32
RFRegulationTest	X	X	0x58
I n i t i a t o r			
InJumpForDEP	X		0x56
InJumpForPSL	X		0x46
InListPassiveTarget	X		0x4A
InATR	X		0x50
InPSL	X		0x4E
InDataExchange	X		0x40
InCommunicateThru	X		0x42
InDeselect	X		0x44
InRelease	X		0x52
InSelect	X		0x54

Command	The PN532 as Initiator	The PN532 as Target	Command Code
InAutoPoll	X		0x60
T a r g e t			
TgInitAsTarget		X	0x8C
TgSetGeneralBytes		X	0x92
TgGetData		X	0x86
TgSetData		X	0x8E
TgSetMetaData		X	0x94
TgGetInitiatorCommand		X	0x88
TgResponseToInitiator		X	0x90
TgGetTargetStatus		X	0x8A

3.2.2 Errors codes

Error cause	Error code
Time Out, the target has not answered	0x01
A CRC error has been detected by the contactless UART	0x02
A Parity error has been detected by the contactless UART	0x03
During a MIFARE anticollision/select operation, an erroneous Bit Count has been detected	0x04
Framing error during MIFARE operation	0x05
An abnormal bit-collision has been detected during bit wise anticollision at 106 kbps	0x06
Communication buffer size insufficient	0x07
RF Buffer overflow has been detected by the contactless UART (bit BufferOvfl of the register <i>CL_ERROR</i>)	0x09
In active communication mode, the RF field has not been switched on in time by the counterpart (as defined in NFCIP-1 standard)	0x0A
RF Protocol error (cf. reference [4], description of the <i>CL_ERROR</i> register)	0x0B
Temperature error: the internal temperature sensor has detected overheating, and therefore has automatically switched off the antenna drivers	0x0D

Error cause	Error code
Internal buffer overflow	0x0E
Invalid parameter (range, format, ...)	0x10
DEP Protocol: The the PN532 configured in target mode does not support the command received from the initiator (the command received is not one of the following: ATR_REQ, WUP_REQ, PSL_REQ, DEP_REQ, DSL_REQ, RLS_REQ, ref. [1]).	0x12
DEP Protocol / Mifare / ISO/IEC 14443-4: The data format does not match to the specification. Depending on the RF protocol used, it can be: <ul style="list-style-type: none"> • Bad length of RF received frame, • Incorrect value of PCB or PFB, • Invalid or unexpected RF received frame, • NAD or DID incoherence. 	0x13
Mifare: Authentication error	0x14
ISO/IEC 14443-3: UID Check byte is wrong	0x23
DEP Protocol: Invalid device state, the system is in a state which does not allow the operation	0x25
Operation not allowed in this configuration (host controller interface)	0x26
This command is not acceptable due to the current context of the the PN532 (Initiator vs. Target, unknown target number, Target not in the good state, ...)	0x27
The the PN532 configured as target has been released by its initiator	0x29
The PN5321 and ISO/IEC 14443-3B only: the ID of the card does not match, meaning that the expected card has been exchanged with another one.	0x2A
The PN5321 and ISO/IEC 14443-3B only: the card previously activated has disappeared.	0x2B
Mismatch between the NFCID3 initiator and the NFCID3 target in DEP 212/424 kbps passive.	0x2C
An over-current event has been detected	0x2D
NAD missing in DEP frame	0x2E

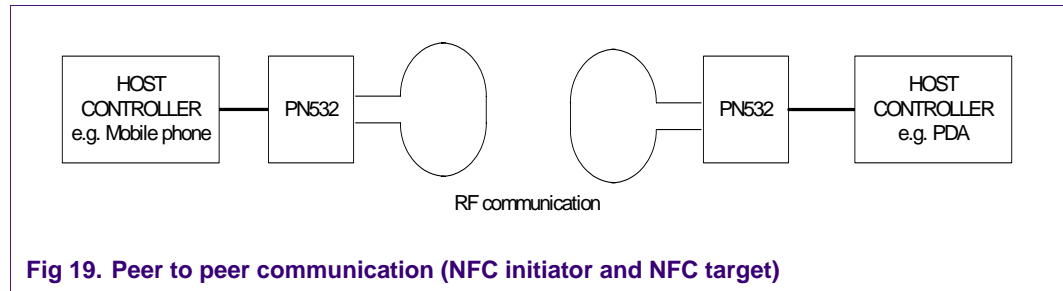
3.3 The PN532 various modes

This paragraph summarizes the PN532 functionalities and shows which commands are associated to them.

The PN532 firmware implements functions to easily behave:

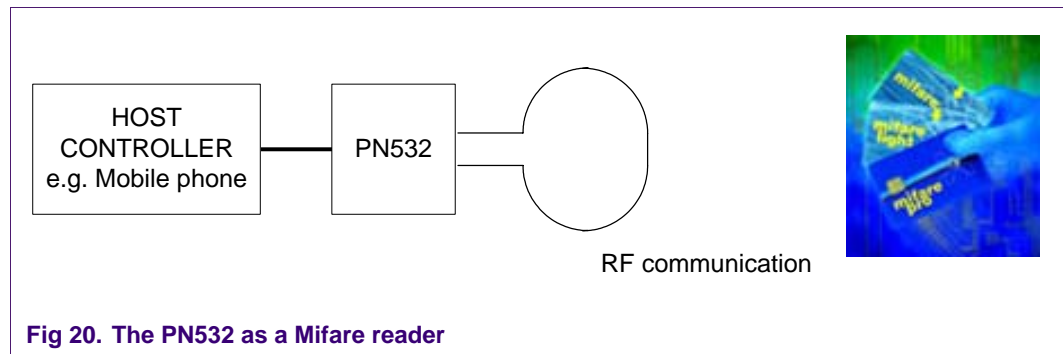
- **As a NFC initiator or a NFC target (according to NFCIP-1 specification).**

In this mode, RF communication is according to NFCIP-1 specification. Two NFC devices can communicate together (peer to peer communication). One device is the initiator: it starts the exchange and chooses the mode. The other device is the target. Passive mode or active mode can be used. In active mode, each device generates RF field when it transmits data (and switches RF field off at the end of the transmission). In passive mode, only the initiator generates RF field. The target answers in a load modulation scheme.



- **As a Mifare reader (Mifare protocol).**

In this mode, RF communication is according to Mifare specification. the PN532 behaves as a Mifare reader. It can communicate with Mifare cards.



The PN532 has been tested with Mifare 1k, 4k, Ultralight, and DesFire cards.

- **As a T=CL reader (ISO/IEC 14443-4 protocol)**

In this mode, RF communication is according to ISO/IEC 14443-4 specification. the PN532 behaves as an ISO/IEC 14443-4 reader. It can communicate with ISO/IEC 14443-4 cards (only ISO compliant cards are supported).

(The PN532 has been tested with CD97BX, CD light, Desfire, P5CN072 (SMX) as ISO/IEC 14443-4 (with JCOP OS))

- **As a Jewel card reader**

The PN532 can communicate with Innovision Jewel cards. It has been tested with IRT5001 card.

- **As a FeliCa reader (FeliCa protocol)**

In this mode, RF communication is according to FeliCa specification. the PN532 has been tested with FeliCa RCS_860 and RCS_854

- **As a ISO/IEC 1443-A card**

The PN532 is able to answer to an ISO/IEC 1443-4A reader. It contains a predefined ATS (only historical bytes are configurable). In this mode, ATS will be sent automatically to the reader which has sent a RATS. the PN532 handles automatically waiting time extension (S(WTX)). The command from the reader is transmitted to the host controller. The host controller builds the response and transmits it to the PN532. the PN532 handles the encapsulation in ISO/IEC 1443-4 frame. Maximum up to 256 data bytes can be transmitted between the reader and the the PN532 ("short APDU").

- **As a virtual card (in combination with a secure smart card)**

In this mode, the PN532 is combined with a secure smart card. An external reader sees the set the PN532+secure smart card as a contactless card.

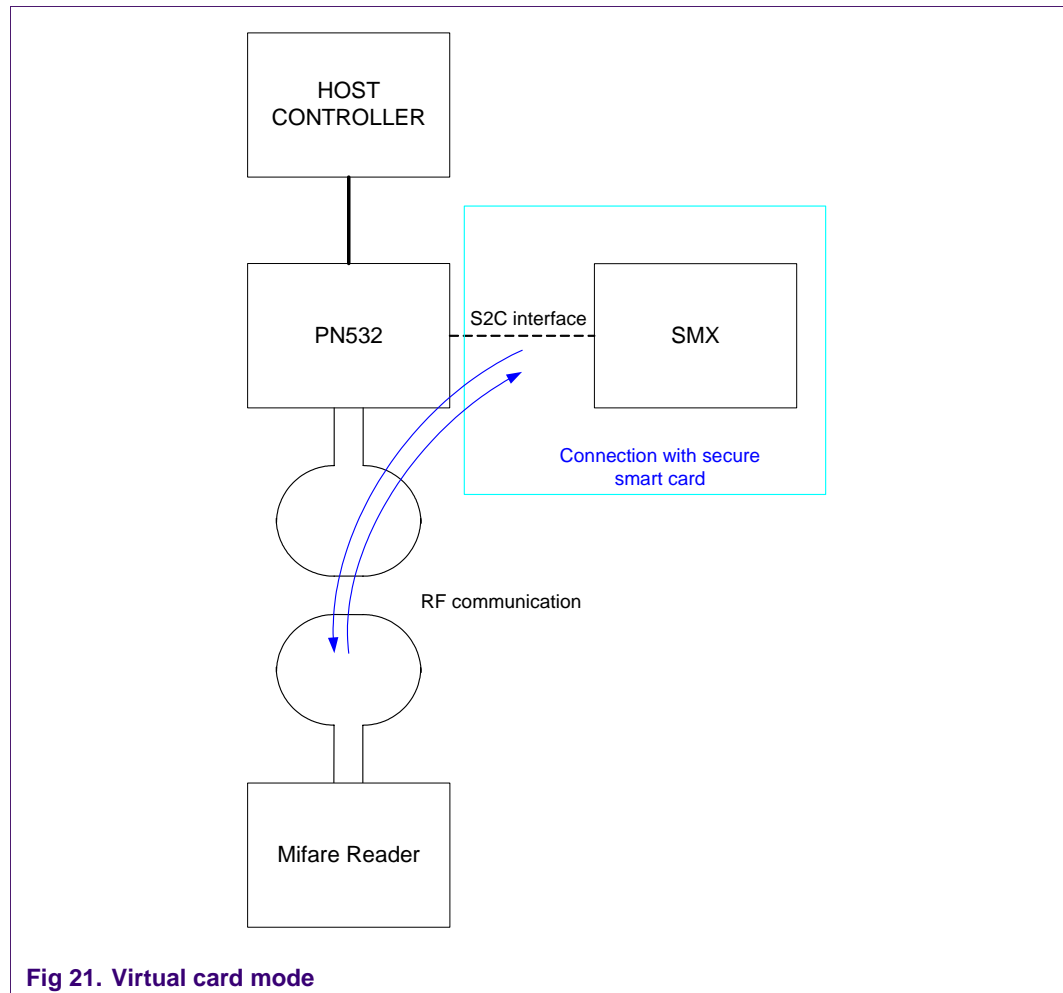


Fig 21. Virtual card mode

RFConfiguration command

This command, described in reference 1, allows changing some registers settings than can influence the RF communication. The default values are described in reference 1. The tuning depends on the environment, on the antenna and on the communication mode.

Very few commands are needed to set up RF communication between the PN532 and another device (reader, card, or other NFC device). The PN532 executes **different RF processes**, depending on the type of communication, **but from the host controller to the PN532, same commands are used** (whatever the baudrate, the mode etc):

Paragraphs below explain which functions to use to communicate in each mode.

3.3.1 How to use the PN532 as a Mifare reader?

Typical sequence (example):

- Scanning for targets (cards) in the field,
- Possibly authenticate with the card,
- Read out the card memory (or any other Mifare commands, such as write),
- Halt the card, select another one, and perform any Mifare command with it

This typical sequence can be performed with the following commands:

- **InListPassivTarget**, to initialise one (several) cards (maximum two cards at the same time)
- **InDataExchange**, to send Mifare commands
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card (this is optional, see paragraph 3.3.7.3 on page 53).

Warning:

In case the card initialized indicates it supports ISO/IEC 14443-4 protocol (bit 5 of SAK, cf. ISO/IEC 14443-3 specification), InListPassiveTarget command of the PN532 performs automatically ISO/IEC 14443-4 activation (i.e. RATS sending). To disable automatic RATS sending, SetParameter command must be used (cf. REFERENCE 1).

Table 1. SetParameter command usage to enable or disable automatic RATS sending (ISO/IEC 14443-4 mode)

Action	Command ¹	Command explanation	Response	Response explanation
Disable automatic sending of RATS command	12 24	Command code: SetParameters Automatic ATR_RES = 1 Automatic ISO/IEC 1443-4A card emu = 1	13	Response command code
RATS will not be performed automatically by next InListPassiveTarget.command, even if the card indicates it supports ISO 14443-4				
Enable automatic sending of RATS command (default configuration of the PN532 at power up)	12 34	Command code: SetParameters Automatic ATR_RES = 1 Automatic RATS = 1 Automatic ISO/IEC 1443-4A card emu = 1	13	Response command code
RATS will be performed automatically by next InListPassiveTarget.command, if the card indicates it supports ISO 14443-4				

Mifare commands are briefly described hereafter. Refer to Mifare card documentation to have a more detailed description of the Mifare command set

The Mifare specific command byte **Cmd** may take one of the possible values:

- 60h / 61h Authentication A / Authentication B (Mifare Standard)
Performs authentication sequence.

- 30h 16 bytes reading
Read one data block (16 bytes) at the selected address of the card.

- A0h 16 bytes writing (Mifare Standard)
Write one data block (16 bytes) to the selected address of the card.

- A2h 4 bytes writing (Mifare Ultralight)
Write one data block (16 bytes) to the selected address of the card.

C1h Increment
Increment the value block at the selected address of the card. The data structure of the value block must be written in advance with a standard write command.

Data structure

Byte	0 3	4 7	8 11	12	13	14	15
	Value	Value complement	Value	Addr	Addr complement	Addr	Addr complement

C0h Decrement
Decrement the value block at the selected address of the card. The data structure of the value block must be written in advance with a standard write command.

B0h Transfer:
This function writes the prior calculated at the selected address of the card. It must be called directly after Increment, Decrement or Restore.

C2h Restore.
This function restores the value block at the selected address of the card.

Mifare Ultralight cards

They support only A2h and 30h commands.

Command code	Command Parameter	Command Data field	Response
30h	1 byte address	-	16 bytes
A2h	1 byte address	4 bytes	-

Mifare Standard cards

Authentication is required before any access to Mifare Standard memory.

Command code	Command Parameter	Command Data field	Response
60h	1 byte address	KeyA (6 bytes, default value FFh) followed by UID (4 bytes)	-
61h	1 byte address	KeyB (6 bytes, default value FFh) followed by UID (4 bytes)	-
30h	1 byte address	-	16 bytes
A0h	1 byte address	16 bytes	-
C1h	1 byte address	4 byte increment value (lower byte first)	-
C0h	1 byte address	4 byte decrement value (lower byte first)	-
B0h	1 byte address	-	-
C2h	1 byte address	-	-

Refer to Mifare card documentation to have a more detailed description of the Mifare command set

The table on next page shows how to use some of those commands (how to fill the parameters, which bytes are returned in response...).

Table 2. The PN532 as a Mifare reader

Action	Command ¹	Command explanation	Response	Response explanation	
Scan for 2 targets in the field and initialize them	4A	Command code: InListPassivTargets	4B	Response command code	
	02	Number of cards to initialize (if present in the field)	02	2 targets detected	
	00	Baud rate = 106 kbits/sec.	01	01	Target number 1
			04 00	04 00	SENS_RES(2) of target 1
			08	08	SEL_RES(2) of target 1
			04	04	NFCID1 length = 4 bytes
			12 67 58 32	12 67 58 32	NFCID1 of target 1
			02	02	Target number 2
			44 00	44 00	SENS_RES(2) of target 2
			00	00	SEL_RES(2) of target 2
08	08	NFCID1 length = 8 bytes			
88 04 B6 E4 00 00	88 04 B6 E4 00 00	NFCID1 of target 2			
00 00	00 00				
Target 1 is a Mifare Standard card and target 2 is a Mifare Ultralight card. The 2 cards are initialised. Card 1 is in halt state.					
Authenticate with Mifare Standard card	40	Command code: InDataExchange	41	Response command code	
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)	
	60	Mifare cmd code (authenticate)			
	07	Mifare cmd code (authenticate)			
	FF FF FF	Mifare address			
	FF FF FF 12 67 58 32	Default authentication key (last bytes are NFCID1 bytes)			
Card2 has been put in halt state. Card1 has been wake up and authentication has been performed.					
Read out card 1 memory	40	Command code: InDataExchange	41	Response command code	
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)	
	30	Mifare cmd code (read 16 bytes)	EE EE EE EE EE EE	Bytes read (example)	
	04	Mifare address	EE EE EE		
			EE EE EE EE EE EE		
EE EE EE					
EE EE EE					
Card2 is still in halt state. Any Mifare commands, for example Read 16 bytes, can be sent to card 2 with InDataExchange function.					
Halt card 1, select card 2 and write in card 2 memory	40	Command code: InDataExchange	41	Response command code	
	02	The cmd is intended to target number 2	00	Status = 0 (OK, no error)	
	A2	Mifare cmd code (write 4 bytes)			
	08	Mifare address			
	FF FF FF	Bytes to be written			
	FF				
Card1 has been put in halt state. Card 2 has been wake up. Any Mifare commands, for example Write 4 bytes, can be sent to card 2 with InDataExchange function.					
Halt card 2	44	Command code: InDeselect	45	Response command code	
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)	
HALT REQ is sent with InDeselect command.					

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

⁽²⁾ SENS_RES and SEL_RES coding is described in ISO/IEC 18092 specification.

3.3.1.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, the PN532 is configured to retry to detect a card as long as there is no card detected. It can be changed using RFConfiguration command, item 5 (MaxRtyPassiveActivation parameter, c.f. UM0502-05).

If there is no card in the field, a timeout occurs after 5 ms. Either the PN532 retries to find a card, if MaxRtyPassiveActivation > 1, or it sends a response to its host controller, indicating that zero target has been found.

Communication phase (InDataExchange command)

By default, the timeout is set to 51.2 ms. It can be changed using RFConfiguration command item 2 (UM0501-02 page 80).

Deactivation phase (InDeselect/InRelease command)

InDeselect or InRelease commands perform a HALTA request. The return status is always "No error" (00h), even if the card did not respond (within 5 ms).

Note: It is not needed to use InDeselect (and InSelect) command to handle two cards. Indeed, when using InDataExchange command, the PN532 automatically wakes up the card corresponding to the desired TgNb, and automatically put in HALT state the other one.

3.3.2 How to use the PN532 as a T=CL reader (ISO/IEC 14443-4)?

A typical sequence can be:

- Scan for targets (cards) in the field, (initialisation and activation of the card)
- Perform any T=CL command
- Deselect the card

This typical sequence can be performed with the following commands:

- **InListPassivTarget**, to initialise one (several) cards (maximum two cards at the same time).

In case of Type A card, the RATS is sent automatically by this command. CID parameter is set to 0 and FSDI is set to 5 (→ FSD = 64 bytes).

In case of Type B card, the default method used is the timeslot one. It can be changed by indicated in the parameter of this command that the probabilistic polling method must be used.

- **InDataExchange**, to send ISO/IEC 14443-4 commands
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card (this is optional, see paragraph 3.3.7.3 on page 53).

Table 3. The PN532 as a ISO/IEC 1443-4 reader. Type A card activation

Action	Command ⁴	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	00	Baud rate = 106 kbits/sec. Type A.	01	Target number 1
			04 07	SENS_RES(1) of target 1
			28	SEL_RES(1) of target 1
			04	NFCID1 length = 4 bytes
			00 D4 1E 92	NFCID1 of target 1
			0D 77 80 81 02 00	ATS
			73 C8 40 13 00 90	
			00	
Bit 5 of SEL_RES indicates the target is ISO/IEC 14443-4 compliant. In that case the PN532 automatically sends the RATS(2). ATS is indicated in the response.				
Read 16 bytes	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	00 B0 81 00	ISO/IEC 14443-4 command	00 11 22 33 44 55	Response of the card
	10		66 77 88 99 AA BB CC DD EE FF 90 00	
ISO/IEC 14443-4 commands, for example Read 16 bytes command, can be sent with InDataExchange command(3).				
Deselect the card	44	Command code: InDeselect	45	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
S(deselect)REQ is sent with InDeselect command.				

⁽¹⁾ SENS_RES and SEL_RES coding is described in ISO/IEC 18092 specification. Please refer to this document for further details.

⁽²⁾ The automatic sending of RATS can be disabled with SetThe PN532Parameters command. Please refer to the PN532 User manual (cf. References table on page 4).

⁽³⁾ the PN532 handles chaining, time out extension, and error handling, according to ISO/IEC 14443-4 protocol.

⁽⁴⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

Table 4. The PN532 as a ISO/IEC 1443-4 reader. Type B card activation (timeslot method)

Action	Command ⁴	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	03	Baud rate = 106 kbits/sec Type B.	01	Target number 1
	00	AFI	50 01 02 03 04 00	ATQB_RES (12 bytes)
		No other parameter : default timeslot method will be used.	00 00 00 00 00 00	
			01	ATTRIB_RES length
			01	ATTRIB_RES
Type B card is activated. The default method used is the timeslot approach.				
Read 16 bytes	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	94 A4 00 00	ISO/IEC 14443-4 command	85 17 00 01 00 00	Response of the card
	02 3F 00		00 10 10 00 00 01	
			03 00 00 00 7E 7E	
			7E 00 00 00 00 00	
			00 90 00	
ISO/IEC 14443-4 commands, for exampleSelect File with CD light card, can be sent with InDataExchange command(3).				
Deselect the card	44	Command code: InDeselect	45	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
S(deselect)REQ is sent with InDeselect command.				

Table 5. The PN532 as a ISO/IEC 1443-4 reader. Type B card activation (probabilistic polling method)

Action	Command ⁴	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	03	Baud rate = 106 kbits/sec Type B.	01	Target number 1
	00	AFI	50 01 02 03 04 00	ATQB_RES (12 bytes)
		Method = 01: probabilistic polling method will be used	00 00 00 00 00 00	
			01	ATTRIB_RES length
			01	ATTRIB_RES
Type B card is activated. The method used is the probabilistic approach.				

3.3.2.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, the PN532 is configured to **retry** to detect a card as long as there is no card detected. It can be changed using RFConfiguration command, item 5 (MaxRtyPassiveActivation parameter). The command is described in reference 1.

If there is no card in the field, a timeout occurs after **5 ms**. Either the PN532 retries to find a card, if MaxRtyPassiveActivation > 1, or it sends a response to its host controller, indicating that zero target has been found.

Communication phase (InDataExchange command)

It depends on value returned by the card (FWT), as specified in ISO/IEC 14443-3 and -4. The waiting time extension mechanism is fully embedded inside the PN532. The error handling and the chaining are also fully managed by the PN532.

Deactivation phase (InDeselect/InRelease command)

InDeselect or InRelease commands perform a S(Deselect) request.

Note: It is not needed to use InDeselect (and InSelect) command to handle two cards. Indeed, when using InDataExchange command, the PN532 automatically wakes up the card corresponding to the desired TgNb, and automatically put in HALT state the other one.

3.3.3 How to use the PN532 as a DESfire cards reader?

DESfire cards can be read/write as other ISO/IEC 1443-4 cards.

Table 6. The PN532 as a DESfire card reader

Action	Command ⁴	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	00	Baud rate = 106 kbits/sec.	01	Target number 1
			44 03	SENS_RES ⁽¹⁾ of target 1
			20	SEL_RES ⁽¹⁾ of target 1
			08	NFCID1 length = 8 bytes
			88 04 4A 5B 09 2C	NFCID1 of target 1
			1C 80	
			06 75 77 81 02 80	
				ATS

Bit 5 of SEL_RES indicates the target is ISO/IEC 14443-4 compliant.
 In that case the PN532 automatically sends the RATS⁽²⁾. ATS is indicated in the response.

Get application ID	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	6A	DESfire command: GetApplicationID	00 01 00 00 02 00	Response of the card
			00 03 00 00 40 00	
			00 04 00 00 05 00	
			00 06 00 00	

DESfire commands, for example GetApplicationID command, can be sent with InDataExchange command⁽³⁾.

Select application	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	5A 06 00 00	SelectApplication 06 00 00	00	Response of the card

3.3.3.1 Timeout and number of retries

Cf. paragraph 3.3.2.1.

3.3.4 How to use the PN532 as a FeliCa reader?

A typical sequence can be:

- Scan for targets (cards) in the field.
- Exchange data with the card.

This typical sequence can be performed with the following commands:

- **InListPassivTarget**, to initialise one (several) cards (maximum two cards at the same time)
- **InDataExchange**, to transfer data/command bytes to the card (The PN532 does not embed FeliCa protocol: it has to be included in the data bytes).
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card.

Table 7. The PN532 as a FeliCa reader

Action	Command ¹	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	01	Baud rate = 212 kbits/sec.	01	Target number 1
	00 FF FF	Payload field of polling request	12	POL_RES length
	00 00		01	Response code
			01 01 06 01 46 05 C3 1A	NFCID2
			04 01 4B 02 4F 49 93 FF	PAD
FeliCa™ card has been initialised.				
Exchange data with the card	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	06	Length of data + 2	06	Response of the card
	F0	FeliCa™ command (echo back) 2	F0	
	00 FF AA BB	Data	00 FF AA BB	
The PN532 transfers the data. Len and Cmd bytes of FeliCa™ protocol must be present in the buffer.				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

⁽²⁾ Exemple of FeliCa command. Not all the FeliCa cards support this command. Refer to FeliCa card specification.

3.3.4.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, the PN532 is configured to retry to detect a card as long as there is no card detected. It can be changed using RFConfiguration command, item 5 (MaxRtyPassiveActivation parameter). The command is described in reference 1

If there is no card in the field, a timeout occurs after $2.42 \text{ ms} + (TSN+1) * 1.21 \text{ ms}$. TSN is the Time Slot Number field of the command.

Either the PN532 retries to find a card, if MaxRtyPassiveActivation > 1, or it sends a response to its host controller, indicating that zero target has been found.

Communication phase (InDataExchange command)

By default, the timeout is set to 51.2 ms. It can be changed using RFConfiguration command item 2. The command is described in reference 1.

Deactivation phase (InDeselect/InRelease command)

InDeselect or InRelease commands perform no request. The return status is always "No error" (00h),

3.3.5 How to use the PN532 as a Jewel cards reader ?

A typical sequence can be:

- Scan for targets (cards) in the field.
- Exchange data with the card.

This typical sequence can be performed with the following commands:

- **InListPassivTarget**, to initialise one (several) cards (maximum two cards at the same time)
- **InDataExchange**, to transfer data/command bytes to the card
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card.

Table 8. The PN532 as a Jewel card reader

Action	Command ¹	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	04	Baud rate = 106 kbits/sec, type = Innovision Jewel	01	Target number 1
			04 00	ATQA_RES
			92 2E 58 32	Jewel ID
Jewel card has been initialised.				
Exchange data with the card	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	00	Command code	01..FF	Response of the card (exemple : 255 bytes, 01 to FF. Not all bytes are shown here)
The PN532 transfers the data.				

InDataExchange command is used to send the Jewel commands, described in Jewel documentation.

Jewel command code	Command description
0x00	Read all bytes
0x01	Read a single byte
0x1A	Write-no-Erase a single byte
0x1C	Write-no-Erase 8 bytes
0x53	Write-with-Erase a single byte
0x55	Write-with-Erase 8 bytes

3.3.6 How to use the PN532 as a reader for several types of cards (or targets)?

In case different types of cards can be used to communicate with the PN532 as reader, InAutopoll command, described in reference 1, allows polling for several types of cards.

The host controller can poll for Mifare cards, FeliCa cards, Jewel cards, ISO/IEC 14443-4 cards, NFC targets.

A maximum of two cards, or one card and one NFC target, can be handled by the PN532 (except in case of FeliCa card, where only one card can be detected with InAutopoll command).

The latest card/target detected remains in active mode, whereas the first one is put in HALT/SLEEP state.

The host controller can specify up to 15 different modes to be polled (combining the type such as Mifare, FeliCa, ISO/IEC 14443-4, Jewel, the baudrate (106, 212 or 424 kbps), and possible the mode (active or passive) for NFC target).

The host controller also specifies the number of polling to be performed (1 to 254 or infinite), and the polling period (i.e. the time duration of one polling, per unit of 150 ms).

After InAutoPoll command has been used, the card or the target is ready to communicate with InDataExchange command.

3.3.7 How to use the PN532 in a NFC peer-to-peer communication?

3.3.7.1 How to use the PN532 as an initiator in a NFC peer-to-peer communication?

Goal: exchange data between two NFC devices.

The host controller of the initiator chooses the mode (active or passive) and the baudrate of the communication. Same command codes are used whatever the mode (only parameters are different).

Typical sequence (example 1):

- Initialise and activate a target
- Exchange some data with the target
- Release the target

Another typical sequence (example 2): compared to example 1, initialisation and activation are done “step by step”:

- Scan for targets in the field
- Activate NFC target
- Perform parameters selection (PSL REQ)
- Exchange some data with the target

Those typical sequences can be performed with the following commands:

- **InJumpForDEP**, or **InJumpForPSL**, to initialise and activate the target (active or passive mode)

OR **InListPassivTarget** (to initialise), and **InATR** (to activate) the target (passive mode only).

- **InPSL**, to change the baud rate (except if InJumpForDep has been used for activation)
- **InDataExchange**, to exchange data with the target (NFCIP-1 transport protocol fully embedded). Please read also paragraph 0.
- **InRelease** to release the target (3.3.7.3 on page 53).

Table 9. The PN532 as a NFC initiator (“reader”) example 1

Action	Command ¹	Command explanation	Response	Response explanation
Initialize and activate a target	56	Command code: InJumpForDep	57	Response command code
	01	Mode (1= active mode)	00	Status (0=no error)
	02	Baud rate = 424 kbits/sec.	01	Target number 1
	01	Optional field present (Payload)	AA 99 88 77 66 55	ATR_RES received (except cmd0 and cmd1 bytes)
	00 FF FF	Payload field of polling request	44 33 22 11	
	00 00		00 00 00 09 01 22	
Initiator has chosen the active mode. the PN532 performed automatically activation and possibly parameter selection (ATR_REQ and possibly PSL_REQ) ⁽²⁾ .				
Exchange data with the card	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	00 01 02 03	Data	99 88 77	Data (Response of the target)
	04 05 06 07			
	08 09 0A			
	0B 0C 0D			
	0E 0F			
The PN532 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.				
Deselect	44	Command code: InDeselect	45	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
DSL_REQ is sent with InDeselect command.				
Select	54	Command code: InSelect	55	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
WUP_REQ is sent with InSelect command.				
Release the target	52	Command code: InRelease	53	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
RLS_REQ is sent with InRelease command.				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

⁽²⁾ Would passive mode have been chosen by the initiator, the PN532 would have performed initialisation (POL_REQ at 212/424 kbits/sec or SENS_REQ, SDD, SEL_REQ at 106 kbits/sec), plus activation

(ATR_REQ) and possible parameter selection (PSL_REQ).

Table 10. The PN532 as NFC initiator (“reader”) example 2

Action	Command ¹	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: InListPassivTargets	4B	Response command code
	01	Number of cards to initialize (if present in the field) = 1	01	1 target detected
	02	Baud rate = 212 kbits/sec.	01	Target number 1
	00 FF FF		12	POL_RES length
	00 00	Payload field of polling request	01	Response code
			01 FE A2 A3 A4 A5 A6 A7	NFCID2
			C0 C1 C2 C3 C4 C5 C6 C7	PAD
The target has been initialised (in passive mode) at the requested baud rate.				
Activate the target	50	Command code: InATR	51	Response command code
	01	The cmd is intended to target number 1	00 AA 99 88 77 66 55 44 33 22 11 00 00 00 09 01	Status = 0 (OK, no error)
The target has been activated.				
Change parameters	4E	Command code: InPSL	4F	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	00 00	New baud rate = 106 kbits/sec		
The baud rate has been changed.				
Exchange data with the card	40	Command code: InDataExchange	41	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
	00 01 02 03	Data	99 88 77	Data (Response of the target)
	04 05 06 07			
	08 09 0A			
	0B 0C 0D			
	0E 0F			
The PN532 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.				
Release the target	52	Command code: InRelease	53	Response command code
	01	The cmd is intended to target number 1	00	Status = 0 (OK, no error)
RLS_REQ is sent with InRelease command.				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

About PSL (Parameter Selection)

When using InJumpForDEP command, the PN532 performs automatically PSL_REQ if the target indicates a Length Reduction value corresponding to a buffer greater than 64 bytes. *But the actual LR used remains 64 bytes since the PN532 does not support more. Moreover, the baudrate is not changed automatically.*

However, as defined in NFCIP-1 specification, further PSL_REQ sending is not allowed. Consequently, if the user wants to change the baudrate (in reception and in transmission), he has to use **InJumpForPSL** command, followed by **InPSL** command.

About InDeselect command

During Data Exchange Protocol (as defined in NFCIP-1), the host controller can use this command to resynchronise target packet numbers (PNI).

Example:

The initiator sends InDataExchange command, an error is returned. Maybe the PNI of the response is incorrect. The initiator sends InDeselect command followed by InDataExchange. The PNI are re-synchronised.

Timeout and number of retries

- **Initialization phase in passive mode (InListPassiveTarget, InJumpForDEP in passive, InJumpForPSL in passive)**

By default, the PN532 is configured to retry to detect a card as long as there is no target detected. It can be changed using RFConfiguration command, item 5 (MaxRtyPassiveActivation parameter).

The timeout depends on the baudrate. At 106 kbps, paragraph 3.3.1.1 applies. At 212 or 424 kbps, paragraph 3.3.4.1 applies.

Either the PN532 retries to find a card, if MaxRtyPassiveActivation > 1, or it sends a response to its host controller, indicating that zero target has been found.

- **Activation phase in passive mode (InATR)**

The default timeout is set to 102.4 ms. It can be changed using RFConfiguration command item 2.

By default, the PN532 is configured to retry an infinite number of times in case no targets are responding. It can be changed using RFConfiguration command, item 5 (MaxRtyATR parameter).

- **Activation phase in active mode (InJumpForDEP in active, InJumpForPSL in active)**

The default timeout is set to 102.4 ms. It can be changed using RFConfiguration command item 2.

3.3.7.2 How to use the PN532 as a target in a NFC peer-to-peer communication?

In this mode, the PN532 is configured as target, meaning it keeps waiting for an initiator command.

The PN532 has no memory to emulate a card. After activation, all data received must be transferred to the host controller. The host controller gets the data, analyse them, and provide the response to the PN532. the PN532 transfers the response from the host to the initiator. Initialisation/activation is handled automatically by the PN532.

Typical exchange:

- Be ready to respond to an initiator, what ever the mode and the baud rate (be able to send SENS_RES, NFCID1, SEL_RES or POL_RES and/or ATR_RES)
- Get data from the initiator and transfer them to the host controller
- Transfer response from the host to the initiator

This typical sequence will be (most of the time) performed with the following commands:

- **TgInitAsTarget**, to configure the PN532 as a target,
- **TgGetData**, to wait for data coming from the initiator,
- **TgSetData**, to respond to the initiator.

Table 11. The PN532 as NFC target

Action	Command ¹	Command explanation	Response	Response explanation
Be ready to respond to an initiator, what ever the mode and the baud rate	8C 00 08 00 12 34 56 40 01 FE a2 a3 a4 a5 a6 a7 c0 c1 c2 c3 c4 c5 c6 c7 FF FF AA 99 88 77 66 55 44 33 22 11 00 00	Command code: TgInitAsTarget Accepted modes : 0 = all SENS_RES NFCID1 SEL_RES Parameters to build POL_RES (16 bytes) NFCID3t (10 bytes)	8D 22 11 D4 00 01 FE A2 A3 A4 A5 A6 A7 00 00 00 00 00	Response command code Mode : passive mode, 424 kbits/s Initiator command received (ATR_REQ)
The target was waiting for any initialisation command. In this example, it has been initialised at 424 kbit/s in passiv mode. POL_Res and ATR_RES have been automatically sent by the PN532				
Wait for data to be transferred to the host controller	86	Command code: TgGetData	87 00 98 76	Response command code Status = 0 (OK, no error) Data received
The target received some data from the initiator.				
Sends data response from the host controller	8E	Command code: TgSetData No data sent back to the controller	8F 00	Response command code Status = 0 (OK, no error)
The target responded to the initiator (it can possibly send some data).				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

What are default timeout values of the PN532 as a target?

WT = 09h (ATR_RES parameter) → RWT = 154ms approx.

RTOX = 07h (Timeout extension request parameter) → RWT_{INT} = 1078ms approx.

How to fill TgInitAsTarget parameters?**Mode** (1 byte)

Mode = 00h: any command (after initialisation if passive mode) is accepted.

Mode = 02h: only ATR_REQ (after initialisation if passive mode) is accepted, i.e. only NFC transport protocol communication will done.

Mode = 04h: only RATS (after initialisation if passive mode) is accepted, i.e. only ISO/IEC 1443-4 transport protocol communication will done.

The three mode can be combined.

Mifare params (6 bytes)

SENS_RES: (2 bytes) **bit 7 and bit 6 must be set to 0** (NFCID1 size = single)

NFCID1t: 3 bytes configurable (NFCID1 is 4 bytes, the first byte is fixed to 08 according to ISO/IEC 18092 specification).

SEL_RES: **bit 6 must be set to 1** to indicate that NFC transport protocol is supported.
Typical value SEL_RES = 40h.

FeliCa™ params (18 bytes)

NFCID2t: 8 bytes. **First two bytes must be set to 01h FEh.**

PAD: 8 bytes

System code: 2 bytes. Typical value = FFh FFh.

NFCID3t (10 bytes)**Gt length** (1 byte)

Length of general bytes (used in NFC transport protocol). It must be between 0x00 and 0x2F.

Gt (maximum 47 bytes)

General bytes.

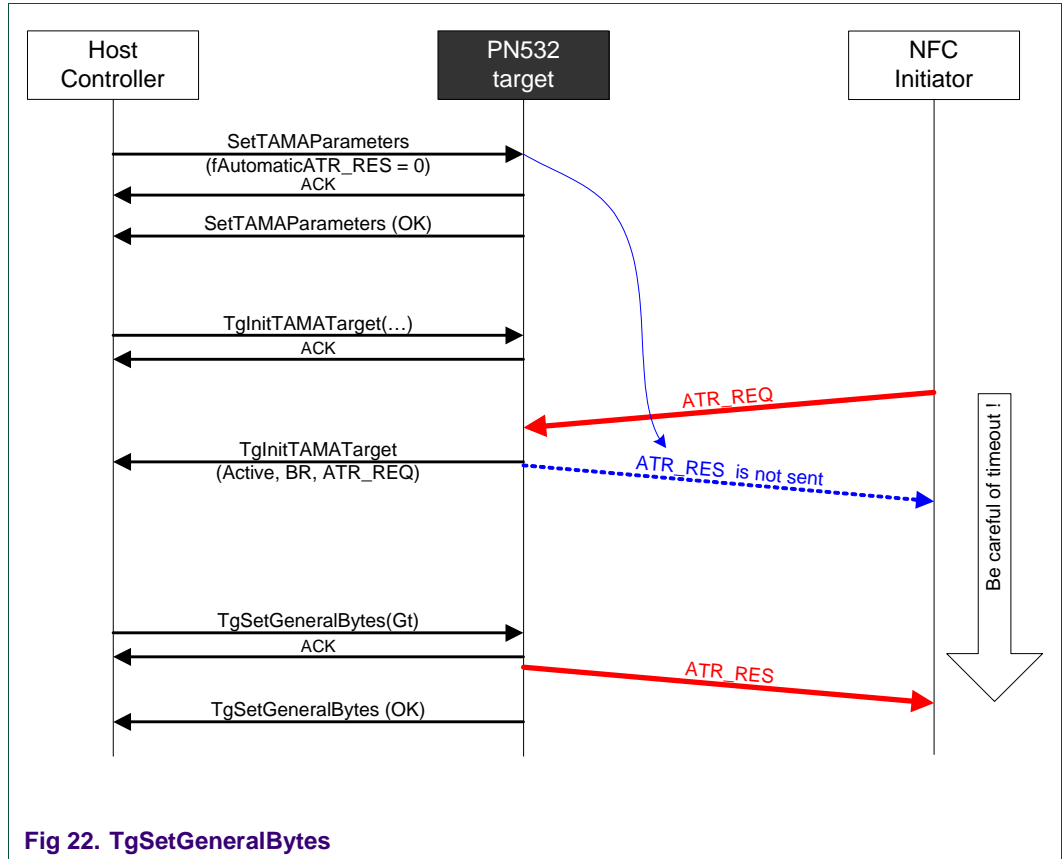
Optional field.

The target uses these bytes to build the ATR_RES, as defined in NFCIP-1 specification. The host controller can provide the target with these bytes:

- Either at start up of target mode, i.e. in TgInitAsTarget parameters.
- Or after having received the ATR_REQ. In that case, the bytes are transmitted from the host controller to the PN532 using TgSetGeneralBytes command. It is useful to use this command if the general bytes values of the ATR_RES are set

depending the received ATR_REQ.

In that case, it is required to use first SetParameters command to disable automatic sending of ATR_RES upon reception of ATR_REQ. The ATR_RES will be sent by TgSetGeneralBytes command.



Tk length (1 byte)

Length of Historical bytes (used in ISO/IEC 14443 protocol)

Tk (maximum 48 bytes)

Optional field.

Tk contains the historical bytes to be used in the ATS when the PN532 is in ISO/IEC 14443-4 PICC emulation mode.

3.3.7.3 Handling of several cards/targets

The PN532 can handle 2 cards “at the same time”, or 1 card and 1 NFC target.

The PN532 memorizes the ID of the target/card and some information about it. It attributes a logical number to each card/target detected. The host controller can communicate with them using InDataExchange command and the appropriate logical number. **The host controller does not need to take care of putting card/target 1 into SLEEP state before communicating with card/target 2: InDataExchange command does it automatically.**

However, the PN532 provides two commands corresponding to relevant RF requests (depending on the mode, the baudrate, and the protocol)

InDeselect performs DSL_REQ or SLP_REQ or S(deselect) REQ (depending on the target)

InSelect performs ALL_REQ or WUPA or POL_REQ or ATR_REQ (depending target)

3.3.7.4 Transfer of large amount of data

Chaining mechanism

- from initiator to target:

Large amount of data are sent by the initiator with InDataExchange function, in packets of 252 bytes of data. The initiator must send InDataExchange command as many times as necessary to transfer the complete amount of data.

The target must perform TgGetData and TgSetData functions as many times as necessary to retrieve all packets sent by the initiator.

Metachaining mechanism

- From initiator to target:

One bit called MI (more information), in InDataExchange first parameter, indicates to the target if data received are part of a large block. In that case, the target can directly continue the exchange with TgGetData (no TgSetData needed).

- From target to initiator:

The target can provide the initiator with large amount of data using TgSetMetaData function. The initiator has sent a InDataExchange function. The response to the initiator is sent via TgSetMetaData function instead of TgSetData function. In that case, one bit indicates to the initiator that some data are still available at target side. The initiator shall go on with a InDataExchange function (with no data sent from the initiator to the target). Last packet of data will be transferred with TgSetData function.

Refer to the PN532 User manual (reference 1) for detailed explanation.

3.3.7.5 Effective throughput

The baudrate on the RF interface is 106 or 212 or 424 kbps (bit rate as defined in NFCIP1 specification).

The time to transfer a certain amount of *useful* data (i.e. excluding NFC protocol bytes and host link protocol bytes), between two host controllers, each connected to the PN532, depends on several parameters:

- The RF baudrate
- The amount of data:
 - o The PN532 length reduction¹ is 0: packets size on RF interface is 64 bytes max. The time to transfer the data depends on the number of packets necessary.
 - o The number of packets on host link influences the transfer time as well. the PN532 host protocol limits the size of useful data transmitted at once to 252 bytes using standard frame or 264 using extended frame.
- The CPU frequency
- The link used between the host controller and the NFC device (SPI or HSU or I2C), and the speed chosen for the link (serial baudrate, I2C or SPI frequency)
- The target host controller speed: the initiator host controller can continue transmitting data only after the target indicates it effectively received them. The slowest the target, the longest the transmission time.
- The communication mode (active or passive) **doesn't** influence the performances.

Depending on these parameters, the transmission speed of *useful data* is up to 60 kbps.

3.3.8 How to emulate a ISO/IEC 1443-4A card ?

By default, the ISO/IEC 1443-4 card emulation is enabled. (It can be disabled or enabled using SetParameters command, described in reference 1).

In this mode, the PN532 sends automatically a predefined ATS (when it receives a RATS). The historical bytes of the ATS can be personalized using TgInitTarget command.

The C-APDU coming from the reader will be transmitted to the the PN532 host controller, and the R-APDU from the host controller will be transmitted to the reader via the the PN532. The the PN532 automatically handles waiting time extension (S(WTX)), so that there is no potential problem of timeout whatever the time needed to elaborate the R-APDU.

Only short APDU are supported.

The commands to use to emulate a ISO1443-4A card are:

- **TgInitAsTarget**, to configure the PN532 as a target
 - o One byte can configure the PN532 to act as a ISO/IEC 14443-4A card only, i.e. not to respond to other readers than ISO/IEC 1443-4A readers
 - o The RF request from the reader will be automatically answered by the PN532, including the ATS.
- **TgGetData**, to wait for data coming from the initiator,
 - o The S(WTX) are automatically sent and managed by the PN532
 - o Up to 255 data bytes can be received (short APDU). The complete frame received is up to 261 data bytes (CLA, INS, P1, P2, P3, 255 data bytes, Le)
- **TgSetData**, to respond to the initiator.
 - o Up to 256 data bytes can be sent to the reader (total frame can be up to 258 bytes : 256 data bytes, SW1, SW2).

Table 12. The PN532 as a ISO/IEC 1443-4A card

Action	Command ¹	Command explanation	Response	Response explanation	
Be ready to respond to a reader	8C	Command code: TgInitAsTarget	8D	Response command code	
	04	Accepted modes : 4 = ISO/IEC 14443-4A card only	08	Mode : ISO/IEC 14443-4A card	
	08 00	SENS_RES	E0 80	Initiator command received (RATS)	
	12 34 56	NFCID1			
	60	SEL_RES			
	01 FE	Parameters to build POL_RES (18 bytes including system code)			
	A2 A3 A4				
	A5 A6 A7				
	C0 C1 C2				
	C3 C4 C5				
	C6 C7 FF				
	FF				
	AA 99 88		NFCID3t (10 bytes)		
	77 66 55 44				
	33 22 11				
00	Length of general bytes				
00	Length of historical bytes				
The card has been initialized and ATS have been automatically sent by the PN532					
Wait for data to be transferred to the host controller	86	Command code: TgGetData	87 00 A0 03 00 00 20 00	Response command code Status = 0 (OK, no error) Data received (exemple of command)	
The card received some data from the reader. S(WTX) are automatically sent to the reader as long as the response is not ready					
Sends data response from the host controller	8E 90 00	Command code: TgSetData Respond 90 00 to the reader	8F 00	Response command code Status = 0 (OK, no error)	
The card responded to the reader.					

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

3.3.9 How to use Smart connectivity (combination of the PN532 and SMX)?

The term SmartConnect (Smart Connectivity) describes the usage of a Smart Card IC in connection to the NFC IC.

Combining the PN532 and SMX (P5CN072) allows dealing with application that requires security such as payment applications.

The frame delay time (FDT) can be adjusted in the PN532, thanks to bit 5 of register address 0x630D. (DELAY_MF_SO bit of Manual Rcv register. See reference 2). The embedded software sets DELAY_MF_SO to 1 (when command SAMConfiguration is sent). To put it back to 0, a WriteRegister command can be used, after SAMconfiguration.

In this document, the PN532 is used in combination with a smart card (SMX). S2C interface is used.

The SMX power is supplied by the PN532 (SVDD). In case an external power supply is used, it has to be between 2.7V and 3.3V.

Commands needed to use the PN532 + SMX are:

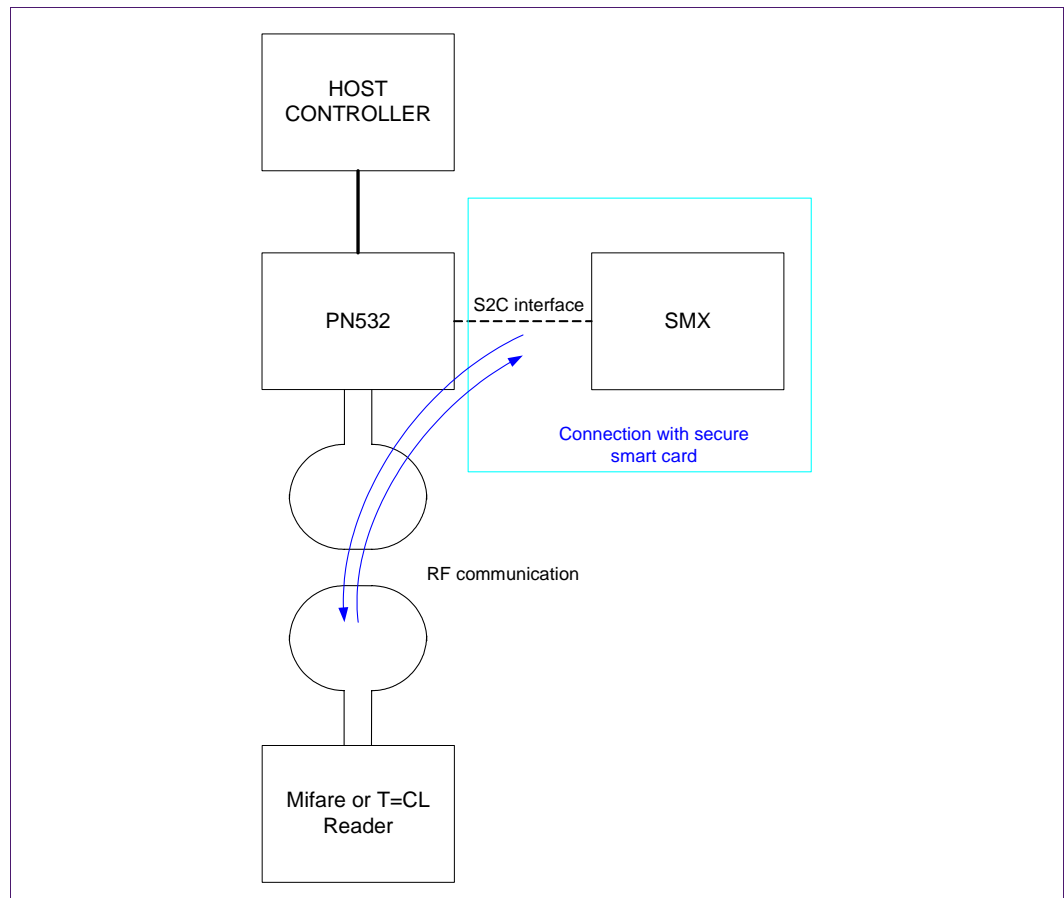
- **SAMConfiguration**, to chose between normal, wired or virtual mode,
- **SetParameters**, to possibly disable automatic RATS sending (T=CL mode).

3.3.9.1 Virtual card mode

In virtual card mode, the PN532 (+SMX P5CN072) is seen as a contact less secure smart card. Only one command, SAMConfiguration, is needed to put the PN532+P5CN072 (SMX) in this mode.

Optionally, the PN532 can be put into power down (the wake up sources are configurable. Usually, it will be waken up by an external RF field or by INT0).

Once configured in virtual card mode, the PN532 only acts a bridge between SMX and the external reader.



Depending on the first command, after initialisation, sent by the reader, the PN532+SMX will act as a Mifare card or as a T=CL card.

Table 13. The PN532 +SMX as virtual card

Action	Command ¹	Command explanation	Response	Response explanation
Set the PN532 in virtual card mode	14 02 00	Command code: SAMConfiguration Virtual card mode No timeout	15	Response command code

The PN532 is configured in virtual card mode. SMX is seen by an external reader as a contactless card.

⁽¹⁾Command code and command parameters. Mandatory protocol encapsulation is not represented.

If handshake mode is used, the host controller will be informed by IRQ pin when a transaction occurred between SMX and an external reader. The host controller shall then send a GetGeneralStatus command, to get information about what happened.

It can then use wired card mode to communicate with SMX to check the result of the transaction (for example, which application has been accessed).

3.3.9.2 Wired card mode

In wired card mode, the host controller can access the SMX. Typically, after a transaction occurred between SMX and an external reader, the PN532 access SMX to check what happened.

SMX can communicate either in Mifare or in ISO/IEC 14443-4 protocol.

The PN532 used as reader sends automatically RATS if T=CL support is indicated in SEL_RES of the card (bit 5). Consequently, to communicate with SMX using in Mifare protocol, automatic sending of RATS by the PN532 must be disable, as shown in Table 15 on page 62.

Table 14. The PN532 +SMX as wired ISO/IEC 1443-4 card

Action	Command ¹	Command explanation	Response	Response explanation
Set the PN532 in wired card mode	14 03	Command code: SAMConfiguration Wired card mode	15	Response command code
The PN532 is configured in wired card mode. SMX is accessed by the PN532 as a contactless card.				
Initialize the SMX	4A 01 00	Command code: InListPassivTargets Number of cards to initialize = 1 Baud rate = 106 kbits/sec.	4B 01 01 04 07 28 04 00 D7 1E 92 0D 77 80 81 02 00 73 C8 40 13 00 90 00	Response command code 1 target detected Target number 1 SENS_RES SEL_RES NFCID1 length NFCID1 ATS (13 bytes)
The PN532 communicates with the SMX as with a card. If SMX indicates T=CL compliance, the PN532 automatically sends RATS command.				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

Table 15. The PN532 +SMX as wired Mifare card

Action	Command ¹	Command explanation	Response	Response explanation
Disable automatic sending of RATS command	12 24	Command code: SetParameters Automatic ATR_RES = 1 Automatic ISO/IEC 1443-4A card emu = 1	13	Response command code
The PN532 is configured in wired card mode. SMX is accessed by the PN532 as a contactless card.				
Set the PN532 in wired card mode	14 03	Command code: SAMConfiguration Wired card mode	15	Response command code
The PN532 is configured in wired card mode. SMX is accessed by the PN532 as a contactless card.				
Initialize the SMX	4A 01 00	Command code: InListPassivTargets Number of cards to initialize = 1 Baud rate = 106 kbits/sec.	4B 01 01 04 07 28 04 00 D7 1E 92	Response command code 1 target detected Target number 1 SENS_RES SEL_RES NFCID1 length NFCID1
As automatic sending of RATS was disabled, it has not been sent by the PN532. If SMX supports Mifare emulation, it is now ready to answer in this mode				
Read some data in the SMX memory	40 01 30 00	Command code: InDataExchange Target number 1 Mifare Read16bytes command (Address 00)	41 00 00 D7 1E 92 5B 28 04 07 00 00 00 00 41 07 11 00	Response command code Status = OK 16 bytes read
The PN532 communicates with the SMX using Mifare commands.				

⁽¹⁾ Command code and command parameters. Mandatory protocol encapsulation is not represented.

3.3.9.3 Dual card mode

In this mode, both the PN532 (as a ISO/IEC 18092 passive 106kbps target) and P5CN072 (ISO/IEC 14443-4A card at 106 kbps) will be visible from an external reader.

2 commands are needed:

- SAMConfiguration
- TgInitAsTarget

Table 16. The PN532 +SMX as wired ISO/IEC 1443-4 card

Action	Command ¹	Command explanation	Response	Response explanation
Set the PN532 in Dual card mode	14 04	Command code: SAMConfiguration Dual card mode	15	Response command code
The PN532 is configured in dual card mode.				
Configure the PN532 as a target	8C 00 08 00 12 34 56 40 01 FE a2 a3 a4 a5 a6 a7 c0 c1 c2 c3 c4 c5 c6 c7 FF FF AA 99 88 77 66 55 44 33 22 11 00 00	Command code: TgInitAsTarget Accepted modes : 0 = all SENS_RES NFCID1 SEL_RES Parameters to build POL_RES (16 bytes) NFCID3t (10 bytes) Length of Gt and Tk	8D 22 11 D4 00 01 FE A2 A3 A4 A5 A6 A7 00 00 00 00 00	Response command code Mode : passive mode, 424 kbits/s Initiator command received (ATR_REQ)

SMX (P5CN072) and the PN532 can respond to a reader only after TgInitAsTarget command has been sent.

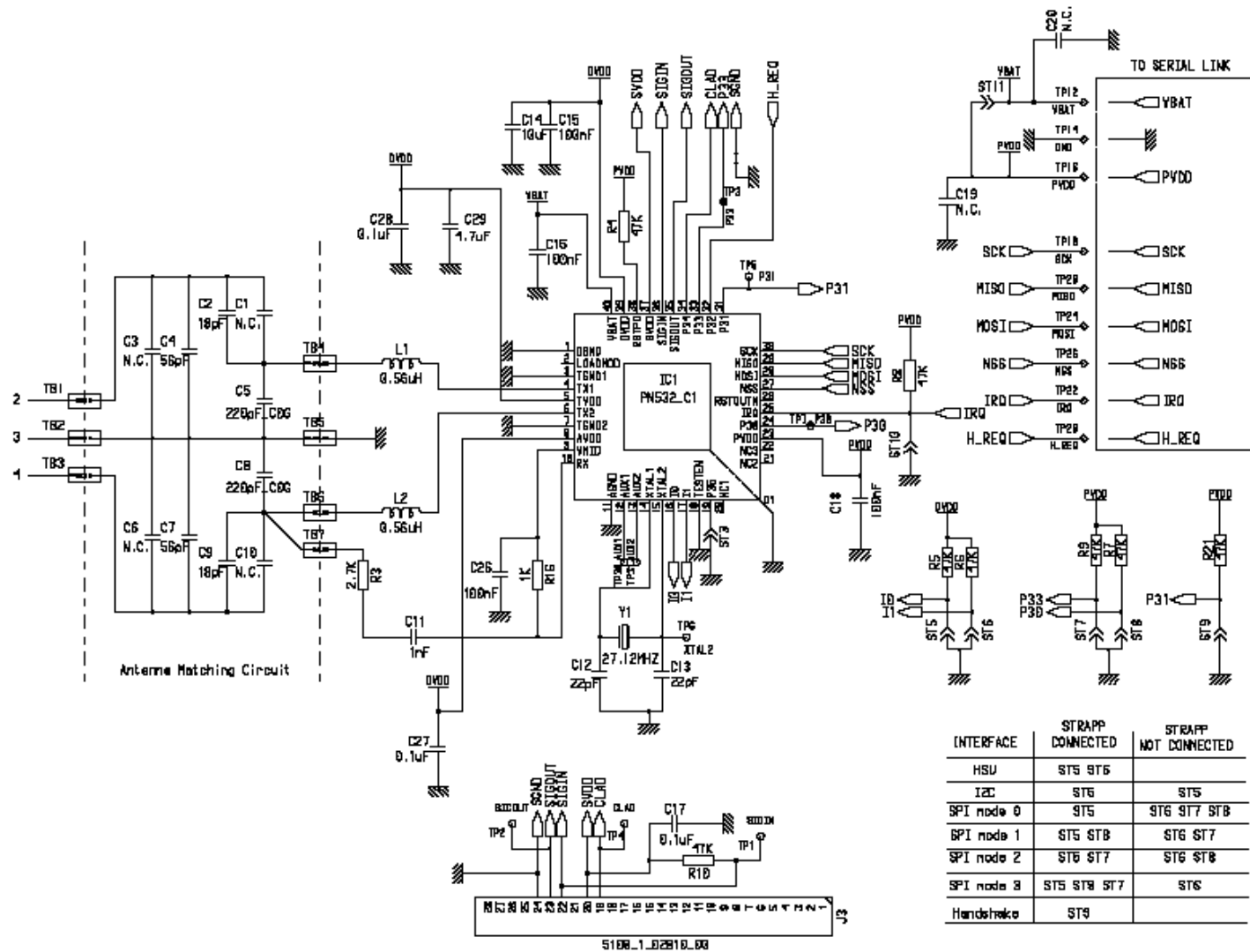
4. Application schematics

Recommendations

The interface is configurable by I0 and I1 pins.

Recommended capacitors:

- On TVDD: 4.7 μ F X7R ceramic in parallel with 100 nF;
- On DVDD: 4.7 μ F (for serial, I2C, SPI) in parallel with 100 nF;
- On AVDD: 100 nF.
- Antenna: use COG capacitors. Concerning the antenna design and tuning, please refer to the application note "NFC Transmission Module Antenna and RF Design Guide".



INTERFACE	STRAPP CONNECTED	STRAPP NOT CONNECTED
HSU	ST5 ST6	
I2C	ST6	ST5
SPI mode 0	ST5	ST6 ST7 ST8
SPI mode 1	ST5 ST8	ST6 ST7
SPI mode 2	ST6 ST7	ST6 ST8
SPI mode 3	ST5 ST8 ST7	ST6
Handshake	ST9	

Fig 23. Serial application schematic – sheet 1/3

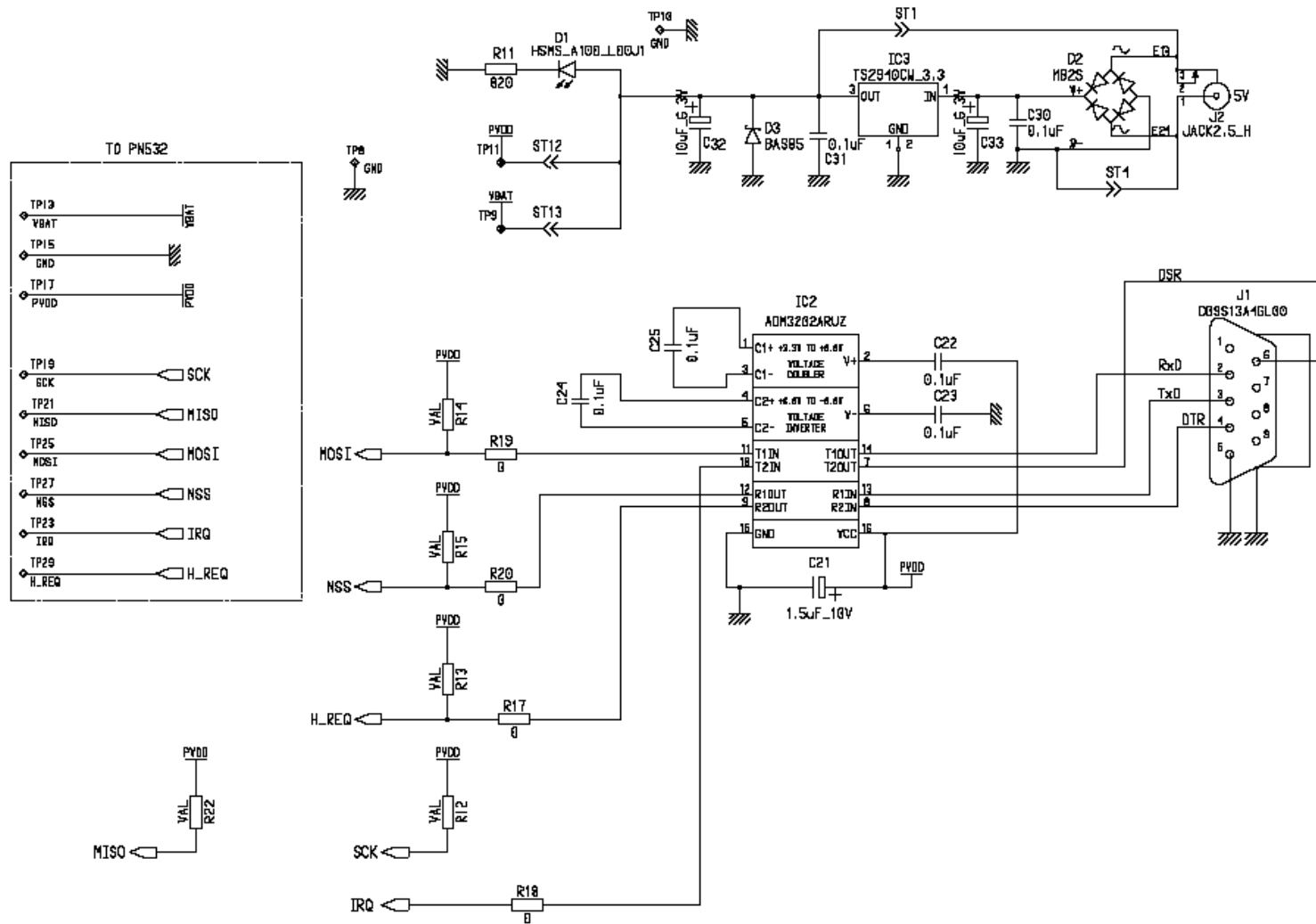


Fig 24. Serial application schematic – sheet 2/3

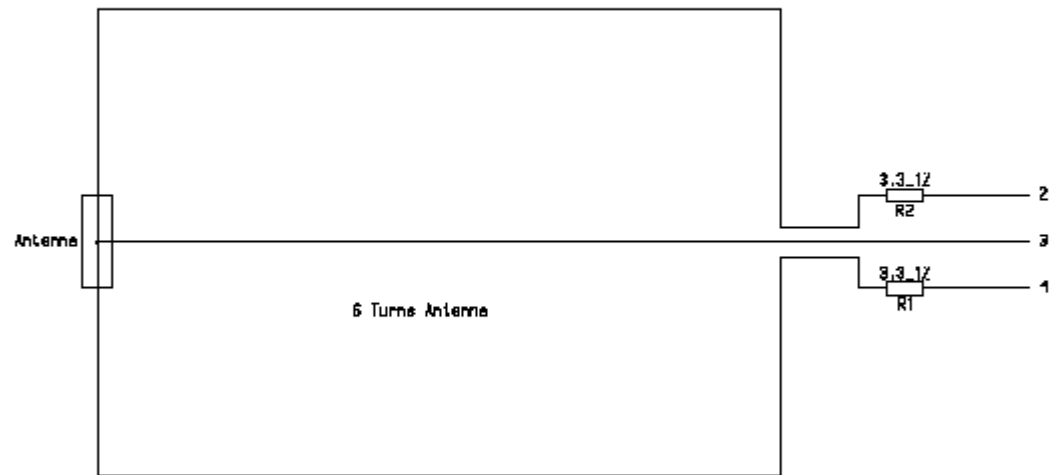


Fig 25. Serial application schematic – sheet 3/3

5. Recommendations/ known limitations

- **Metachaining in case of bad RF condition (RF error handling)**

It is recommended not to use Metachaining functionality without a frame integrity check mechanism implemented at the host side, because the PN532 can lose some bytes, in case RF conditions are bad (this happens only in case of RF communication problems)

DEP Metachaining on the target side:

When the tox-req is not seen over the air by the initiator on the last packet in a metachained frame, the last packet erases the previous one in the response of the command TgGetData.

DEP Metachaining on the initiator side:

The repetition of a frame, in case of non-receiving ACK, does not concatenate the remaining bytes of a previous InDataExchange command

The host controller (of both target and initiator) must implement a frame integrity check mechanism, or shall use chaining mechanism only.

- **Echo Back Test in 106 kbps on the target side:**

The Diagnose command (NumTst = 0x05) is not functional the first time it is launched. Workaround: The host controller shall send the command TgInitAsTarget before launching the Diagnose command (NumTst = 0x05) in 106 kbps

- **ISO/IEC 14443-4A PICC emulation: R(ACK) resent after R(NACK) reception (RF error handling)**

Just after reset, in a chained frame, the R(ACK) is resent when a R(NACK) has been received. In a second chained frame, the R(ACK) (with wrong block number) is resent with some other data (the last TgSetData length) when a R(NACK) has been received.

The host should reset the the PN532 acting as PICC by sending a soft reset (writing 0x01 in the ControlRegister at address 0x6203)

6. Legal information

6.1 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.

6.2 Disclaimers

General — 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.

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 medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a 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 for 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.

6.3 Licenses

Purchase of NXP <xxx> components

<License statement text>

6.4 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>

6.5 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.

7. Contents

1.	Introduction	3	3.3.3.1	Timeout and number of retries	40
2.	Interfaces with the host controller	6	3.3.4	How to use the PN532 as a FeliCa reader?	41
2.1	The PN532 block diagram	6	3.3.4.1	Timeout and number of retries	42
2.2	The PN532 host link protocol	7	3.3.5	How to use the PN532 as a Jewel cards reader ?	43
2.2.1	Standard frame	7	3.3.6	How to use the PN532 as a reader for several	
2.2.2	Extended frame	9		types of cards (or targets)?	44
2.2.3	Additional lines (IRQ only or IRQ and H_REQ)	10	3.3.7	How to use the PN532 in a NFC peer-to-peer	
2.3	Typical application diagram	11		communication?	45
2.4	How to interface with the PN532?	13	3.3.7.1	How to use the PN532 as an initiator in a NFC	
2.4.1	SPI interface	13		peer to peer communication?	45
2.4.1.1	Sequence to use communicate in SPI	13	3.3.7.2	How to use the PN532 as a target in a NFC	
2.4.1.2	SPI mode selection	14		peer-to-peer communication?	49
2.4.2	I2C interface	15	3.3.7.3	Handling of several cards/targets	53
2.4.2.1	Handshake mode in I2C	15	3.3.7.4	Transfer of large amount of data	54
2.4.3	Recommended operating conditions	18	3.3.7.5	Effective throughput	55
2.4.4	HSU interface	19	3.3.8	How to emulate a ISO/IEC 1443-4A card ?	56
2.4.4.1	Baudrate	19	3.3.9	How to use Smart connectivity (combination of	
2.4.4.2	Handshake mode in HSU	19		the PN532 and SMX)?	58
3.	The PN532 commands	24	3.3.9.1	Virtual card mode	59
3.1	Global view	24	3.3.9.2	Wired card mode	61
3.2	The PN532 embedded software API: commands		3.3.9.3	Dual card mode	63
and errors lists	25	4.	Application schematics	64	
3.2.1	Commands codes	25	5.	Recommendations/ known limitations	68
3.2.2	Errors codes	26	6.	Legal information	69
3.3	The PN532 various modes	28	6.1	Definitions	69
3.3.1	How to use the PN532 as a Mifare reader?	31	6.2	Disclaimers	69
3.3.1.1	Timeout and number of retries	35	6.3	Licenses	69
3.3.2	How to use the PN532 as a T=CL reader		6.4	Patents	69
(ISO/IEC 14443-4)?	36	6.5	Trademarks	69	
3.3.2.1	Timeout and number of retries	39	7.	Contents	70
3.3.3	How to use the PN532 as a DESfire cards				
reader?	40				

continued >>