Semiconductors

AN10394_7

PN531 application note

Rev. 1.6 — 2006-11-27

Application note

Document information

Info	Content
Keywords	NFC, ISO/IEC 18092, PN531 C2, PN531 C203 (TAMA v4.2)
Abstract	PN531 C203 Application note. How to use PN531 C203 as a reader, as a card, or in a NFC peer to peer communication

PHILIPS

Revision history

Rev	Date	Description
1.6	2006 Nov 27	Add ISO14443-4 case 1 limitation.
1.5	2006 Oct 10	Add limitations description. AN10394_6
1.4	2006 Apr 4	Correction in Mifare command set tables (A0h and A2h commands)
1.3	2006 Jan 17	Add some recommendations how to use PN531 C203 (v4.2)
1.2	2005 Dec 5	Update for PN531 C203 (v4.2)
1.1	2005 Oct 20	Corrections. - Changed “39.3 seconds” into “39.3 milliseconds” on page 37 - Changed “P5CT072” to “P5CN072” on page 39.
1.0	2005 Sep 2	Draft. From AN10379 of PN531 C1. - Update according to changes between PN531C1 and PN531C2 - Add precisions about SPI (2.4) and I2C interfaces (2.5) - Add example about DESfire cards reading (3.2.3) - Add explanation about use of IRQ OR handshake (IRQ+HREQ) (2.2.1 and 2.2.2) - Add paragraph about data throughput (3.2.5.5)

Contact information

For additional information, please visit: http://www.semiconductors.philips.com

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

1. Introduction

PN531 C203 (also named TAMA v4.2) is a highly integrated transmission module for contactless communication at 13.56 MHz including microcontroller functionality based on a 80C51 core with 32 kbytes of ROM and 1 kbyte of RAM.

TAMA 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.

TAMA C203 includes a switch to power an external SAM connected to S2C interface. It is controlled by the embedded firmware.

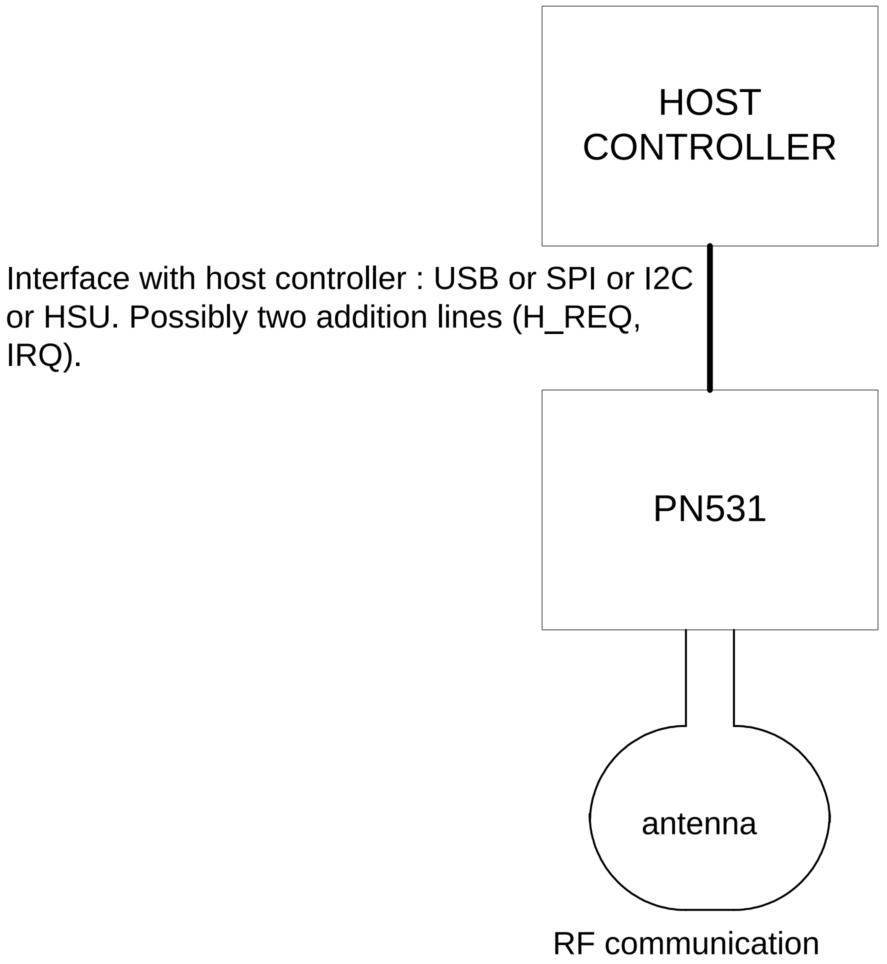


Fig 1. Simplified system view

This document intends to allow the customer getting quickly started with PN531 C203 (v4.2). It summarizes commands needed to use PN531 C203 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 PN531 firmware can be found in PN531 User manual (cf. References table on page 4).

Full description of PN531 IC can be found in PN531 Datasheet.

References

Document name	*Please refer to:*
PN531 v4.2 user manual (UM0501-02)	111802.pdf
PN531 C2 Datasheet	111903.pdf
NFC Transmission Module Antenna and RF Design Guide	100720.pdf
DESfire cards specification	M075031.pdf
Mifare specification	www.semiconductors.philips.com/markets/identification/datasheets/
ISO14443 specification (T=CL)	ISO14443-3 specification ISO14443-4 specification
NFCIP-1 specification	ISO/IEC 18092 specification

Glossary

NFC	Near Field Communication
TAMA	PN531
HSU	High Speed UART
SMX	Philips SmartMX (Memory Extension)

2. How to interface with PN531 C2?

2.1 PN531 C203 block diagram

PN531 is based on a 8051 core, with 32 kbytes ROM and 1kbyte RAM. The chip contents a contactless UART (with S^2C interface SigIn, SigOut and P34), and a contactless front end. PCR block controls clocks and power. The chip can be powered by USB (typ. 5V) or by VDD (typ. 3.3V).

The 4 MHz oscillator is only needed in case of USB interface.

In PN531 C2, SVDD switch is included, and **SVDD is an output.**

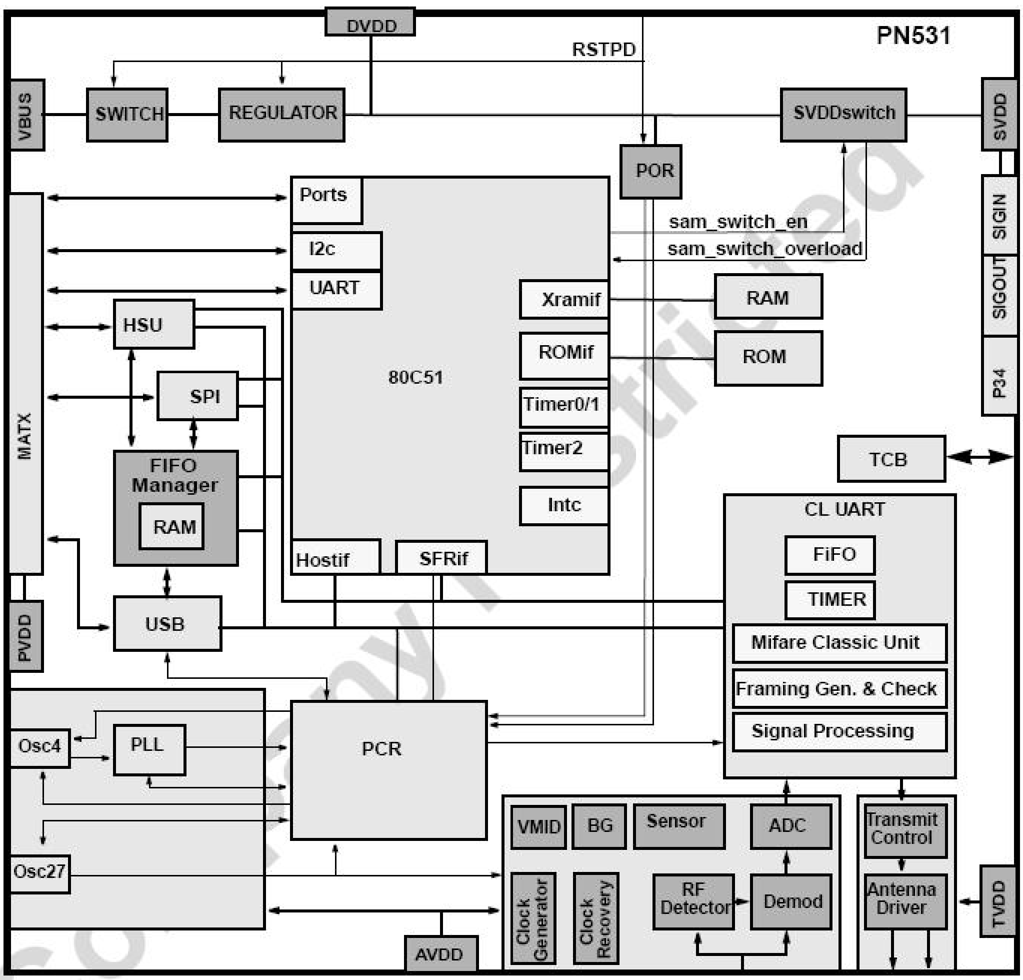


Fig 2. PN531 block diagram

Four interfaces are available : USB, 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	0
I2C	1	0
SPI	0	1
USB	1	1

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

2.2 PN531 host link protocol

The protocol used on host link is fully described in PN531 User manual (cf. References table on page 4).

A basic exchange consists in a command frame sent by the host controller to PN531, an ACK frame sent by PN531 as soon as the command is correctly received, and a response frame, read by the host controller (polling mechanism or use of IRQ). Full explanation of each interface can be found in PN531 User manual (cf. References table on page 4).

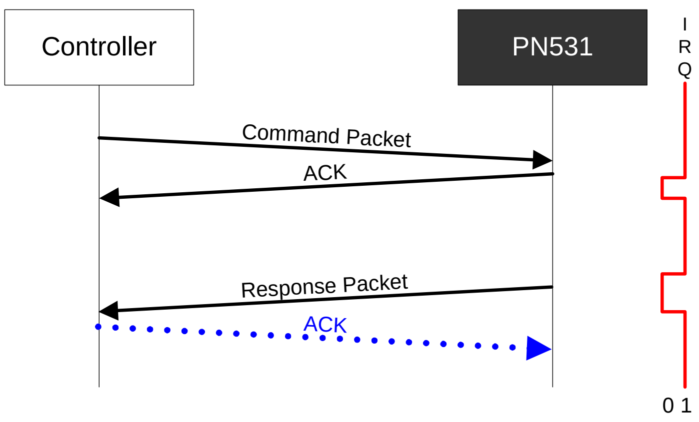


Fig 3. Normal exchange between host controller and PN531

Command and response frame structure is shown in figure 3.

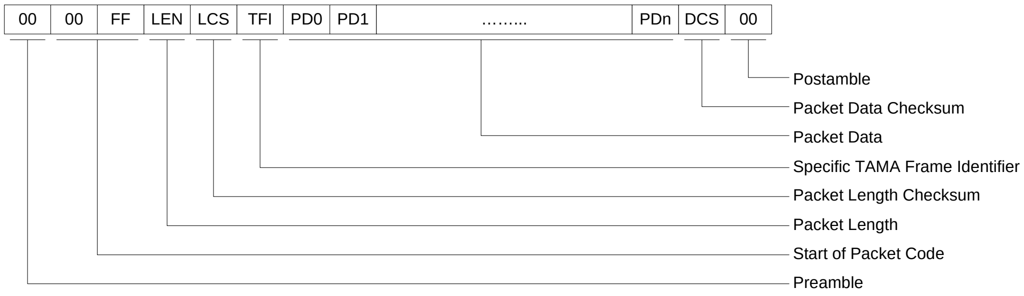


Fig 4. Information frame

- **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 TAMA 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 PN531
- **D5h** in case of a frame from the PN531 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.

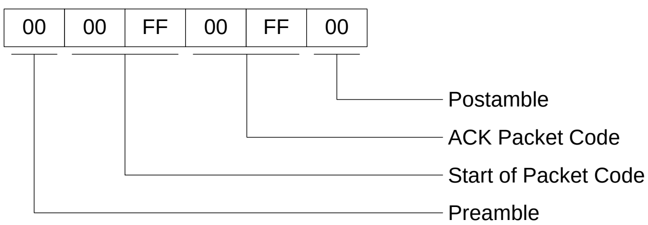


Fig 5. ACK frame

List of available commands (PD0 byte) is provided in section 3.1.

2.2.1 Use of IRQ

In addition to the lines needed for the interface, one IRQ line can be used. IRQ indicates the host controller when response data are ready.

To enable IRQ line, SetTAMAParameter command (code 12) is used with parameter 1Ch (bit 3 is fIRQused, other bits are default value (described in User manual (cf. References table on page 4))).

The IRQ behaviour is depicted on Fig 3. IRQ is low when ACK frame is ready or when response frame is ready.

2.2.2 Use of handshake mode

"Handshake mode" is only available with HSU and I2C links. In this mode, 2 more lines (in addition to RX/TX or SDA/SCL) are connected between the host controller and PN531: H_REQ (P32) and IRQ.

P31 is used to select handshake or standard mode:

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

This mode allows:

- the controller to wake up PN531 before sending a command
- PN531 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 SetTAMAParameter to enable IRQ**

This mode is explained in PN531 User manual (cf. References table on page 4) and in following paragraphs I2C interface and HSU interface.

In the rest 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 USB interface

PN531 USB interface is compatible with USB 2.0 specification and is running at 12 MB/s (full speed device). Control endpoint is 8 bytes and Bulk in/out endpoint is 64 bytes long.

2.3.1 CPU frequency in USB mode

PN531 C203 (v4.2) firmware set CPU frequency at 27.12 MHz.

2.3.2 Recommended operating conditions

It is recommended to use the default CPU frequency.

2.4 SPI interface

PN531 is slave. Functional operation is guaranteed up to 1.2 MHz. A Status byte (Bit 0 of Status byte) indicates if PN531 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 tama)

xxxx xx11 : Read data (transmission from tama 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 PN531.

Bytes are transmitted LSB first.

2.4.1 Sequence to use communicate in SPI

```
set NSS low

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

// Wait for ACK of TAMA
// ================
write "read status" ie xxxx xx10
read one byte. Response of TAMA 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 TAMA
// ==================
write "read status" ie xxxx xx10
read one byte. Response of TAMA 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 PN531 and another device.

2.4.2 CPU frequency in SPI mode

CPU frequency during RF communication is 27.12 MHz.

CPU frequency during host communication is 27.12 MHz.

CPU frequency in other case is 6.78 MHz.

2.4.3 Recommended operating conditions

It is recommended to use the default CPU frequency.

2.4.4 SPI mode selection

In PN531 C203 (v4.2), CKPOL and CKPHA are configurable with P3.0 and P3.1

SPI mode		P30	P31
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

2.5 I2C interface

I2C clock speed is up to 400 kHz.

PN531 is a slave. Its address is 48h.

In case PN531 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 PN531 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.

2.5.1 CPU frequency in I2C mode

CPU frequency during RF communication is 27.12 MHz.

CPU frequency during host communication is 27.12 MHz.

CPU frequency in other case is 6.78 MHz.

It is recommended not to change the CPU frequency.

2.5.2 Handshake mode in I2C

4 lines are connected between the host controller and PN531: SCL, SDA, H_REQ and IRQ.

Remark: in handshake mode, IRQ line is not set low by PN531 when ACK frame is ready (different behaviour than standard mode and use of IRQ).

2.5.2.1 PN531 is not in power down, the host controller initiates an exchange.

H_REQ line is not needed. IRQ line is asserted by PN531 when the response to the command is ready.

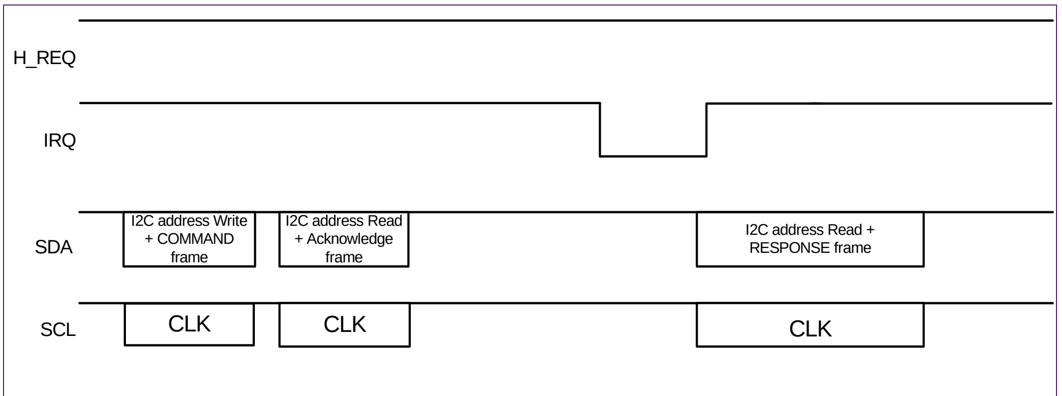
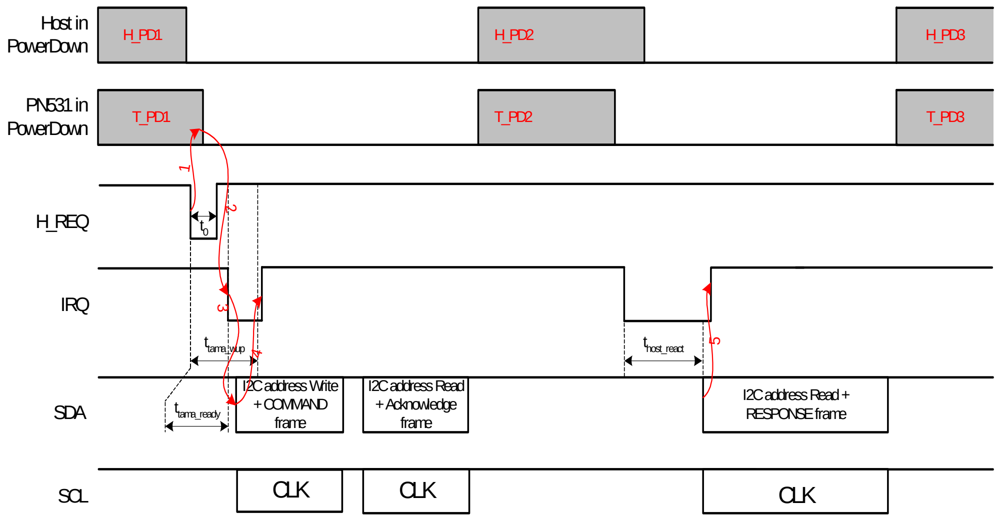


Fig 6. I2C exchange in handshake mode, case 1

2.5.2.2 PN531 may be in power down, the host controller initiates an exchange.

I

f PN531 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. PN531 will also generate set IRQ low to indicate that the response is ready.



t_{TAMA_READY} if PN531 awake < 10µs
t_{TAMA_READY} if PN531 asleep < 170µs
t_0 if PN531 awake > 1µs
t_0 if PN531 asleep >175µs

Fig 7. I2C exchange in handshake mode, case 2

2.5.2.3 PN531 is in power down, after TgInitTAMATarget command.

After TgInitTAMATarget command, PN531 automatically goes into power down (if there is no external field). It saves power until an initiator is detected. The initiator wakes up PN531 and PN531 is configured as a target. PN531 will assert IRQ line to indicate to the host controller that response to TgInitTAMATarget is ready.

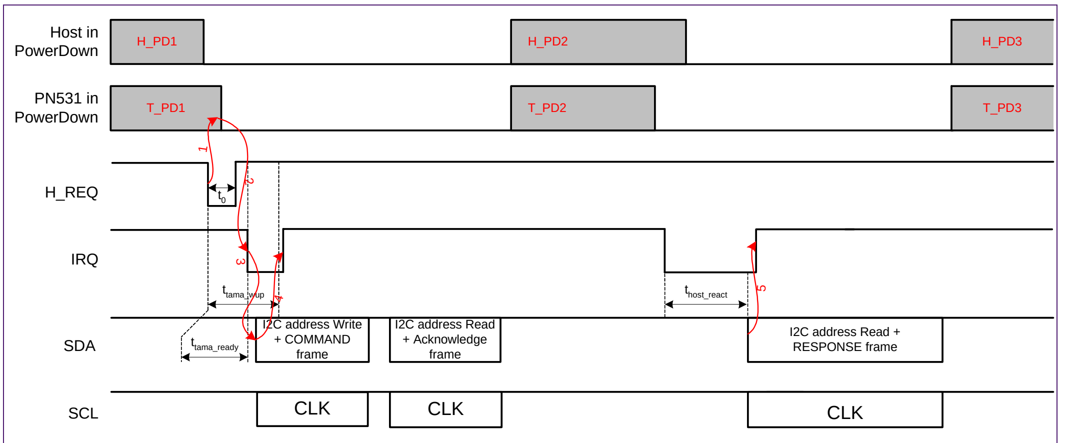


Fig 8. I2C exchange for TgInitTAMATarget command

2.5.2.4 Virtual card mode.

PN531 informs the host controller that something happened.

The host controller is possibly in power down. PN531 is woken up by an external event, for example RF detection in virtual card mode. PN531 is automatically switched to standard mode. IRQ is asserted by PN531 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)).

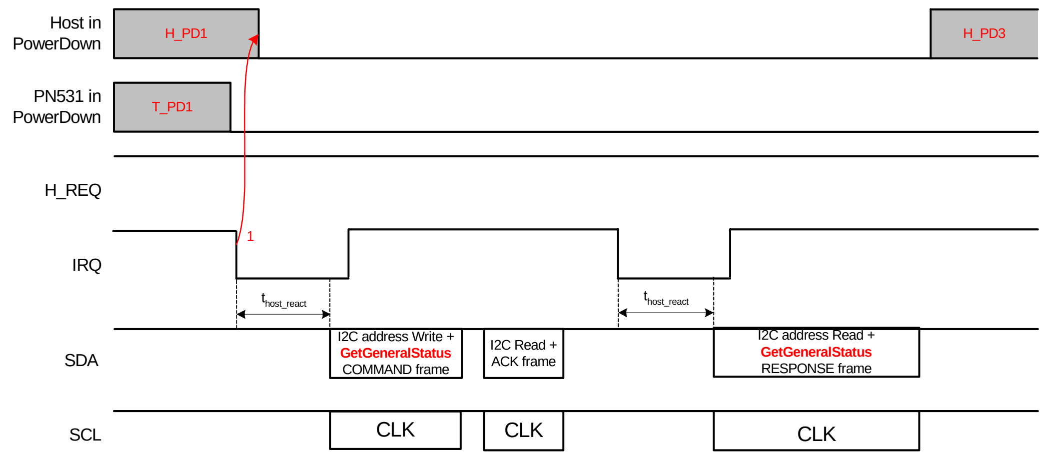


Fig 9. I2C exchange after virtual card mode

2.5.3 Recommended operating conditions

It is recommended to use the handshake mode, described in paragraph 2.5.2, 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.

It is recommended to use the default CPU frequency.

2.6 High speed UART interface

2.6.1 CPU frequency in HSU mode

CPU frequency during RF communication is 27.12 MHz.

CPU frequency during host communication is 27.12 MHz.

CPU frequency in other case is 6.78 MHz.

2.6.2 Recommended operating conditions

It is recommended to use the default CPU frequency.

2.6.3 Baudrate

Default baudrate is 9600 bauds.

PN531 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. PN531 will switch to the new baudrate only after reception of this ACK frame.

2.6.3.1 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
```

2.6.4 Handshake mode in HSU

4 lines are connected between the host controller and PN531: RX, TX, H_REQ and IRQ.

Remark: in handshake mode, IRQ line is **not** set low by PN531 when ACK frame is ready (different behaviour than standard mode + use of IRQ).

2.6.4.1 PN531 is not in power down, the host controller initiates an exchange.

No handshake mechanism is needed.

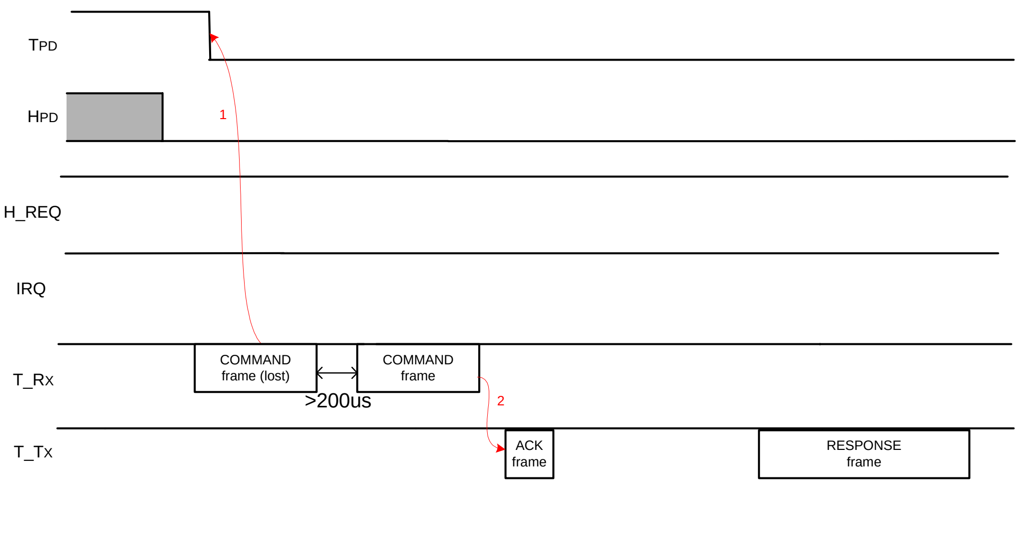


Fig 10. HSU exchange in handshake mode, case 1

2.6.4.2 PN531 may be in power down, the host controller initiates an exchange.

If PN531 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.

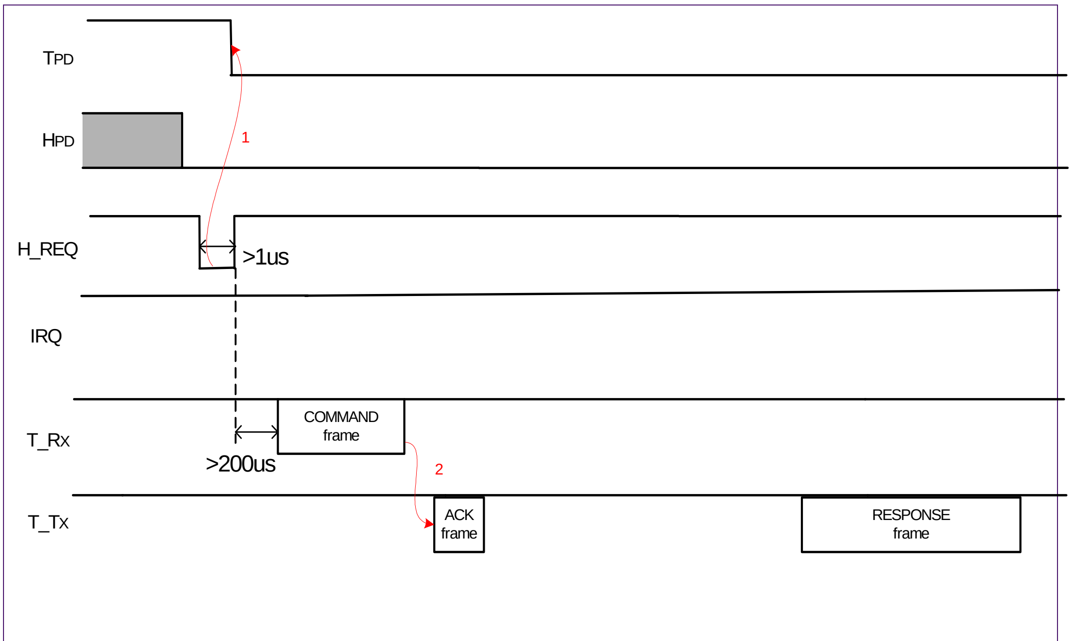


Fig 11. HSU exchange in handshake mode, case 2

2.6.4.3 PN531 is in power down, after TgInitTAMATarget command.

After TgInitTAMATarget command, PN531 automatically goes into power down (if there is no external field). It saves power until an initiator is detected. The initiator wakes up PN531 and PN531 is configured as a target. PN531 will assert IRQ line to indicate to the host controller that response to TgInitTAMATarget is ready. The host acknowledges with a pulse on H_REQ line.

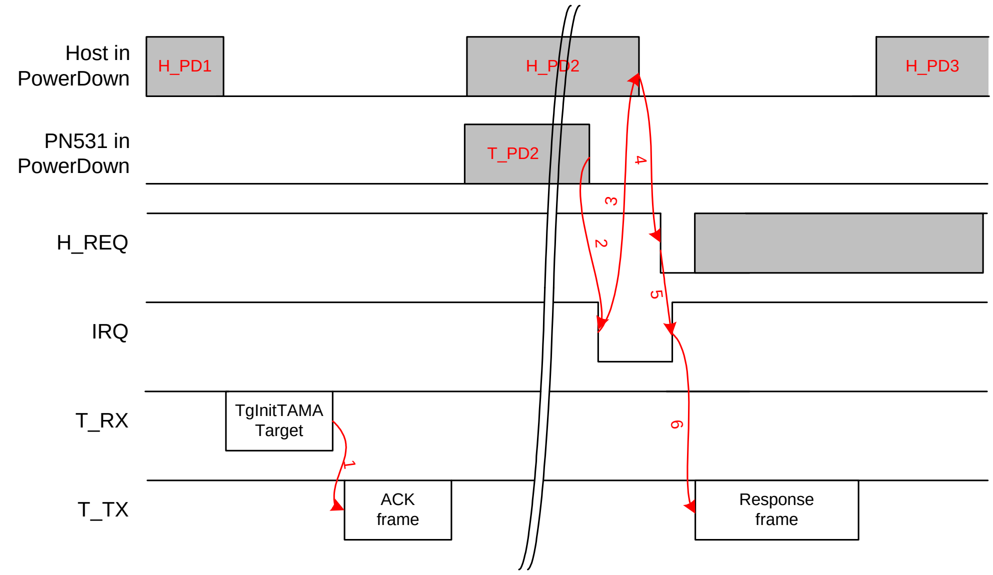


Fig 12. I2C exchange for TgInitTAMATarget command

2.6.4.4 Virtual card mode.

PN531 informs the host controller that something happened.

The host controller is possibly in power down. PN531 is woken up by an external event, for example RF detection in virtual card mode. PN531 is automatically switched to standard mode. IRQ is asserted by PN531 to wake up the host controller. The host controller acknowledges with a falling edge on H_REQ, and shall send a GetGeneralStatus command to get information about what happened (see GetGeneralStatus command description in User manual (cf. References table on page 4)).

Warning:

Bit 3 of SAMStatus, in GetGeneralStatus response, must not be taken into account. Information about RF field is indicated in byte 3 (Field) of GetGeneralStatus response.

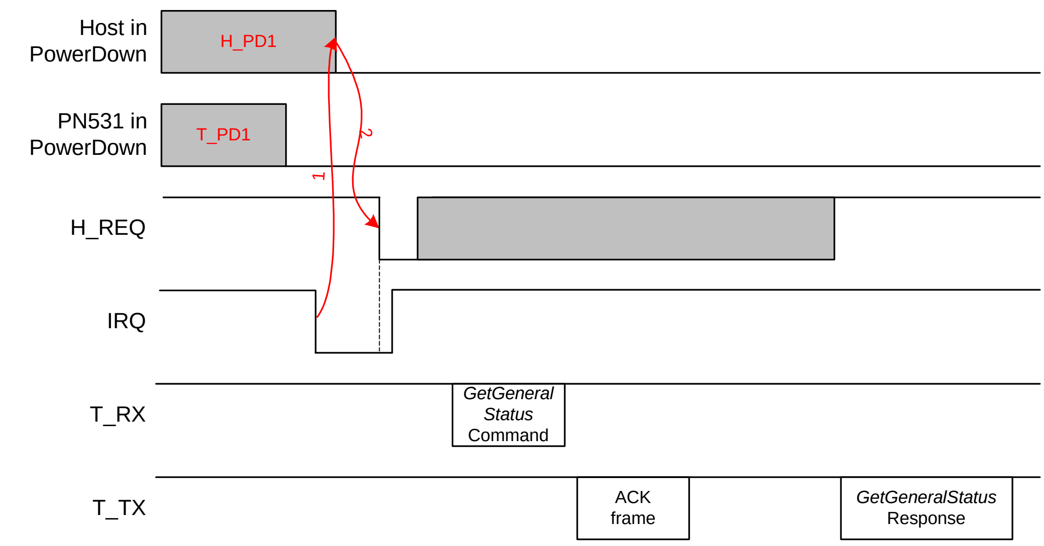


Fig 13. I2C exchange after virtual card mode

3. How to use PN531?

3.1 PN531 embedded software API: commands and errors lists

PN531 is interfaced with the host controller via USB, SPI, I2C or HSU (see section 2). Table 1: lists the command codes of all commands. Those commands allow an easy set up of a RF communication between PN531 and another device, as explained in paragraph 3.2.

Table 1: Table of commands

Command name	Command code (PD0)	Command relevant in target mode	Command relevant in initiator mode
Diagnose	00h	yes	yes
GetFirmwareVersion	02h	yes	yes
GetGeneralStatus	04h	yes	yes
ReadRegister	06h	yes	yes
WriteRegister	08h	yes	yes
ReadGPIO	0Ch	yes	yes
WriteGPIO	0Eh	yes	yes
SetSerialBaudRate	10h	yes	yes
SetTAMAParameters	12h	yes	yes
SAMConfiguration	14h	yes	yes
PowerDown	16h	yes	yes
RFConfiguration	32h	yes	yes
RFRegulationTest	58h	n.a.	n.a.
InJumpForDEP	56h	no	yes
InJumpForPSL	46h	no	yes
InListPassiveTarget	4Ah	no	yes
InATR	50h	no	yes
InPSL	4Eh	no	yes
InDataExchange	40h	no	yes
InCommunicateThru	42h	no	yes
InDeselect	44h	no	yes
InRelease	52h	no	yes
InSelect	54h	no	yes
TgInitTAMATarget	8Ch	yes	no
TgSetGeneralBytes	92h	yes	no
TgGetDEPData	86h	yes	no
TgSetDEPData	8Eh	yes	no
TgSetMetaDEPData	94h	yes	no
TgGetInitiatorCommand	88h	yes	no
TgResponseToInitiator	90h	yes	no
TgGetTargetStatus	8Ah	yes	no

Some commands return a status byte. Status byte signification is shown in next table.

Table 2: Table of errors

Error cause	Error code (status)
No error	**00h**
Time Out (the target has not answered)	01h
A CRC error has been detected by the contactless UART	02h
A Parity error has been detected by the contactless UART	03h
An erroneous Bit Count has been detected during Mifare® Anti-collision/select operation	04h
Framing error during Mifare® operation	05h
Abnormal bit Collision detected during bit wise anti-collision at 106 kbps	06h
Ccommunication Buffer size too small	07h
RF Buffer overflow has been detected by the contactless UART	09h
In active mode, the RF field has not been switched on in time by the counterpart (as defined in NFCIP-1)	0Ah
RF Protocol error (cf. CL_ERROR register description in PN531 specification)	0Bh
Temperature error: the internal temperature sensor has detected overheating, and therefore has automatically switched off the antenna drivers	0Dh
Internal buffer overflow	0Eh
Invalid parameter (range, format, …)	10h
DEP Protocol: PN531 configured in target mode does not support the command received from the initiator (ATR_REQ, WUP_REQ, PSL_REQ, DEP_REQ, DSL_REQ, RLS_REQ)	12h
DEP Protocol / Mifare® / ISO14443-4: the data format does not match the specification. Depending on the RF protocol used, it can be: - Bad length of RF received frame - Icorrect value of PCB or PFB - Invalid or unexpected received frame - NAD or DID incoherence	13h
Mifare®: Authentication error	14h
ISO14443-3: UID Check byte is wrong	23h
DEP Protocol: Invalid device state, the system is in a state which does not allow the operation	25h
Operation not allowed in this configuration (host interface)	26h
This command is not acceptable due to the current context of PN531 (Initiator vs. Target, unknown target number, Target not in the good state, etc)	27h
PN531 (target) has been released by its initiator	29h

3.2 PN531 various modes

This paragraph summarizes PN531 functionalities and shows which commands are associated to them.

PN531 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.

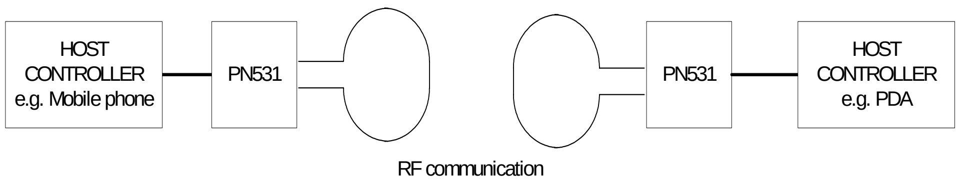


Fig 14. Peer to peer communication (NFC initiator and NFC target)

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

In this mode, RF communication is according to Mifare specification. PN531 behaves as a Mifare reader. It can communicate with Mifare cards.

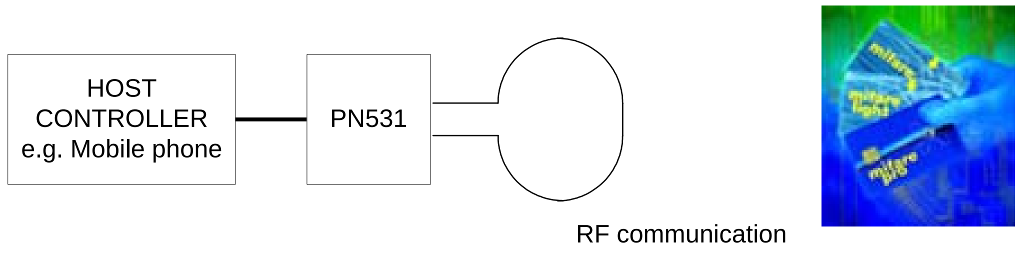


Fig 15. PN531 as a Mifare reader

- **As a T=CL reader (ISO14443-4 protocol)**

In this mode, RF communication is according to ISO14443-4 specification. PN531 behaves as a ISO14443-4 reader. It can communicate with ISO14443-4 cards.

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

In this mode, RF communication is according to FeliCa specification.

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

In this mode, PN531 is combined with a secure smart card. An external reader sees the set PN531+secure smart card as a contactless card.

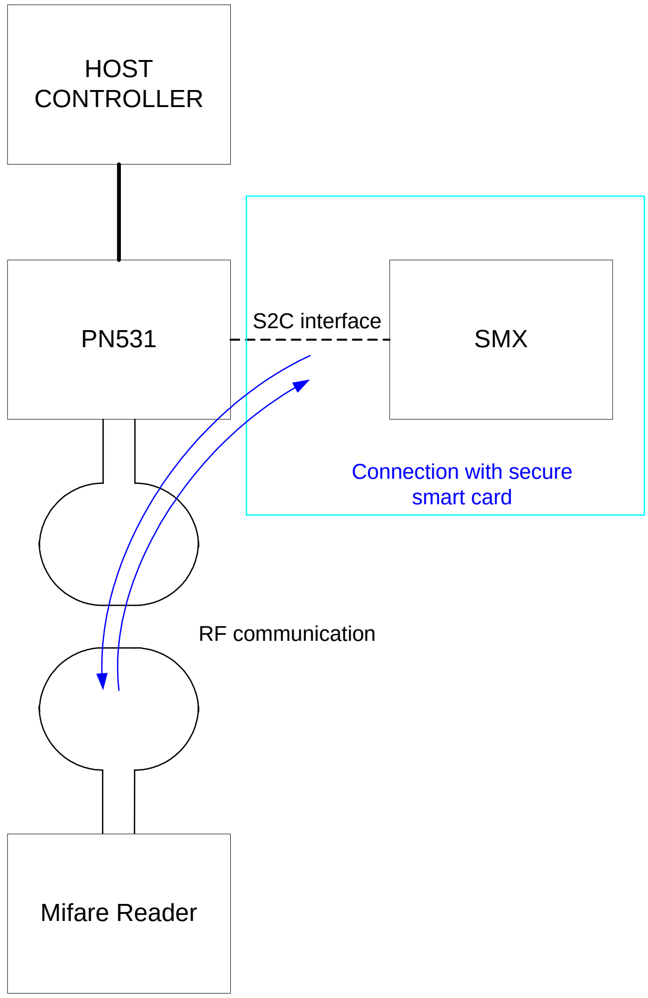


Fig 16. Virtual card mode

Very few commands are needed to set up RF communication between PN531 and another device (reader, card, or other NFC device). PN531 executes **different RF processes**, depending on the type of communication, **but from the host controller to PN531, same commands are used** (whatever the baudrate, the mode etc):

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

3.2.1 How to use PN531 as a Mifare® reader?

Goal: read/write Mifare cards.

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 3.2.5.3 on page 47).

Warning:

In case the card initialized indicates it supports ISO14443-4 protocol (bit 5 of SAK, cf. ISO14443-3 specification), InListPassiveTarget command of PN531 performs automatically ISO14443-4 activation (i.e. RATS sending). To disable automatic RATS sending, SetTAMAParameter command must be used (cf. UM0501-02 page 67 paragraph 4.2.9).

Table 3: SetTAMAParameter command usage to enable or disable automatic RATS sending (ISO14443-4 mode)

Action	Command[1]	Command explanation	Response	Response explanation
Disable automatic sending of RATS command	12 04	Command code: **SetTAMAParameters** Automatic ATR_RES = 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 PN531 at power up)	12 0C	Command code: **SetTAMAParameters** Automatic ATR_RES = 1 Automatic RATS = 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 below shows how to use some of those commands (how to fill the parameters, which bytes are returned in response…).

Table 4: PN531 as a Mifare® reader

Action	Command[1]	Command explanation	Response	Response explanation
Scan for 2 targets in the field and initialize them	4A 02 00	Command code: **InListPassivTargets** Number of cards to initialize (if present in the field) Baud rate = 106 kbits/sec.	4B 02 01 04 00 08 04 12 67 58 32 02 44 00 00 08 88 04 B6 E4 00 00 00 00	Response command code 2 targets detected Target number 1 SENS_RES[2] of target 1 SEL_RES[2] of target 1 NFCID1 length = 4 bytes NFCID1 of target 1 Target number 2 SENS_RES[2] of target 2 SEL_RES[2] of target 2 NFCID1 length = 8 bytes NFCID1 of target 2
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 01 60 07 FF FF FF FF FF FF 12 67 58 32	Command code: **InDataExchange** The cmd is intended to target number 1 Mifare® cmd code (authenticate) Mifare® address Default authentication key (last bytes are NFCID1 bytes)	41 00	Response command code Status = 0 (OK, no error)
Card2 has been put in halt state. Card1 has been wake up and authentication has been performed.				
Read out card 1 memory	40 01 30 04	Command code: **InDataExchange** The cmd is intended to target number 1 Mifare® cmd code (read 16 bytes) Mifare® address	41 00 EE EE EE EE EE EE EE EE EE EE EE EE EE EE EE EE	Response command code Status = 0 (OK, no error) Bytes read (example)
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 02 A2 08 FF FF FF FF	Command code: **InDataExchange** The cmd is intended to target number 2 Mifare® cmd code (write 4 bytes) Mifare® address Bytes to be written	41 00	Response command code Status = 0 (OK, no error)
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 01	Command code: **InDeselect** The cmd is intended to target number 1	45 00	Response command code 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. Please refer to this document for further details.

Instead of 2 cards, as shown in the previous example, it is also possible to detect a Mifare card and a NFC target. Mifare card will be associated with the logical number 01 for example, and NFC target with the logical number 02. Then data are received and sent to/ from the card or the target with InDataExchange function followed by the logical number 01 or 02 (in the same way than with two Mifare cards). But it is not possible to switch several times from the NFC target to the Mifare card. The second time the host controller will try to select the NFC target, it will fail.

Workaround: the host controller of the target will receive an error (error code 25h) in response to GetDEPData command. It shall then send a new TgInitTAMATarget command, followed by TgGetDEPData command.

At 106 kbps, InListPassiveTarget command allows to look for cards in field but also to activate a known card in the field with its UID or part of its UID. At least 4 bytes of UID must be filled in the parameter field in order to be able to activate the known card.

3.2.1.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, PN531 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). Cf. UM0501-02 page 82.

If there is no card in the field, a timeout occurs after 5 ms. Either PN531 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, PN531 automatically wakes up the card corresponding to the desired TgNb, and automatically put in HALT state the other one.

3.2.2 How to use PN531 as a T=CL reader (ISO14443-4)?

Goal: read/write T=CL cards.

Typical sequence (example):

- Scan for targets (cards) in the field, and send RATS_REQ. CID parameter is set to 0 and FSDI is set to 5 (➔ FSD = 64 bytes).
- 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)
- **InDataExchange**, to send ISO14443-4 commands
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card (this is optional, see paragraph 3.2.5.3 on page 47).

For detailed explanation about each command, please refer to PN531 User manual (cf. References table on page 4), ISO14443-3 and ISO14443-4 specification.

Warning
PN531 can handle incorrectly an ISO7816-4 Case 1 APDU command in ISO14443-4 R/W mode.

If a host command is sent to PN531 containing an ISO7816-4 Case 1 APDU command (CLA INS P1 P2) **and** DCS byte is greater than value 0xFA, an error 0x07 will be returned by PN531

Table 5: PN531 as a T=CL reader

Action	Command[4]	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A 01 00	Command code: **InListPassivTargets** Number of cards to initialize (if present in the field) = 1 Baud rate = 106 kbits/sec.	4B 01 01 04 07 28 04 00 D4 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[(1)] of target 1 SEL_RES[(1)] of target 1 NFCID1 length = 4 bytes NFCID1 of target 1 ATS
Bit 5 of SEL_RES indicates the target is ISO14443-4 compliant. In that case PN531 automatically sends the RATS[(2)]. ATS is indicated in the response.				
Read 16 bytes	40 01 00 B0 81 00 10	Command code: **InDataExchange** The cmd is intended to target number 1 ISO14443-4 command	41 00 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF 90 00	Response command code Status = 0 (OK, no error) Response of the card
ISO14443-4 commands, for example Read 16 bytes command, can be sent with InDataExchange command[(3)].				
Deselect the card	44 01	Command code: **InDeselect** The cmd is intended to target number 1	45 00	Response command code 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 SetPN531Parameters command. Please refer to PN531 User manual (cf. References table on page 4).

[(3)] PN531 handles chaining, time out extension, and error handling, according to ISO14443-4 protocol.

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

3.2.2.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, PN531 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). Cf. UM0501-02 page 82.

If there is no card in the field, a timeout occurs after 5 ms. Either PN531 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 ISO14443-3 and -4. The waiting time extension mechanism is fully embedded inside PN531. The error handling and the chaining are also fully managed by PN531.

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, PN531 automatically wakes up the card corresponding to the desired TgNb, and automatically put in HALT state the other one.

3.2.3 How to use PN531 as a DESfire cards reader?

DESfire cards can be read/write as T=CL cards.

Table 6: PN531 as a DESfire card reader

Action	Command[4]	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A 01 00	Command code: **InListPassivTargets** Number of cards to initialize (if present in the field) = 1 Baud rate = 106 kbits/sec.	4B 01 01 44 03 20 08 88 04 4A 5B 09 2C 1C 80 06 75 77 81 02 80	Response command code 1 target detected Target number 1 SENS_RES[(1)] of target 1 SEL_RES[(1)] of target 1 NFCID1 length = 8 bytes NFCID1 of target 1 ATS
Bit 5 of SEL_RES indicates the target is ISO14443-4 compliant. In that case PN531 automatically sends the RATS[(2)]. ATS is indicated in the response.				
Get application ID	40 01 6A	Command code: **InDataExchange** The cmd is intended to target number 1 DESfire command: GetApplicationID	41 00 00 01 00 00 02 00 00 03 00 00 40 00 00 04 00 00 05 00 00 06 00 00	Response command code Status = 0 (OK, no error) Response of the card
DESfire commands, for example GetApplicationID command, can be sent with InDataExchange command[(3)].				
Select application	40 01 5A 06 00 00	Command code: **InDataExchange** The cmd is intended to target number 1 SelectApplication 06 00 00	41 00 00	Response command code Status = 0 (OK, no error) Response of the card

3.2.3.1 Timeout and number of retries

Cf. paragraph 3.2.2.1.

3.2.4 How to use PN531 as a FeliCa reader?

Goal: read/write FeliCa cards

Typical sequence (example):

- 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 (PN531 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: PN531 as a FeliCa reader

Action	Command[1]	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A 01 01 00 FF FF 00 00	Command code: **InListPassivTargets** Number of cards to initialize (if present in the field) = 1 Baud rate = 212 kbits/sec. Payload field of polling request	4B 01 01 12 01 01 01 06 01 46 05 C3 1A 04 01 4B 02 4F 49 93 FF	Response command code 1 target detected Target number 1 POL_RES length Response code NFCID2 PAD
FeliCa™ card has been initialised.				
Exchange data with the card	40 01 06 F0 00 FF AA BB	Command code: **InDataExchange** The cmd is intended to target number 1 Length of data + 2 FeliCa™ command (echo back) Data	41 00 06 F0 00 FF AA BB	Response command code Status = 0 (OK, no error) Response of the card
PN531 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.

3.2.4.1 Timeout and number of retries

Activation phase (InListPassiveTarget command)

By default, PN531 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). Cf. UM0501-02 page 82.

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

Either PN531 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 no request. The return status is always “No error” (00h),

3.2.5 How to use PN531 in a NFC peer-to-peer communication?

3.2.5.1 How to use PN531 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 (3.2.5.3 on page 47).

Table 8: PN531 as a NFC initiator ("reader") example 1

Action	Command[1]	Command explanation	Response	Response explanation
Initialize and activate a target	56 01 02 01 00 FF FF 00 00	Command code: **InJumpForDep** Mode (1= active mode) Baud rate = 424 kbits/sec. Optional field present (Payload) Payload field of polling request	57 00 01 AA 99 88 77 66 55 44 33 22 11 00 00 00 09 01 22	Response command code Status (0=no error) Target number 1 ATR_RES received (except cmd0 and cmd1 bytes)
Initiator has chosen the active mode. PN531 performed automatically activation and possibly parameter selection (ATR_REQ and possibly PSL_REQ)[2].				
Exchange data with the card	40 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Command code: **InDataExchange** The cmd is intended to target number 1 Data	41 00 99 88 77	Response command code Status = 0 (OK, no error) Data (Response of the target)
PN531 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.				
Deselect	44 01	Command code: **InDeselect** The cmd is intended to target number 1	45 00	Response command code Status = 0 (OK, no error)
DSL_REQ is sent with InDeselect command.				
Select	54 01	Command code: **InSelect** The cmd is intended to target number 1	55 00	Response command code Status = 0 (OK, no error)
WUP_REQ is sent with InSelect command.				
Release the target	52 01	Command code: **InRelease** The cmd is intended to target number 1	53 00	Response command code 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, PN531 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 9: PN531 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 01 02 00 FF FF 00 00	Command code: **InListPassivTargets** Number of cards to initialize (if present in the field) = 1 Baud rate = 212 kbits/sec. Payload field of polling request	4B 01 01 12 01 01 FE A2 A3 A4 A5 A6 A7 C0 C1 C2 C3 C4 C5 C6 C7	Response command code 1 target detected Target number 1 POL_RES length Response code NFCID2 PAD
The target has been initialised (in passive mode) at the requested baud rate.				
Activate the target	50 01	Command code: **InATR** The cmd is intended to target number 1	51 00 AA 99 88 77 66 55 44 33 22 11 00 00 00 09 01	Response command code Status = 0 (OK, no error)
The target has been activated.				
Change parameters	4E 01 00 00	Command code: **InPSL** The cmd is intended to target number 1 New baud rate = 106 kbits/sec	4F 00	Response command code Status = 0 (OK, no error)
The baud rate has been changed.				
Exchange data with the card	40 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Command code: **InDataExchange** The cmd is intended to target number 1 Data	41 00 99 88 77	Response command code Status = 0 (OK, no error) Data (Response of the target)
PN531 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.				
Release the target	52 01	Command code: **InRelease** The cmd is intended to target number 1	53 00	Response command code 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, PN531 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 PN531 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.

In case of use of InPSL in T=CL mode, ModWidth register value may be adapted (in order to improve RF communication) by appending new value at the end of the command parameters. (Default Modwidth value : 26h)

InPSL command code	Target number	Baudrate from initiator to target	Baudrate from target to initiator	Modwidth(optional and relevant only for T=CL)

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, PN531 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). Cf. UM0501-02 page 82.

The timeout depends on the baudrate. At 106 kbps, paragraph 3.2.1.1 applies. At 212 or 424 kbps, paragraph 3.2.4.1 applies.

Either PN531 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 (UM0105-02 page 80).

By default, PN531 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). Cf. UM0501-02 page 82. It is recommended to set it to 0 or 1 (see following page "Recommendation to avoid a potential problem in passive mode, during activation")

- **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 (UM0501-02 page 80).

By default, PN531 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). Cf. UM0501-02 page 82. It is recommended to set it to 0 or 1 (see following page “Recommendation to avoid a potential problem in passive mode, during activation”)

- **Communication phase (InDataExchange command)**

The timeout value depends on value returned by the target (WT), as specified in ISO/IEC 18092. The timeout extension mechanism is fully embedded inside PN531. The error handling and the chaining are also fully managed by PN531.

- **Deactivation phase (InDeselect/InRelease command)**

InDeselect command performs a DSL_REQ. InRelease command performs a RLS_REQ. The timeout value is the same than during communication phase.

Note: It is not needed to use InDeselect (and InSelect) command to handle two cards. Indeed, when using InDataExchange command, PN531 automatically wakes up the card corresponding to the desired TgNb, and automatically put in SLEEP state the other one.

Recommendation to avoid a potential problem in active mode, during activation:

Problem description:

When trying to activate a target in active mode, no response seems received from the target.

Root cause:

This problem happens in case the target is at the limit of the RF field (e.g. the target is just entering the field).

In case the target is at the limit of the RF field, the initiator may detect a RFError, even if the target answered correctly to the ATR_REQ. As the target actually responded, the RF field of the initiator is switched on. However, the initiator didn’t correctly understand the answer of the target, so it retries to activate, by setting the bit InitialRFOn and sending a new ATR_REQ. But setting bit InitialRFOn when RF field is on (which the case of the initiator) prevents the RF field to be switched off at the end of the transmission. As the initiator does not switch its RF field off, the target cannot answer.

Workaround:

A “retry mechanism” must be implemented in the software of the initiator’s host controller. After expiration of a timeout, the host controller shall resend the activation command.

Recommendation to avoid two potential problems in passive mode, during activation, at 106 kbps:

Problem 1 description:

When trying to activate a target in passive mode, no response seems received from the target.

Root cause:

This problem happens in case the target is moved away from the initiator, just between the end of the passive initialisation (SEL_RES) and the beginning of the activation (ATR_REQ), and then presented again to the initiator.

Using default PN531 settings, the initiator sends an infinite numbers of ATR_REQ until a target answers. Consequently, the initiator keeps sending some ATR_REQ, when the target is moved away, and then moved close. The target detects an active mode when it is put again close to the initiator (since it receives a ATR_REQ), so it won't answer until the initiator has switched off its RF field. But the initiator will not switch off its field since it initiated a passive communication. Consequently, the target and the initiator will not be able to "see" each other.

Workaround:

Set the number of retry sending ATR_REQ to 0 or 1, using RFConfiguration command item 5. The command (without protocol encapsulation) consists in "32 05 00 xx xx" ; and keep the number of retry passive activation (MaxRtyPassiveActivation) to a value higher than 1, so that in case of problem, the initiator restarts the complete sequence (default PN531 setting corresponds to an infinite number of passive activation retries).

Problem 2 description:

When trying to activate a target in passive mode, the initiator is blocked, although the target answered.

Root cause:

This problem may happen in case the target is at the limit of the RF field of the initiator. The target sends correctly its ATR_RES. The initiator receives the beginning of the response (including the length byte) but not the end. It keeps waiting for the end of the response.

Workaround:

A "retry mechanism" must be implemented in the software of the initiator's host controller. After expiration of a timeout, the host controller shall resend the activation command(s) (e.g. InJumpForDEP).

3.2.5.2 How to use PN531 as a target in a NFC peer-to-peer communication?

In this mode, PN531 is configured as target, meaning it keeps waiting for an initiator command.

PN531 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 PN531. PN531 transfers the response from the host to the initiator. Initialisation/activation is handled automatically by PN531.

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:

- **TgInitTamaTarget**, to configure PN531 as a target,
- **TgGetDEPData**, to wait for data coming from the initiator,
- **TgSetDEPData**, to respond to the initiator.

Table 10: PN531 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	Command code: **TgInitTamaTarget** 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 A7	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 PN531				
Wait for data to be transferred to the host controller	86	Command code: **TgGetDEPData**	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: **TgSetDEPData**	8F 00	Response command code Status = 0 (OK, no error) No data sent back to the controller
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 PN531 as a target?

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

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

How to fill TgInitTamaTarget 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.

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 (47 bytes maximum)

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 TgInitTamaTarget parameters.
- Or after having received the ATR_REQ. In that case, the bytes are transmitted from the host controller to PN531 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 SetTamaParameters command to disable automatic sending of ATR_RES upon reception of ATR_REQ. The ATR_RES will be sent by TgSetGeneralBytes command.

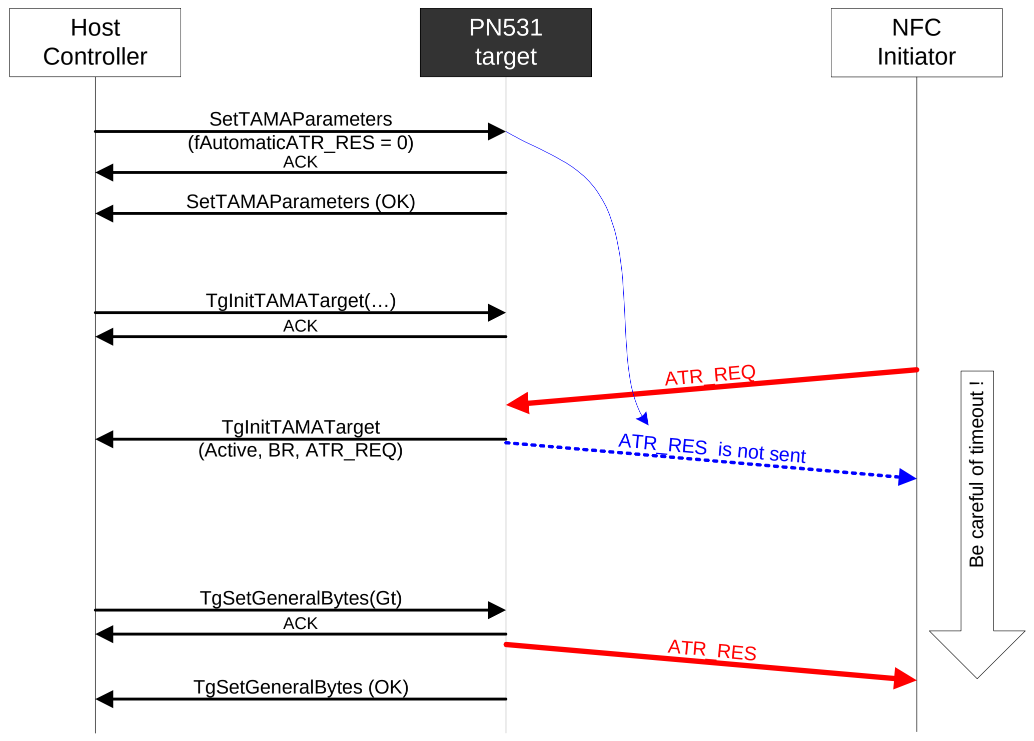


Fig 17. TgSetGeneralBytes

3.2.5.3 Handling of several cards/targets

PN531 can handle 2 cards “at the same time”, or 1 card and 1 NFC target.

PN531 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, PN531 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)

Table 3 on page 26 shows an example with 2 Mifare cards.

Warning:

In case of one Mifare card and one NFC target (in passive mode at 106 kbps). The host controller can communicate with the Mifare card and then switch to the NFC target (using InDataExchange function and the logical number), but it cannot switch again to the Mifare card. In the same way, the host controller can start communicating with the NFC target and switch to the Mifare card, but then it cannot switch again to the NFC target. See the end of paragraph 3.2.1

3.2.5.4 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 TgGetDEPData and TgSetDEPData 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 (larger that 252 bytes). In that case, the target can directly continue the exchange with TgGetDEPData (no TgSetDEPData needed).

- From target to initiator:

The target can provide the initiator with large amount of data (larger than 252 bytes) using TgSetMetaData function. The initiator has sent a InDataExchange function. The response to the initiator is sent via TgSetMetaData function instead of TgSetDEPData 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 intiator to the target). Last packet of data (smaller tha 252 bytes) will be transferred with TgSetDEPData function.

Please refer to PN531 User manual (cf. References table on page 4) for detailed explanation.

3.2.5.5 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 NFC TAMA, depends on several parameters:

- The RF baudrate
- The amount of data:
    - o TAMA length reduction[1] 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. TAMA host protocol limits the size of useful data transmitted at once to 252 bytes

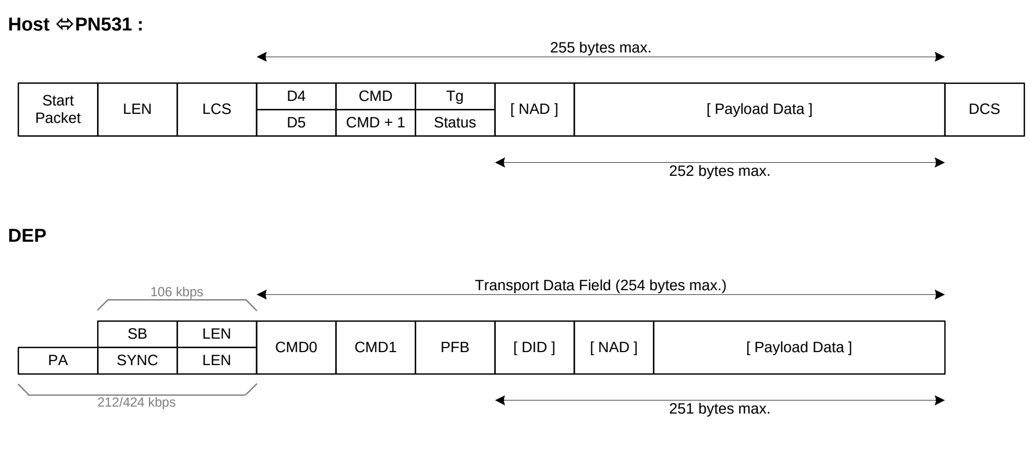


Fig 18. NFC Data Exchange protocol and PN531 host link protocol overhead

- The CPU frequency
- The link used between the host controller and the NFC device (SPI or HSU or USB 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 is between 5 kbps and 60 kbps. (Worst case: HSU at 9600 bauds, CPU frequency at 6.78 MHz)

(What is measured is A - (1+2+3+C) according to next figure)

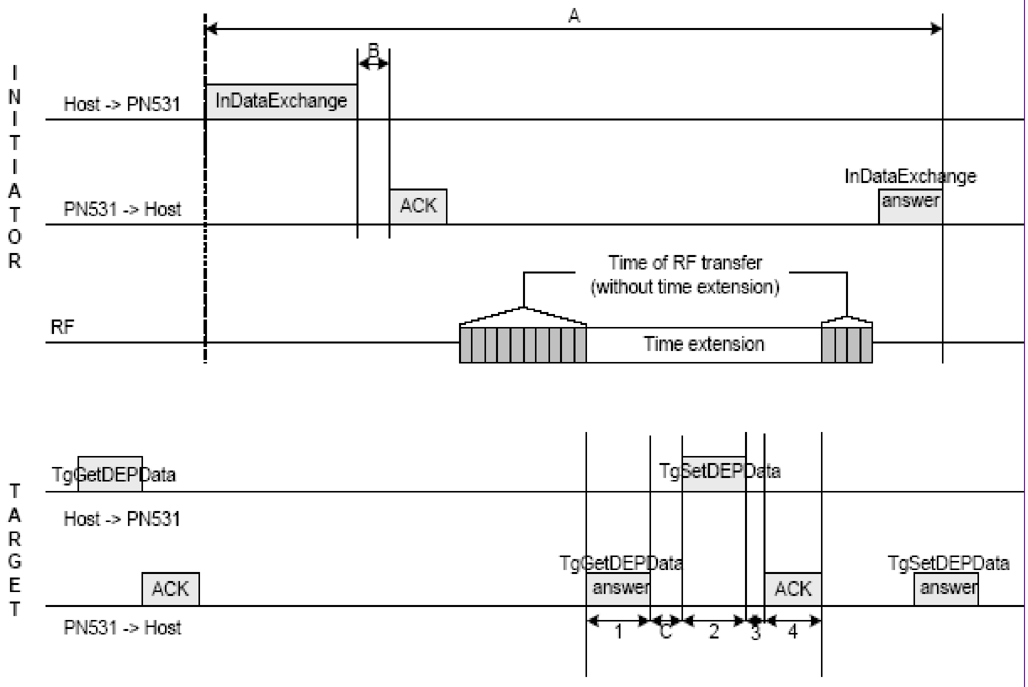


Fig 19. Throughput measurement description

Initiator	A	The time between the beginning of the InDataExchange and the end of its answer. It represents a complete cycle of transfer.
	B	The time between the end of the InDataExchange and the beginning of its acknowledge. This is the delay in which PN531 initiator analyzes the validity of the frame sent by the host controller.
Target	C	The time between the end of the answer of the TgGetDEPData and the beginning of the TgSetDEPData. This is a timing related to the Host Controller of the target.
	1	This is the time of the TgGetDEPData answer. For these tests this time never changes.
	2	This is the time of the TgSetDEPData. For these tests this time never changes.
	3	The time between the end of the TgSetDEPData and the beginning of its acknowledge. This is the delay in which PN531 target analyzes the data received by the host controller.
	4	This is the time of the TgSetDEPData acknowledge. For this tests this, time never changes.

Fig 20. Legend of previous figure

3.2.6 How to use Smart connectivity (combination of PN531 and SMX)?

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

Combining PN531 and SMX (P5CN072) allows to deal with application that requires security such as payment applications.

In this document, PN531 is used in combination with a smart card (SMX). S2C interface is used.

Commands needed to use PN531 + SMX are:

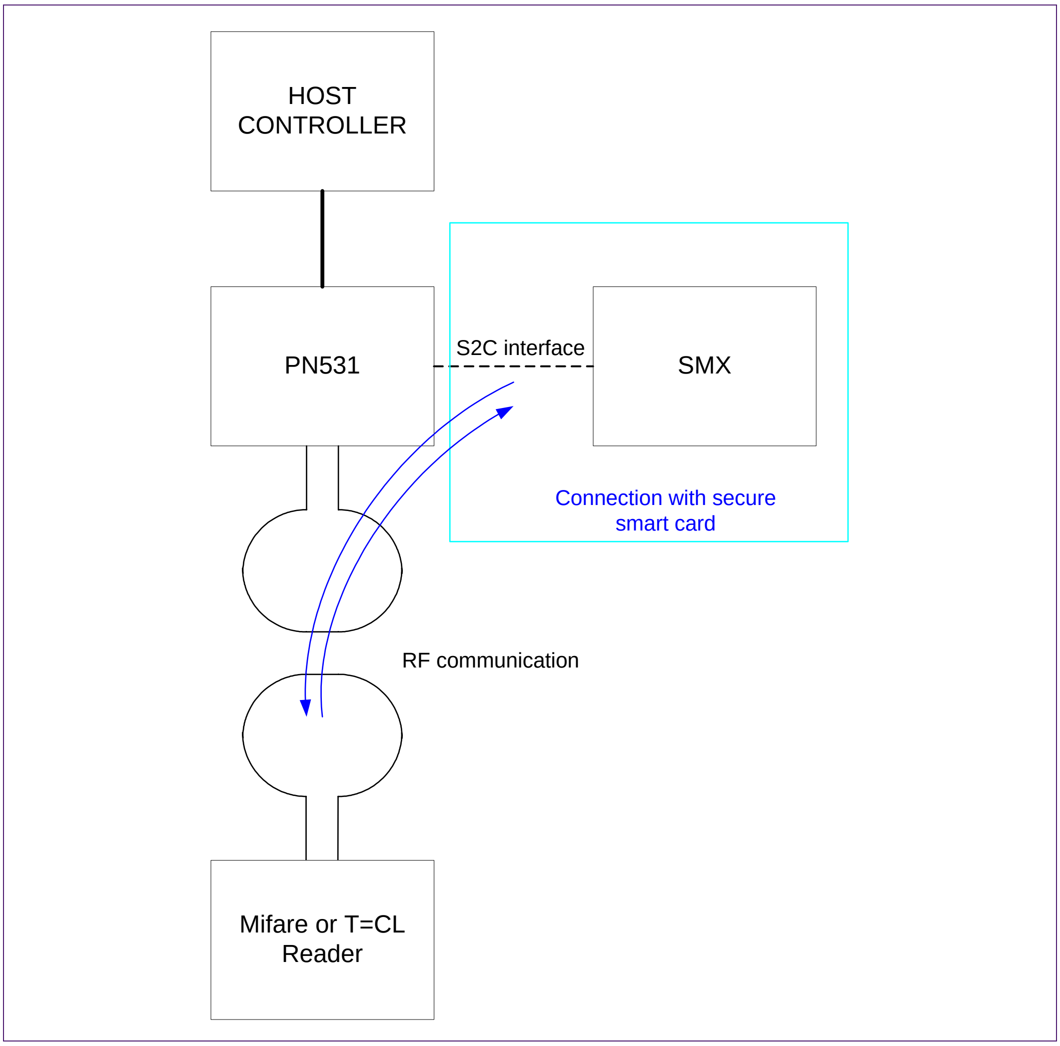
- **SAMConfiguration**, to chose between normal, wired or virtual mode,
- **SetTAMAParameters**, to possibly disable automatic RATS sending (T=CL mode).

3.2.6.1 Virtual card mode

In virtual card mode, PN531 (+SMX) is seen as a contact less secure smart card. Only one command, SAMConfiguration, is needed to put TAMA+SMX in this mode.

Optionally, PN531 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, TAMA only acts a bridge between SMX and the external reader.



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

Table 11: PN531 +SMX as virtual card

Action	Command[1]	Command explanation	Response	Response explanation
Set PN531 in virtual card mode	14 02 00	Command code: **SAMConfiguration** Virtual card mode No timeout	15	Response command code
	PN531 is configured in virtual card mode. SMX is seen by an external reader as a contactless card.			
Power down	16 09	Command code: **PowerDown** Wake up sources : INT0 and RF field[2]	17 00	Response command code Status = 0 (OK, no error)
	PN531 has been switched to power down and can be wake up either by external RF field or by INT0.			

(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).

3.2.6.2 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, PN531 access SMX to check what happened.

SMX can communicate either in Mifare or in ISO14443-4 protocol.

PN531 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 PN531 must be disable, as shown in Table 13: on page 55.

Table 12: PN531 +SMX as wired ISO1443-4 card

Action	Command[1]	Command explanation	Response	Response explanation
Optional reset SAM mode	14 01	Command code: **SAMConfiguration** reset	15	Response command code
Reset SAM configuration to normal mode				
Set PN531 in wired card mode	14 03	Command code: **SAMConfiguration** Wired card mode	15	Response command code
PN531 is configured in wired card mode. SMX is accessed by PN531 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)
PN531 communicates with the SMX as with a card. If SMX indicates T=CL compliance, PN531 automatically sends RATS command.				

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

Table 13: PN531 +SMX as wired Mifare® card

Action	Command[1]	Command explanation	Response	Response explanation
Optional reset SAM mode	14 01	Command code: **SAMConfiguration** reset	15	Response command code
Reset SAM configuration to normal mode				
Disable automatic sending of RATS command	12 04	Command code: **SetTAMAParameters** Automatic ATR_RES = 1	13	Response command code
PN531 is configured in wired card mode. SMX is accessed by PN531 as a contactless card.				
Set PN531 in wired card mode	14 03	Command code: **SAMConfiguration** Wired card mode	15	Response command code
PN531 is configured in wired card mode. SMX is accessed by PN531 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 PN531. 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
PN531 communicates with the SMX using Mifare® commands.				

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

3.2.6.3 Dual card mode

In this mode, both TAMA (as a target) and SMX will be visible from an external reader. 2 commands are needed:

- SAMConfiguration
- TgInitTAMATarget

Table 14: PN531 +SMX as wired ISO1443-4 card

Action	Command[1]	Command explanation	Response	Response explanation
Optional reset SAM mode	14 01	Command code: **SAMConfiguration** reset	15	Response command code
Reset SAM configuration to normal mode				
Set PN531 in Dual card mode	14 04	Command code: **SAMConfiguration** Dual card mode	15	Response command code
PN531 is configured in dual card mode.				
Configure TAMA 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	Command code: **TgInitTamaTarget** 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 A7	Response command code Mode : passive mode, 424 kbits/s Initiator command received (ATR_REQ)
SMX and TAMA can respond to a reader only after TgInitTAMATarget command has been sent.				

3.2.7 Power down mode

To put PN531 in power down, PowerDown command shall be used. PowerDown command parameters indicate which wakes up sources are enabled.

3.2.7.1 Going into power down

When PowerDown command is executed:

- SVDD is switched off (SAM interface power supply)

- The contactless UART is put into power down (included RF level detector, unless it is selected as wake up source)

- The microcontroller is put in power down mode (PCON = 02h),

Warning:

The RF field generation is not automatically switched off. The host controller must send a RFConfiguration command to switch off the RF field, before going into power down (command byte and parameters: “ 32 01 00”).

3.2.7.2 Waking up

Wake up sources byte description:

Bit7 and Bit6: RFU

Bit5: SPI.

When set to 1, the system can be woken up by maintaining NSS line low during 300 µs (depending on the temperature, the power supply, … refer to PN531 datasheet).

Bit4: HSU.

When set to 1, the system can be woken up by RX line (5 rising edge). PN531 needs time to wake up (about 200 µs. Refer to PN531 datasheet for details). To be sure that the command is not lost, the host controller can send a command with a large preambule, or send a dummy command byte e.g. 55h and wait before sending the relevant command frame.

Bit3: RF level detector

When set to 1, the system can be woken up by the (external) RF field. When this wake up source is selected, the contact-less UART is not put in power down mode (only the microcontroller is).

Bit2: USB.

This bit must not be used, since in USB mode, PN531 manages itself suspend mode and resume.

Bit1: INT1 (P33)

When set to 1, the system can be woken up by INT1 interrupt.

Bit0: INT0 (P32)

When set to 1, the system can be woken up by INT0 interrupt.

Particularly, when handshake mode is selected, INT0 wake up source allows the system to wake up on a negative pulse on H_REQ.

Table 15: PowerDown example

Action	Command[1]	Command explanation	Response	Response explanation
Switch RF field off	32 01 00	Command code: **RFConfiguration** Item 1 RF off	33	Response command code
We make sure that RF field generation is off				
PutPN531 in power down	16 09	Command code: **PowerDown** Wake up sources : RF field and INT0	17 00	Response command code Status OK
PN531 is in power down.				

4. Application schematics

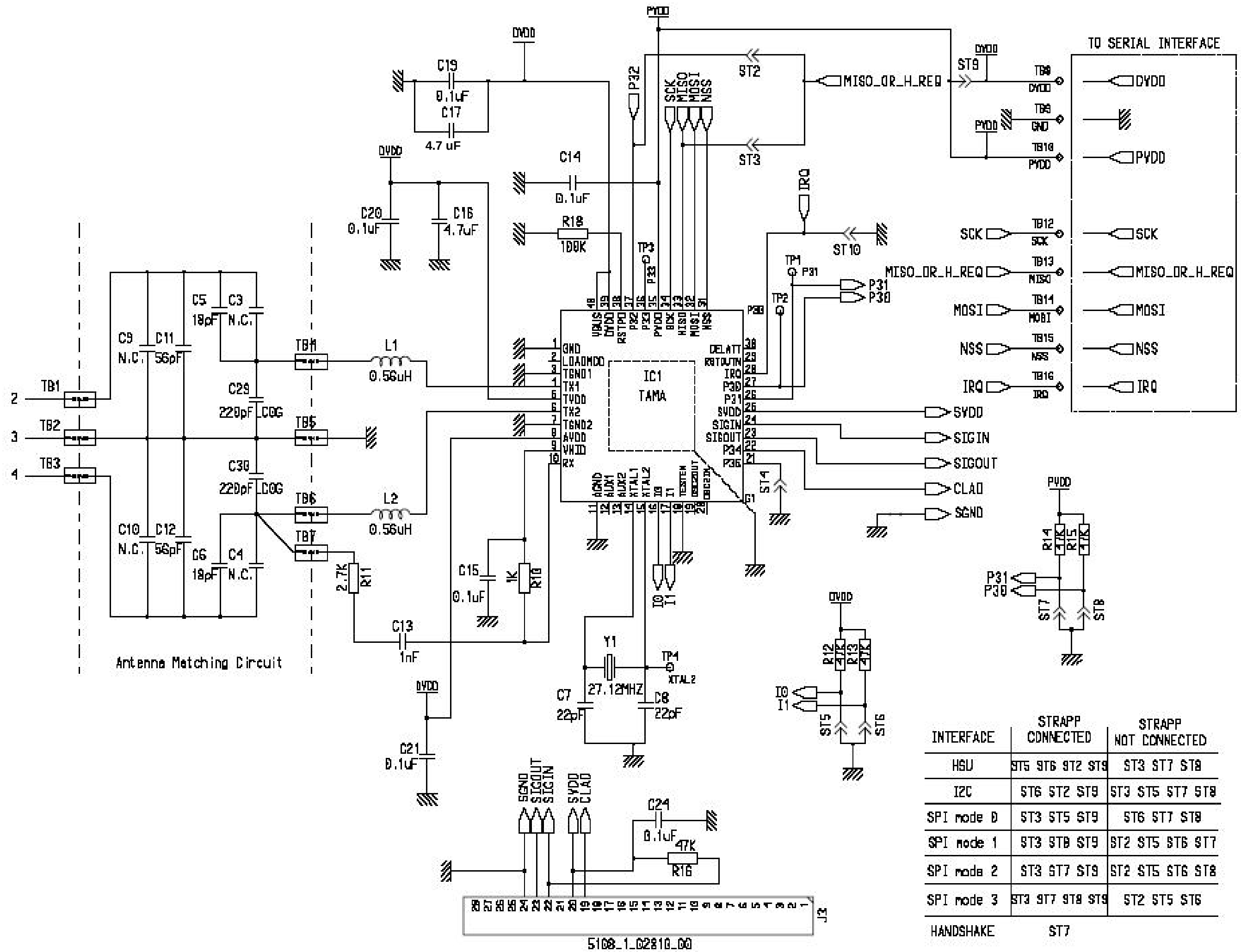
4.1 Recommendations

The interface is configurable by I0 and I1 pins (pins 16 and 17), as described in paragraph 2 on page 5.

Recommended capacitors:

- On TVDD: 4.7 µF X7R ceramic in parallel with 100 nF;
- On DVDD: 4.7 µF (for serial, I2C, SPI) or 10 µF (for USB) 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”.

4.2 Serial application schematics



INTERFACE	STRAPP CONNECTED	STRAPP NOT CONNECTED
HSU	ST5 ST6 ST2 ST9	ST3 ST7 ST8
I2C	ST6 ST2 ST9	ST3 ST5 ST7 ST8
SPI mode 0	ST3 ST5 ST9	ST6 ST7 ST8
SPI mode 1	ST3 ST8 ST9	ST2 ST5 ST6 ST7
SPI mode 2	ST3 ST7 ST9	ST2 ST5 ST6 ST8
SPI mode 3	ST3 ST7 ST8 ST9	ST2 ST5 ST6
HANDSHAKE	ST7	

Fig 21. Serial application schematic – sheet 1/3

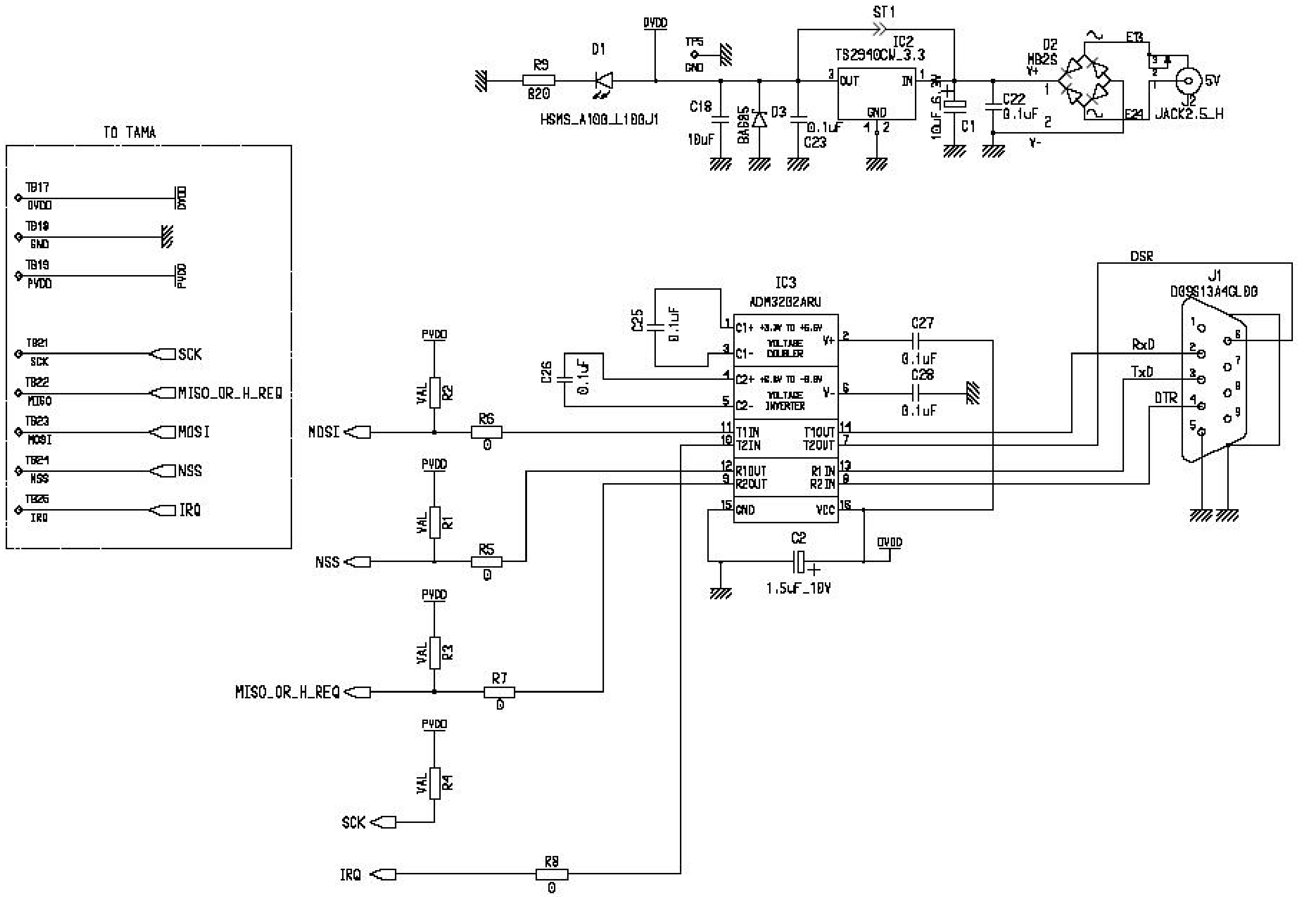


Fig 22. Serial application schematic – sheet 2/3

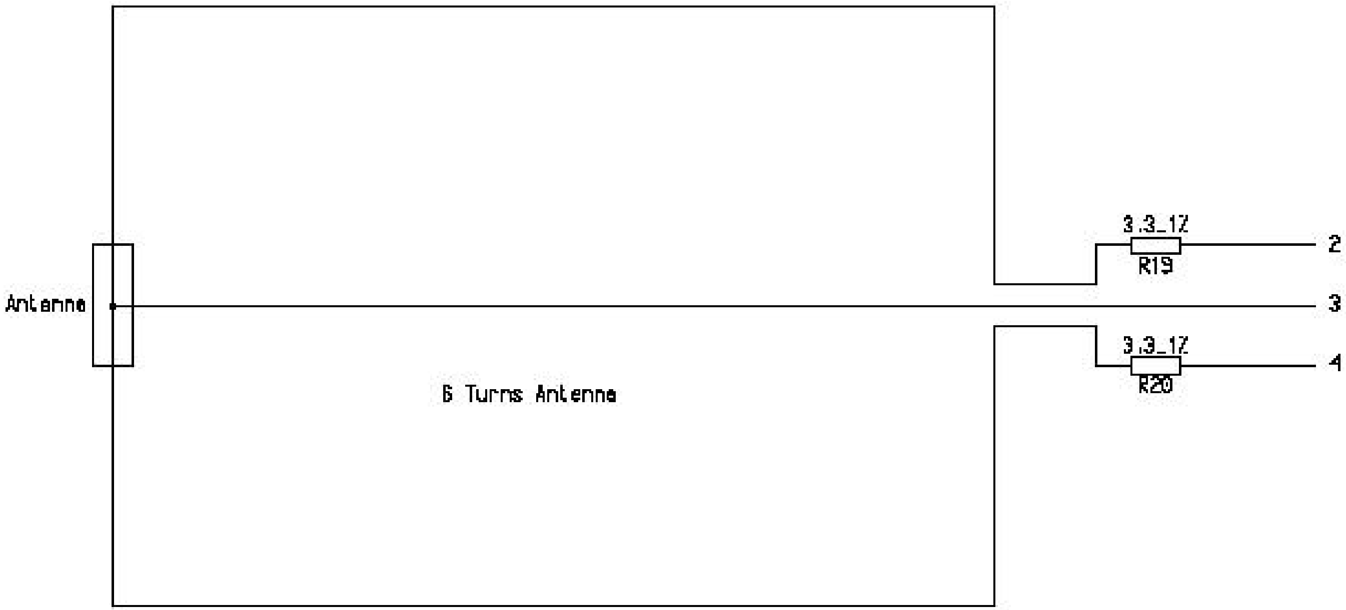


Fig 23. Serial application schematic – sheet 3/3

4.3 USB application schematics

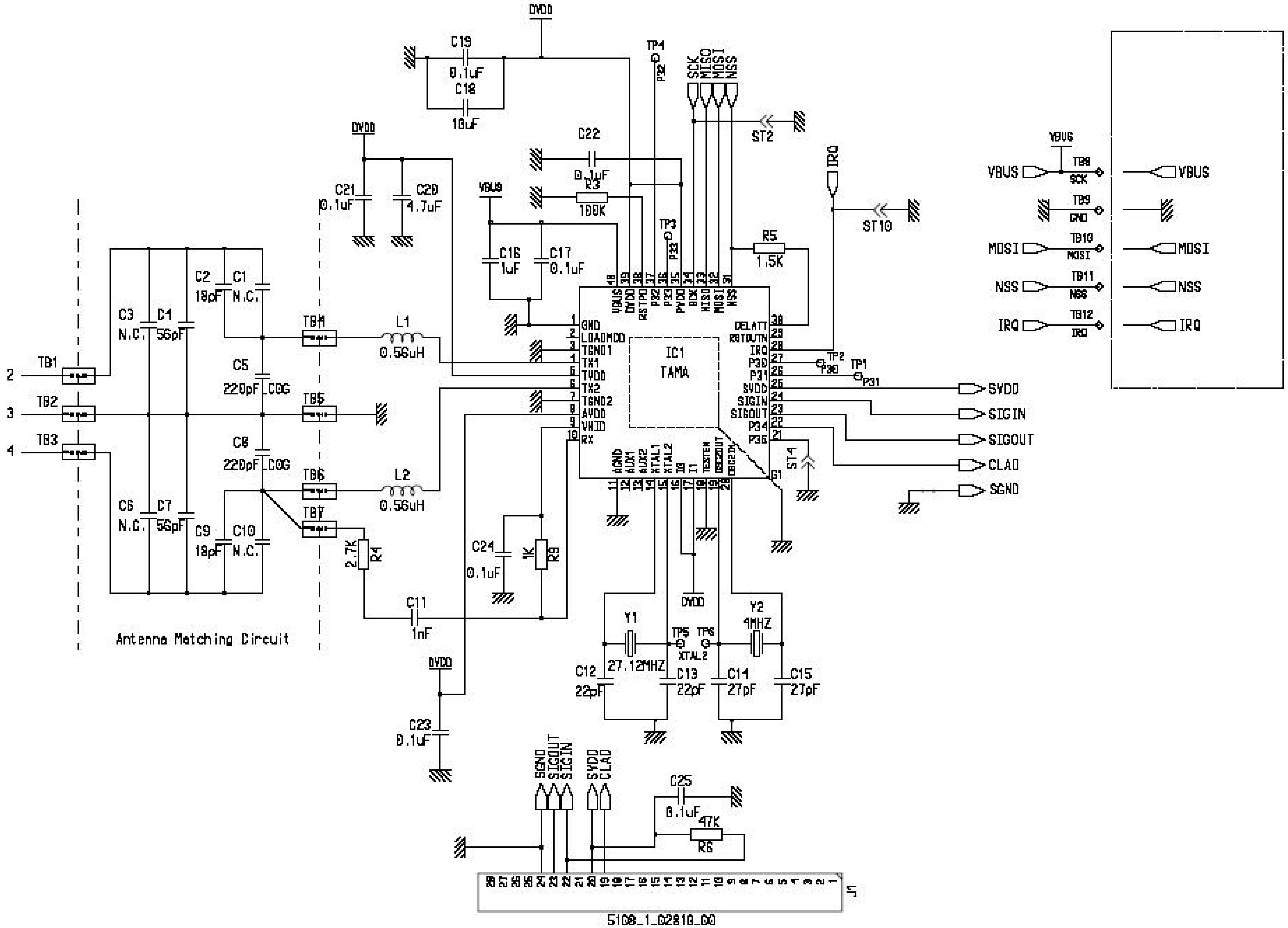


Fig 24. USB application schematic – sheet 1/3

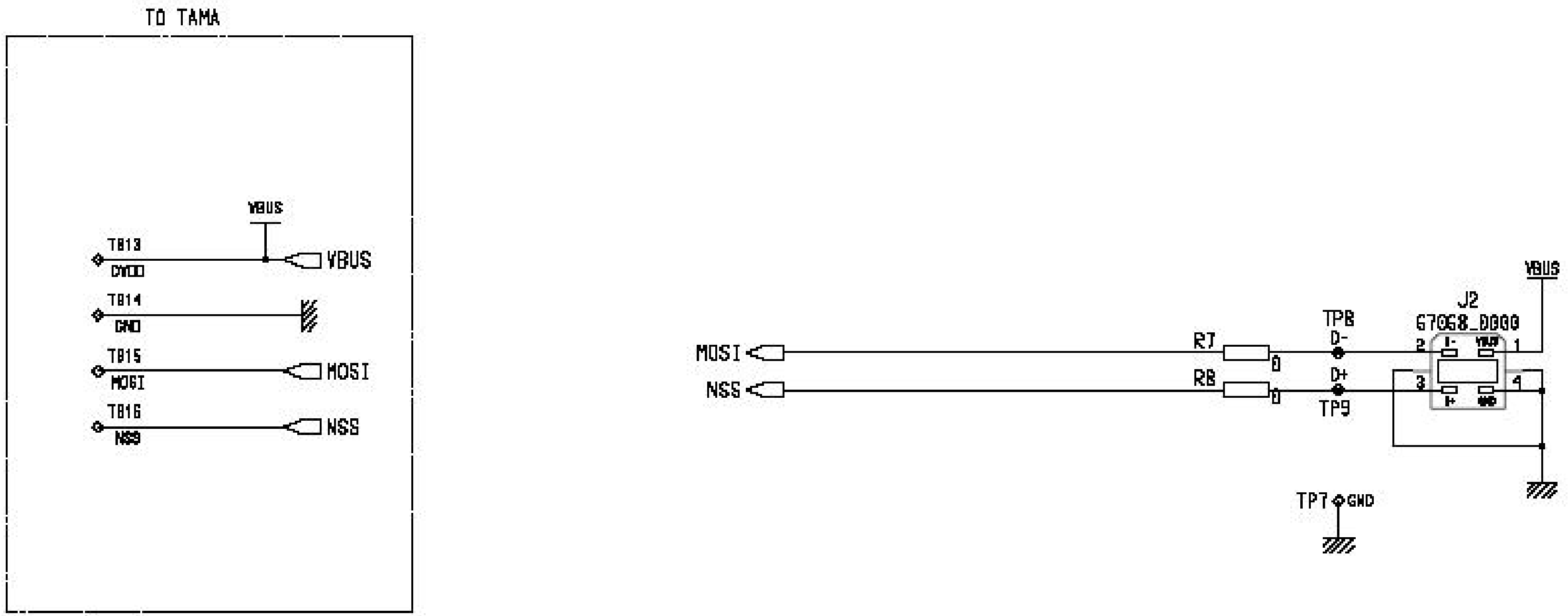


Fig 25. USB application schematic – sheet 2/3

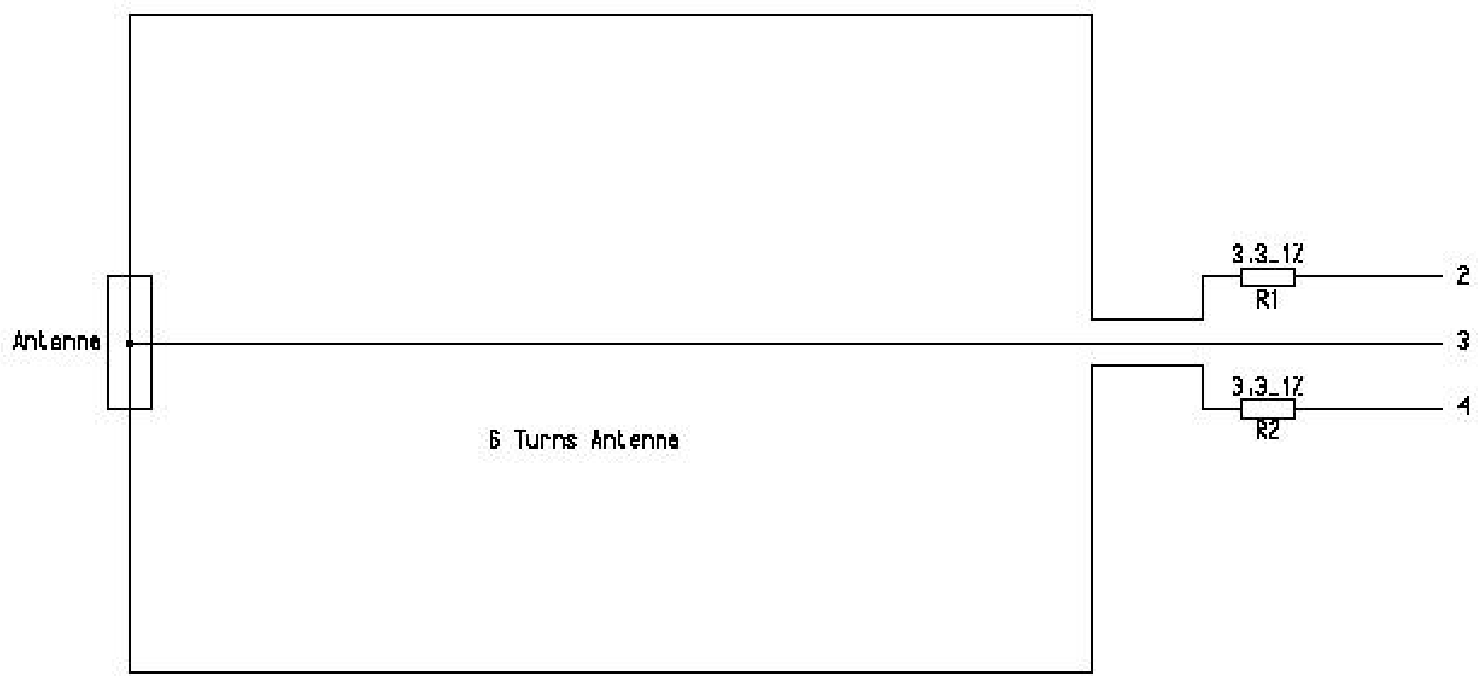


Fig 26. USB application schematic – sheet 3/3

5. Recommendations

This paragraph summarizes the recommendations to work around potential problems.

- To activate a card (Mifare or ISO14443) with a known UID, use at least four bytes of its UID in InListPassiveTarget command. Cf. end of paragraph 3.2.1
- When PN531 acts as an initiator communicating with one Mifare card and one NFC target, it is not possible to switch several times from the card to the target, unless the host controller of the target restart TgInitTAMATarget command followed by TgGetDEPData command. Cf. end of paragraph 3.2.1 and paragraph 3.2.5.3.
- Before using command PowerDown, make sure the RF field is switch off using RFConfiguration command. Cf. paragraph 3.2.7.1
- Use default CPU frequency.
- In peer to peer mode (active or passive), if the target is at the limit of the RF field, there may be some communication problems. The host controller shall implement a “retry mechanism” (after a timeout expired, re-launch the activation command). Cf. pages 41 and 42.
- In passive mode, if the target is at the limit of the RF field, there may be some communication problems. The number of retries ATR_REQ (MaxRtyATR parameter in RFConfiguration command) must be set to 0 or 1, so that the initiator restarts the complete passive activation sequence. Cf. page 42.
- DID and NAD shall not be used during peer-to-peer communication.

  PN531 never uses DID nor NAD during time out extension request or response. Consequently, two PN531 configured to use DID or NAD will be able to communicate together, but there can be communications problems between PN531 and other NFC devices.
- In I2C, in target mode, after exchange of data (TgGetDEPData, TGSetDEPData), a command can be not acknowledge (I2C NACK). To work around this potential issue, the following write registers command must be send before configuring the target mode:

  Write 00 at address 01D9, and Write 00 at address 01DA: command frame (without the protocol encapsulation) 08 01 D9 00 01 DA 00.
- PN531 can handle incorrectly an ISO7816-4 Case 1 APDU command in ISO14443-4 R/W mode.

  If a host command is sent to PN531 containing an ISO7816-4 Case 1 APDU command (CLA INS P1 P2) **and** DCS byte is greater than value 0xFA, an error 0x07 will be returned by PN531.

  Following issues can happen in case of bad RF communication conditions:
- A target will send twice the data packet to its host controller after an ATTENTION REQUEST has been received

  The host controller shall implement a timeout mechanism, and reinitiate the transaction.
- An infinite loop of TOX_REQ/TOX_RES can happen after an ATTENTION REQUEST.

  The host controller shall implement a timeout mechanism, and reinitiate the transaction.
- When PN531 as an initiator receives no ACK from the Target after a DEP_REQ, it does not send ATTENTION REQUEST as it should do. It keeps waiting forever for ACK, without sending any error back to the host. The host controller shall implement a timeout mechanism, and reinitiate the transaction.

- Using Metachaining functionality, when RF conditions are bad (i.e. for example TOX_REQ is not seen by the initiator, or ACK is not seen after a DEP_REQ), some bytes can be lost.

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

6. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status ‘Production’), relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

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

7. Licenses

Purchase of Philips I^2C components

Purchase of Philips I^2C components conveys a license under the Philips’ I^2C patent to use the components in the I^2C system provided the system conforms to the I^2C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

Purchase of Philips RC5 components

Purchase of Philips RC5 components conveys a license under the Philips RC5 patent to use the components in RC5 system products conforming to the RC5 standard UATM-5000 for allocation of remote control commands defined by Philips.

8. 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.

9. Trademarks

FeliCa ™ is a trademark of Sony Corporation

Mifare® is a registered trademark of Koninklijke Philips Electronics N.V.

10. Contents

Date of release:27 november 2006

Document number: AN10394_4

Published in The Netherlands

Date of release:2006-11-27

Document number <12NC>

Published in The Netherlands