

# AN10682

## How to use PN533

Rev. 3.1 — 10 July 2018  
157631

Application note  
COMPANY PUBLIC

### Document information

Info	Content
Keywords	NFC, ISO/IEC 14443, ISO/IEC 18092, PN533, PN5331B3HN, PayPass, EMVCo
Abstract	The PN533 Application note.



**Revision history**

Rev	Date	Description
3.1	20180710	Editorial updates
3.0	20171031	Security status changed into Company public, no content change
2.2	20101108	Update: HSU application schematic added, more info about communication with a MIFARE Plus card
2.1	May 27,2010	Update: Host powered application (HSU interface), communication with a MIFARE Plus card, SAM naming removed.
2.0	March 29, 2010	Update: HSU interface, NFC-WI interface
1.3	November 14, 2008	Updated chapter 12.3: "The PN533 various modes"
1.2	September 25, 2008	Updated chapter "USB 2.0 compliancy"
1.1	September 5, 2008	Update: Chapter "USB 2.0 compliancy" added, RF amplifier schematic ...
1.0	June 20, 2008	Final version for PN5331B3HN
0.2	April 24, 2008	Update
0.1	April 10, 2008	Creation: application note for samples PN533/C250

**Contact information**

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

## 1. Introduction

The PN533 is a highly integrated transmission module for contactless communication at 13.56 MHz including microcontroller functionality based on an 80C51 core.

The PN533 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 PN533 includes a switch to power an external secure IC connected to the NFC-WI interface. It is controlled by the embedded firmware.

The PN533 has also a Master I<sup>2</sup>C interface enabling to drive an I<sup>2</sup>C peripheral (memory or contact card reader IC).

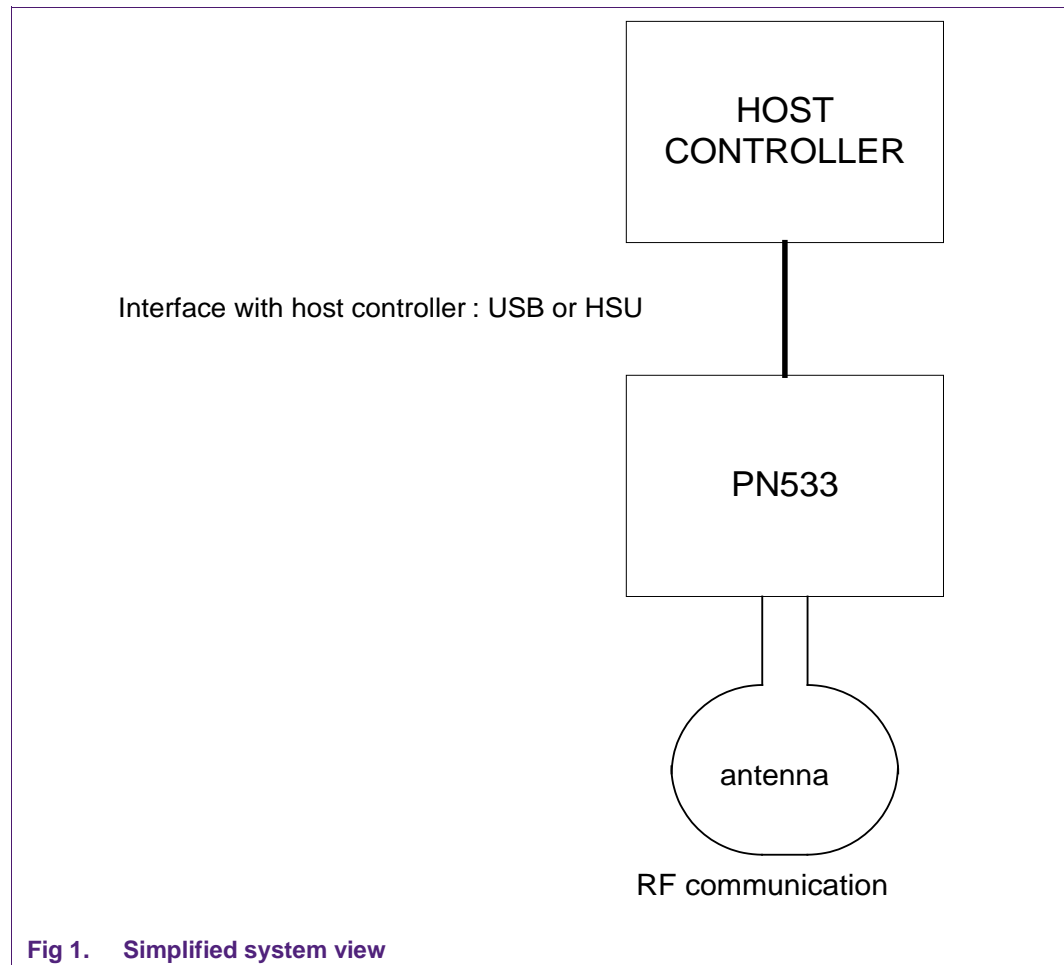


Fig 1. Simplified system view

This document intends to allow the customer getting quickly started with the PN533. It summarizes commands needed to use the PN533 as a card reader/writer, or in a NFC peer-to-peer communication.

Detailed description of the PN533 firmware can be found in the PN533 User manual (refer to Section [15 "References"](#)).

More description of the PN533 IC can be found in the PN533 data sheet.

The **PN5331B3HN** with **embedded firmware** has following features:

- Supports ISO/IEC 14443A reader/writer up to 847Kbit/s
- Supports ISO/IEC 14443B reader/writer up to 847Kbit/s
- Supports MIFARE Classic with 1K/4K encryption in reader/writer mode at 106Kbit/s
- Supports all NFCIP-1 modes up to 424Kbit/s. The PN533 handles the complete NFC framing and error detection.
- Supports contactless RF communication according to the Felica protocol at 212Kbit/s and 424Kbit/s
- Embedded firmware commands allow compliancy with Paypass v1.1 and EMVCo v2.0 specifications
- Embedded firmware commands allow use of the NFC secure layer
- Embedded firmware commands allow RF Activation application
- Reader mode for Innovision Jewel cards
- Includes 80C51 micro-controller
- Integrated LDO to allow 2.7 to 5.4V power supply voltage
- Integrated antenna component detector
- Host interfaces:
  - USB 2.0 full speed interface
  - High Speed UART (data rate up to 1.288 Mbit/s)
- USB bus-powered or host-powered mode possibility
- On-chip PLL to generate internally 96 MHz for the USB interface
- I<sup>2</sup>C master interface to fetch PID, VID, USB descriptor and RF settings from an external EEPROM
- I<sup>2</sup>C master interface to support the bridge to the TDA8029 contact reader (2 dedicated GP-IOs)
- NFC-WI interface compliant with ECMA 373 to connect an external secure IC.
- Power switch for external secure companion chip.
- 3 additional GP-IOs for external devices control

In this document the term „MIFARE Classic card“ refers to a MIFARE Classic IC-based contactless card, the term „MIFARE Ultralight card“ refers to a MIFARE Ultralight IC-based contactless card and the term „MIFARE DESFire card“ refers to a MIFARE DESFire IC-based contactless card.

## 2. PN533 configuration modes

---

Default mode of PN533 is standard mode.

To behave in standard mode, two GPIOs need to be configured during the power-on reset phase of the IC.

Notice that no external resistors are needed to pull selection pins to high level; they can be left open.

**Table 1. Configuration modes**

Mode	Selection Pins	
	P70_IRQ (pin #21)	P35 (pin #20)
Standard	1	1
	0	0
PN512 emulation	1	0
RF field ON	0	1

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

**Table 2. IC correspondence**

Commercial name	Application Note name
PN5331 B3HN	PN533

## 2.1 Standard mode

This is the default mode of the PN533. The description of this mode is detailed in this document.

## 2.2 PN512 emulation mode

In this test mode, the PN533 is configured to act as real PN512 IC using serial interface.

The PN512 is a transmission module for contactless communication at 13.56 MHz. It integrates a modulation and demodulation concept for different kind of contactless communication methods and protocols.

The link used is RS232 at 9600 bauds. It is not possible to change the value of the baud rate.

The emulation of the PN512 IRQ pin is supported as well; the pin used is P70\_IRQ.

The level of the P70\_IRQ pin is low when an interrupt occurs.

## 2.3 RFfieldON Mode

In this mode, the PN533 is configured to switch on its RF field immediately after the reset.

The modulation and the baud rate used depend on the selection GPIOs P33\_INT1 and P34 and random data bytes are continuously sent.

In this mode, the temperature sensor is not activated, so that tests can be done at temperature higher than 125°C.

**Table 3. TX framing and TX speed in RFfieldON configuration**

TX framing – TX speed	Selection Pins	
	P33_INT1 (pin #33)	P34 (pin #34)
MIFARE Classic - 106 kbps	1	1
	0	0

TX framing – TX speed	Selection Pins	
	P33_INT1 (pin #33)	P34 (pin #34)
FeliCa - 212 kbps	0	1
FeliCa - 424 kbps	1	0

### 3. PN533 block diagram

The PN533 is based on an 8051 core. The chip contains a contactless UART, a contactless front end and a “PCR” block that controls clocks and power.

It can be connected to the host controller through USB or HSU interface. The interface is selected using I0 and I1 pins.

The integrated fractional PLL generates clock for the USB interface from the 27.12MHz crystal oscillator.

The IC can be powered directly from USB supply VBUS (between 4.02V and 5.25V).

An internal voltage regulator (LDO) is used to generate other supply voltages (typical output= 3.3V).

The pad power supply PVDD must be provided between 1.6V and 3.6V.

The PN533 integrates an I<sup>2</sup>C bus Master interface to drive an external memory EEPROM or a contact reader IC like the TDA8029. External pull-up resistors connected to DVDD supply are mandatory on I<sup>2</sup>C bus lines.

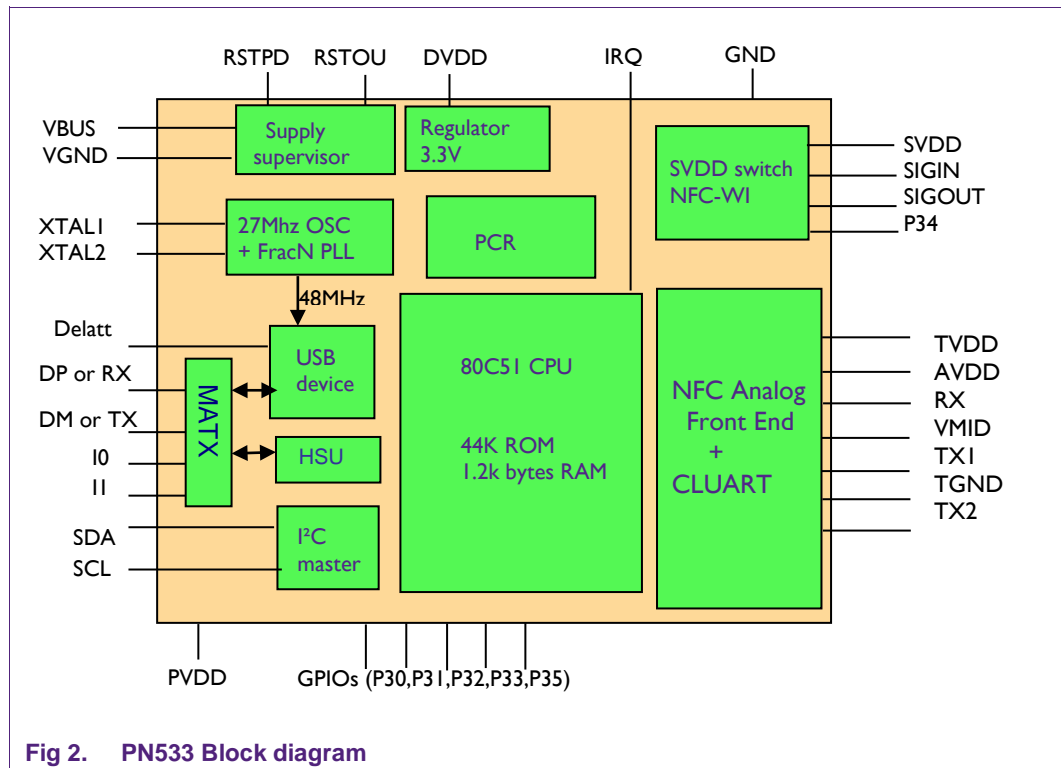


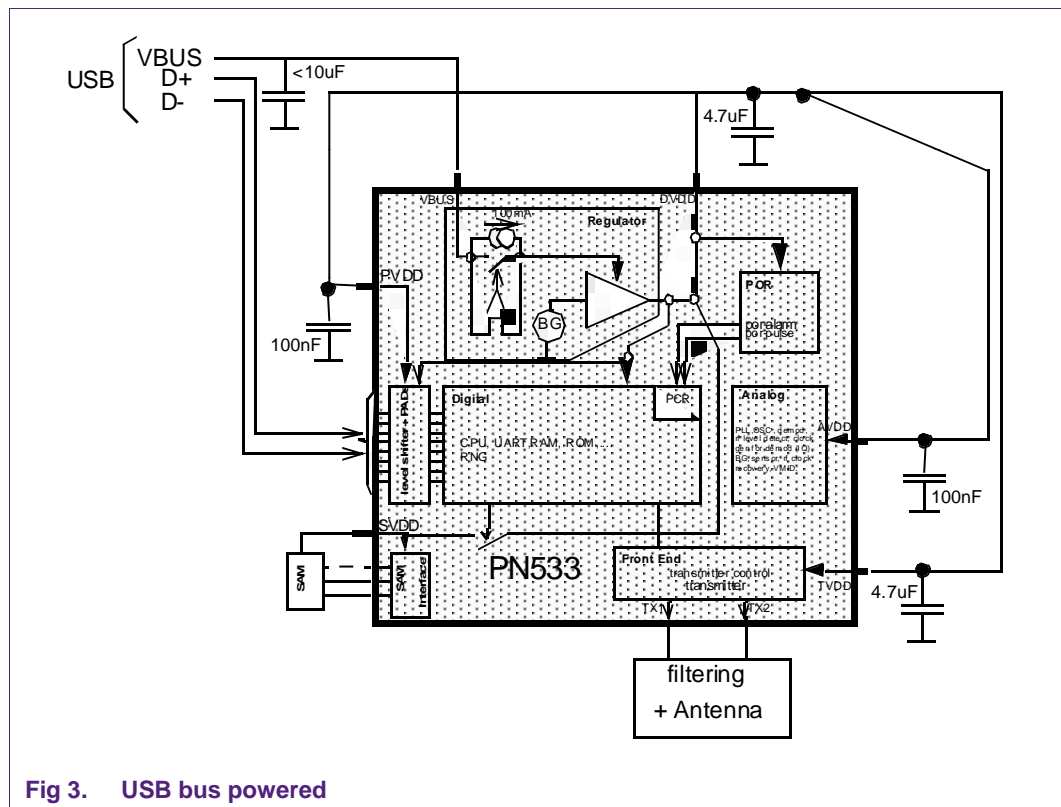
Fig 2. PN533 Block diagram

## 4. Power distribution

Here is defined power distribution scheme of the PN533 according to different system configurations.

### 4.1 USB bus powered application

The PN533 is only supplied by the USB supply (VBUS) and the PN533 internal regulator generates supply voltage for all parts. During power-up phase, inrush current is limited to less than 100mA. Total current consumption of the PN533 application when RF field is emitted is below 100mA (=80mA typical value).





### 4.2 Host powered application (single source)

Host powered application is used when the HSU interface is used; the internal regulator is then deactivated.

Here is the case when the PN533 is supplied by a single power supply source.

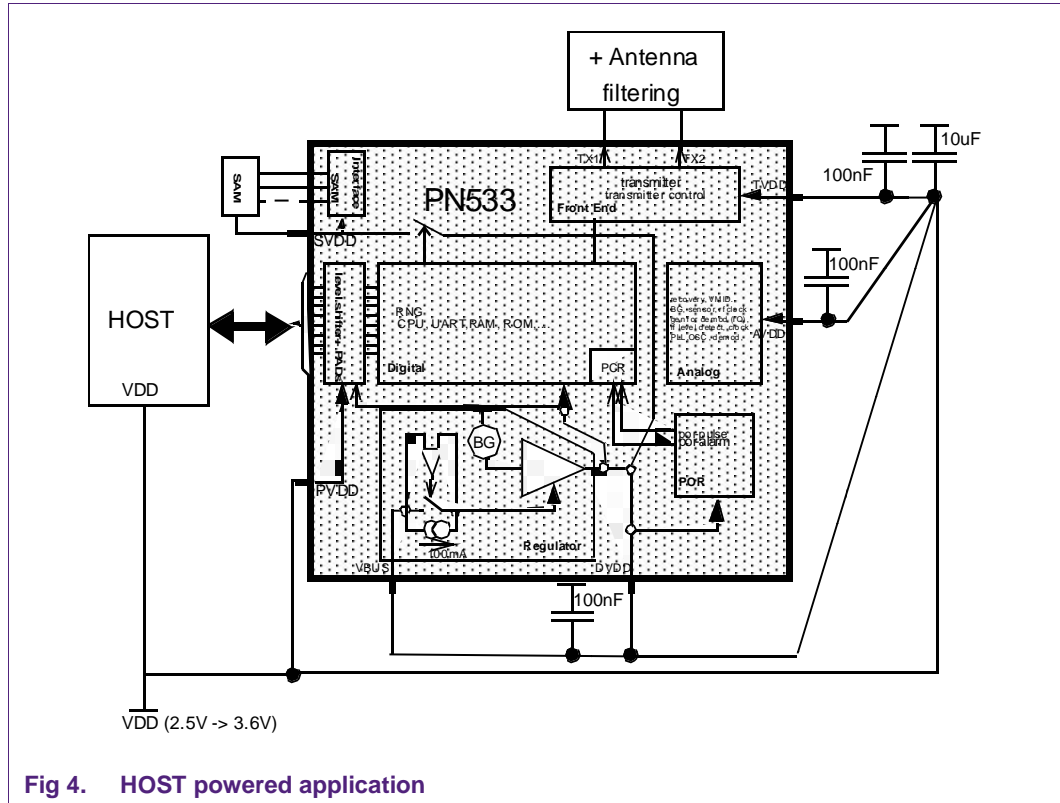


Fig 4. HOST powered application

### 4.3 Host powered application (two sources)

Host powered application is used when the HSU interface is used; the internal regulator is then deactivated.

Here is the case when the PN533 is supplied by two power supply sources.

Two different power supplies are used for the Host controller supply and the PN533 supply. The Host controller supply has to be also connected to the PVDD supply pin of the PN533.

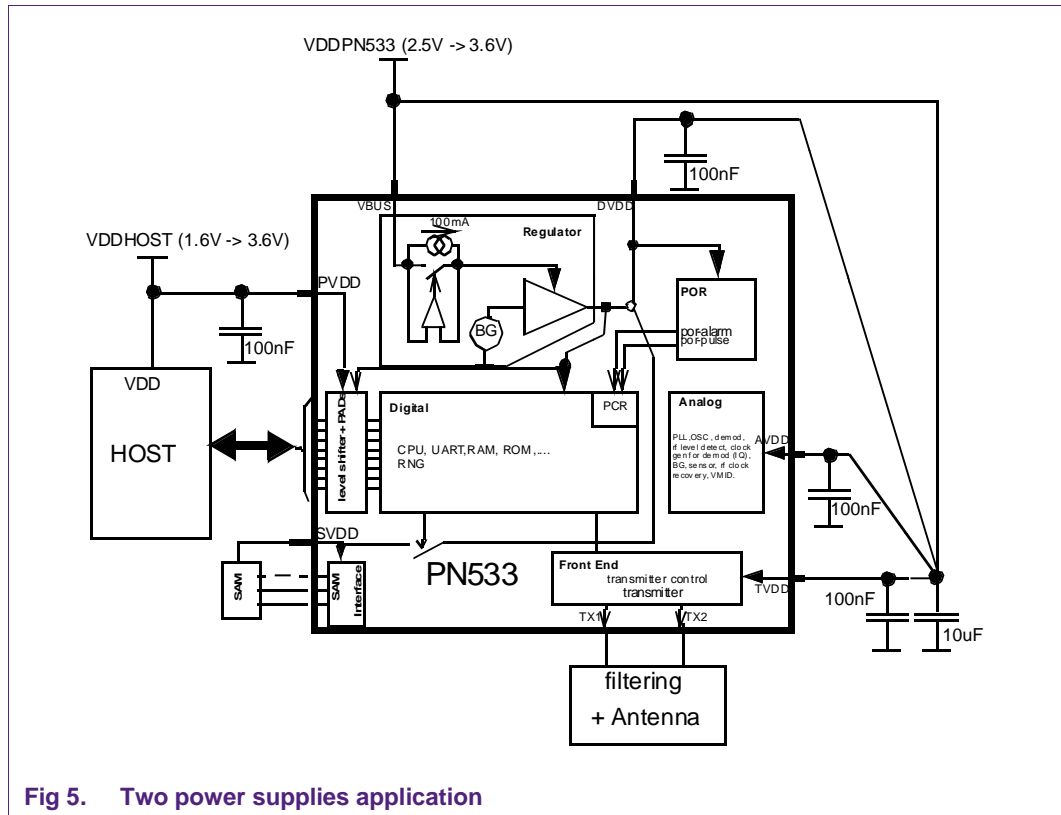


Fig 5. Two power supplies application

## 5. Interfaces with the host controller

### 5.1 Interface selection

The system host controller can communicate with the PN533 by using the USB or the HSU interface.

The pins 24 and 25 of the HVQFN40 package are used for both interfaces. They are referenced to the PVDD supply.

The pin 24 is the DM data line in USB mode or TX in HSU mode.

The pin 25 is the DP data line in USB mode or RX in HSU mode.

The selection of the interface requires a hardware configuration (Interface mode lines I0-I1) during the power up sequence of the chip.

Notice that the HSU interface is selected by connecting the I0 and I1 pins to ground, therefore the internal voltage regulator is deactivated: please refer to the host powered schematic in previous chapter for power supply connection.

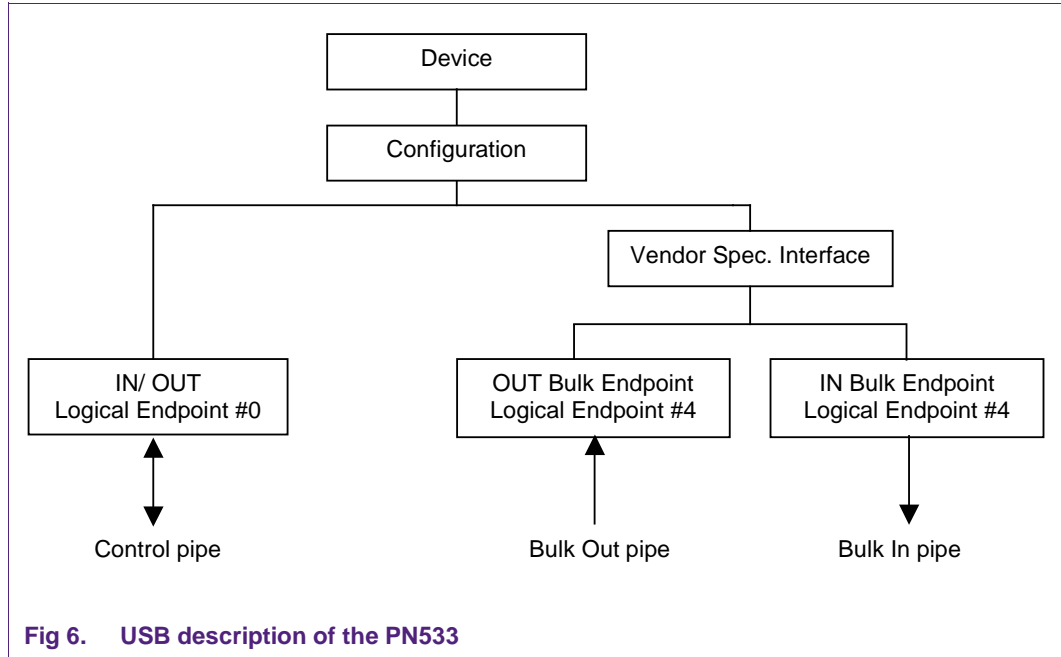
**Table 4. Interface selection**

	Interface Selection Pin	
	I0 (pin #17)	I1 (pin #18)
USB	1 DVDD	1 DVDD
HSU	0 Ground	0 Ground
Reserved for Future Use (Don't use)	0 Ground 1 DVDD	1 DVDD 0 Ground

The embedded software manages the communication with the host controller and the communication on the RF side.

## 5.2 USB interface

A USB interface modeling of the PN533 is proposed below:



The PN533 uses 2 endpoints that are part of the Vendor Specific Interface in addition to two mandatory default endpoints IN/OUT #0.

- Logical endpoint 0, **control in/out**:
  - it is needed for initializing and configuring the logical device once the device is attached and powered
  - it provides access to the device’s information and allows generic USB status and control access
  - supports control transfers
  - buffer size: 8 bytes
- Logical endpoint 4, **bulk out**:
  - this endpoint performs transfers to supply data to the PN533
  - address of the endpoint: 0x04 (logical endpoint 4, type **OUT**)
  - attributes: bulk, no synchronization, data endpoint
  - buffer size: 64 bytes
- Logical endpoint 4, **bulk in**:
  - this endpoint performs transfers to retrieve data from the PN533
  - address of the endpoint: 0x84 (logical endpoint 4, type **IN**)
  - attributes: bulk, no synchronization, data endpoint
  - buffer size: 64 bytes

- Logical endpoints 1, 2 and 3, **interrupt in:**
  - not used

Device descriptor:

In the device descriptor during the enumeration phase, the Vendor ID (offset 8 in the Device Descriptor) and the Product ID (offset 10 in the Device Descriptor) are sent by the PN533 to the USB controller.

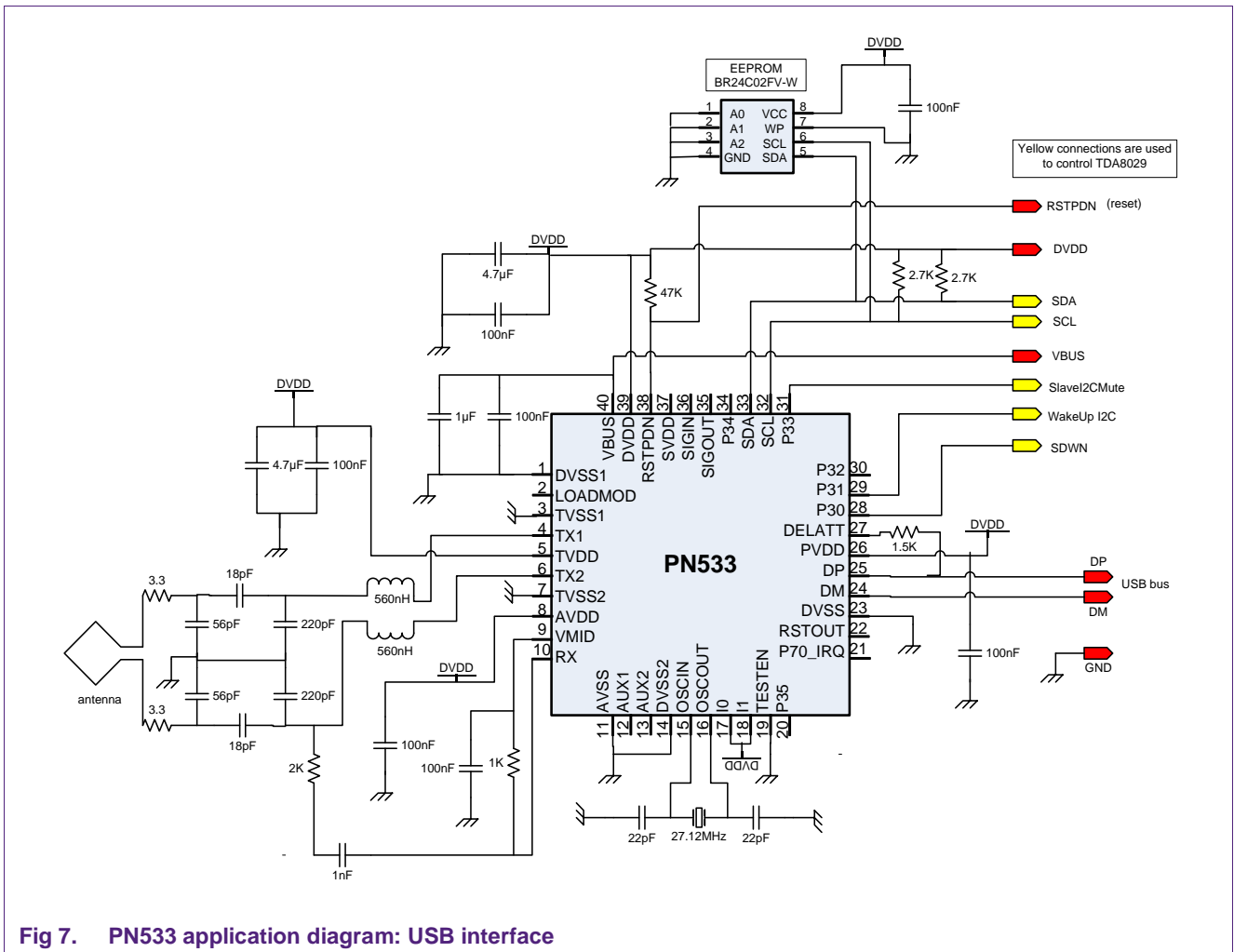
**Table 5. USB VID/PID**

PN533 PID/ViD	
PID	0x2533
VID	0x04CC

Remote WakeUp

The PN533 does not support Remote WakeUp feature.

More details about USB descriptor can be found in the User Manual (see Ref. [1] in chapter 1).



**Fig 7. PN533 application diagram: USB interface**

### 5.3 HSU interface

The HSU supports only one operational mode, which has the following characteristics:

8 data bits, 1 start bit, 1 stop bit (number is programmable), no parity.

Please refer to the datasheet for details.

#### Baud rate:

Default baud rate is 115200 bauds.

The PN533 HSU is up to 1288 kbauds. The HSU speed is changeable with SetSerialBaudrate command (command code 10h, parameter = baudrate).

**Table 6. HSU baud rates**

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 PN533 will switch to the new baud rate 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
```

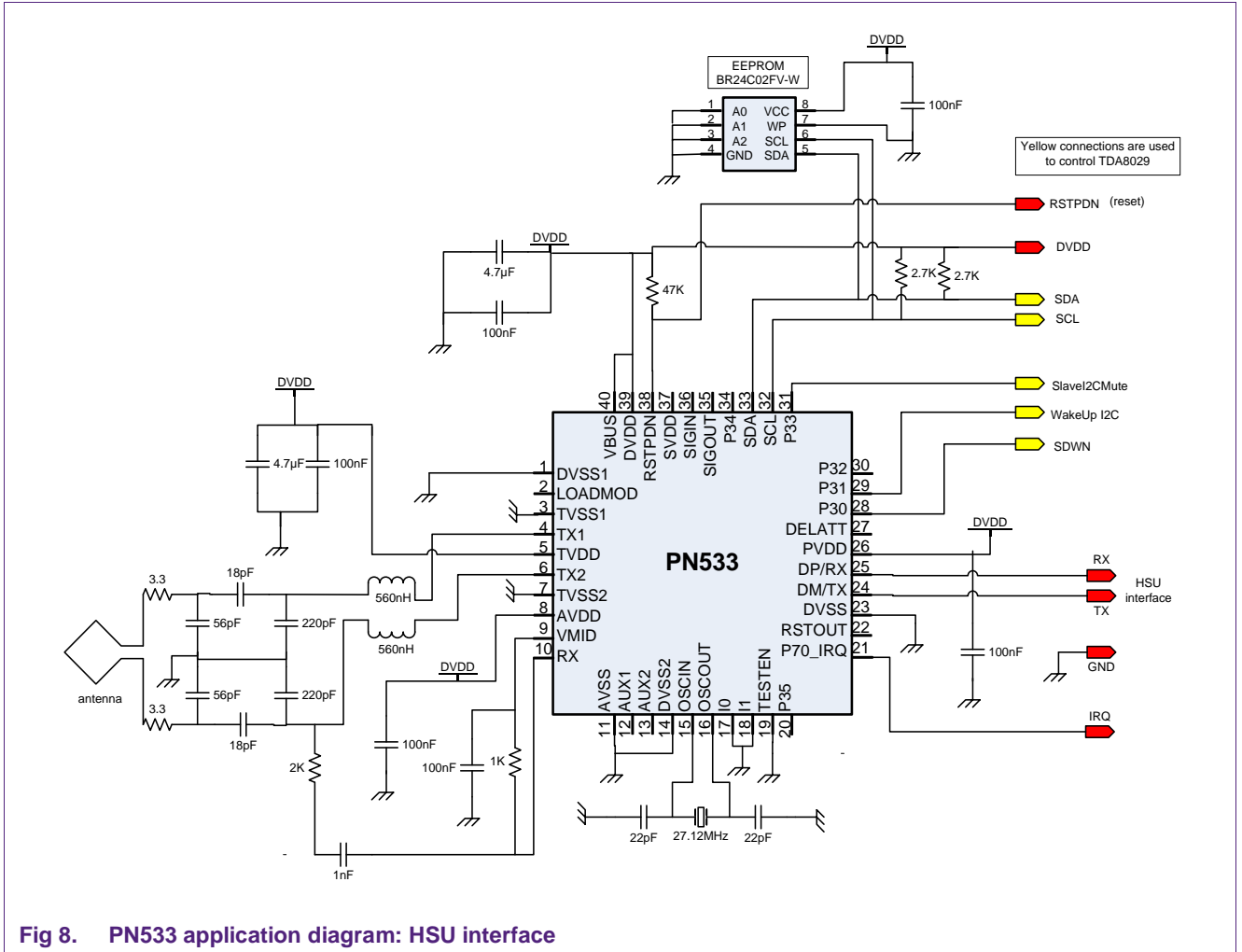


Fig 8. PN533 application diagram: HSU interface

## 6. USB 2.0 compliancy

To reach the USB 2.0 Low/Full speed compliancy, a specific routine needs to be implemented in the PN533 driver.

This chapter describes:

- Why these functions were implemented?
- How it is implemented?

### Current limitation of the firmware:

With the current PN533, it is possible to corrupt the USB Descriptors information.

If some large exchanges of data are performed, it is possible to overwrite some USB data which are necessary during a power management state transition of the computer.

For example, the following case could occur:

**Table 7. Possible issue for thePN533**

Step n°	Description
1	Plug the PN533 device on a Desktop computer. The PN533 device will be properly enumerated.
2	Perform an ePassport reading. The PN533 will read lot of data from the ePassport. This reading can corrupt the USB information.
3	Restart the computer. The PN533 is connected to a desktop, then the VBus line will stay at 5V during this restart. The PN533 will not be reset. It will keep its current settings.
4	During enumeration of the PN533, the computer will ask all USB Descriptor information to the PN533.  If the USB Descriptor was corrupted during the Step n°2, the PN533 will not be properly enumerated.

### Modification to implement in the driver for fixing this issue:

USB Descriptor data are stored in a specific location of the PN533's XRam.

When the computer wants to go to suspend or shutdown state, a routine in the PCSC driver shall store the proper USB data into the PN533's XRam. This writing operation is performed in the driver Exit point. It uses the WriteRegister command for performing this writing.

USB Descriptor data are stored in the PN533's XRam at the following address: [0x0019; 0x0055].

When the driver will detect a modification power state of the device (see in Msdn the enum Wdf\_Power\_Device\_State), the driver will issue the following WriteRegister commands:

- 08 00 19 XX 00 1A XX 00 1B XX 00 1C XX;  
The driver will write the proper USB data in XRam between [0x0019; 0x001C].



- 08 00 1D XX 00 1E XX 00 1F XX 00 20 XX;;  
The driver will write the proper USB data in XRam between [0x001D; 0x0020].
- [...]  
The driver will write the proper USB data in XRam between [0x0021; 0x004F].
- 08 00 4E XX 00 4F XX 00 50 XX 00 51 XX;  
The driver will write the proper USB data in XRam between [0x004E; 0x0051].
- 08 00 52 XX 00 53 XX 00 54 XX 00 55 XX;  
The driver will write the proper USB data in XRam between [0x0052; 0x0055].

## 7. PN533 host link protocol

The protocol used on host link is fully described in the PN533 User manual (see Reference #1 in chapter 1)

### 7.1 Standard frame

A basic exchange consists in a command frame sent by the host controller to the PN533, an ACK frame sent by the PN533 as soon as the command is correctly received, and a response frame to be read by the host controller (polling mechanism or use of IRQ).

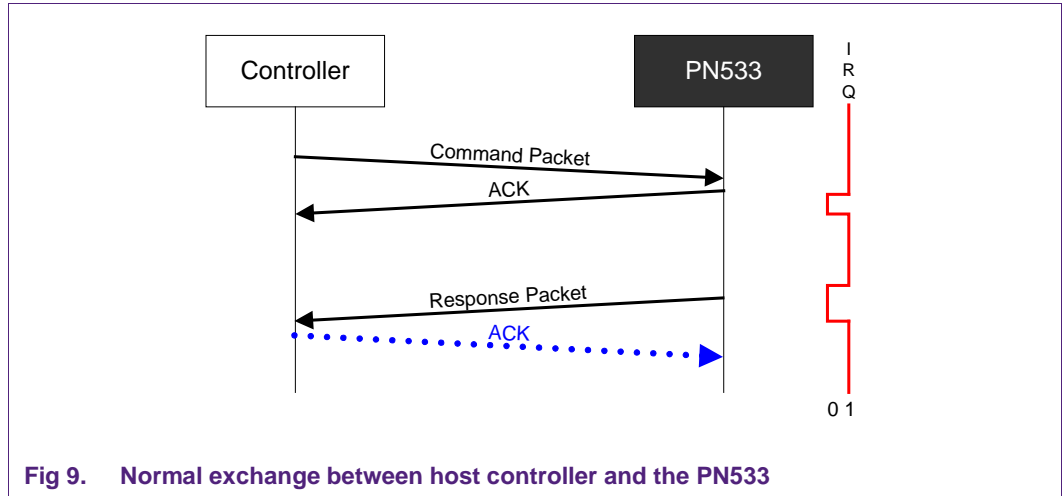


Fig 9. Normal exchange between host controller and the PN533

Command and response frame structure is shown in figure 10.

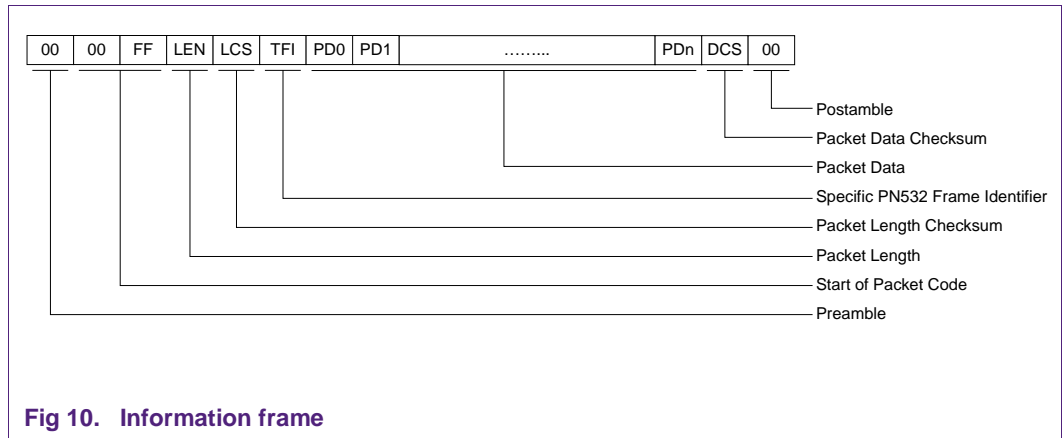


Fig 10. Information frame

- **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 PN533 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 PN533
- **D5h** in case of a frame from the PN533 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 below.

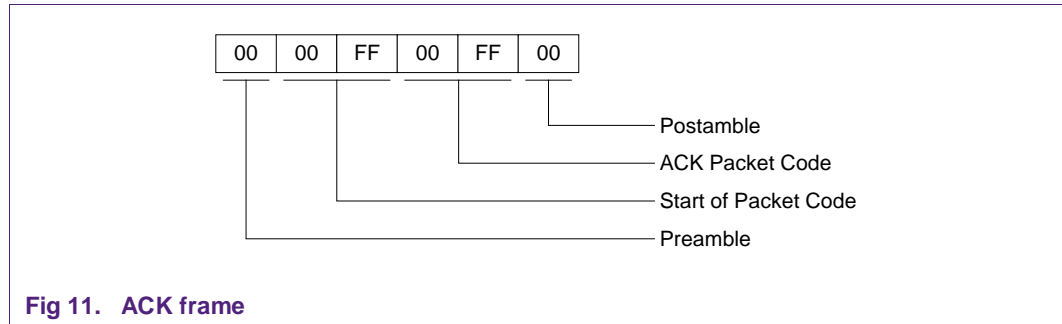


Fig 11. ACK frame

List of available commands (PD0 byte) is provided in this application note.

## 7.2 Extended frame

The information frame has an extended definition allowing exchanging more data between the host controller and the PN533.

In the firmware implementation of the PN533, 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:

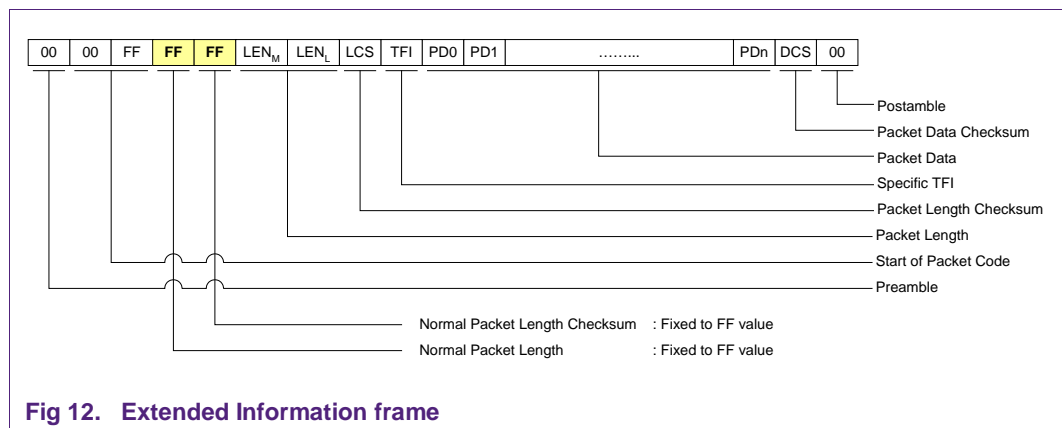


Fig 12. Extended Information frame

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<sub>M</sub>** (MSByte) and **LEN<sub>L</sub>** (LSByte) with:

$$\text{LENGTH} = \text{LEN}_M \times 256 + \text{LEN}_L \text{ coding the number of bytes in the data field (TFI and PD0 to PDn)}$$

- **LCS**      1 Packet Length Checksum LCS byte that satisfies the relation:  
Lower byte of  $[\text{LEN}_M + \text{LEN}_L + \text{LCS}] = 0x00$ ,
- **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 PN533 always uses the suitable type of frame, depending on the length (Normal Information Frame for frame  $\leq 255$  bytes and Extended Information Frame for frame  $> 255$  bytes).

Preamble and Postamble:

In the way from the host controller to the PN533, both the preamble and postamble fields may have a length different from one byte (0 to n) and the value has no impact on the frame processing.

### 7.3 NACK frame

The specific NACK frame is used for the synchronization of the packets.

This frame is used only from the host controller to the PN533 to indicate that the previous response frame has not been successfully received, then asking for the retransmission of the last response frame from the PN533 to the host controller.

NACK frame:

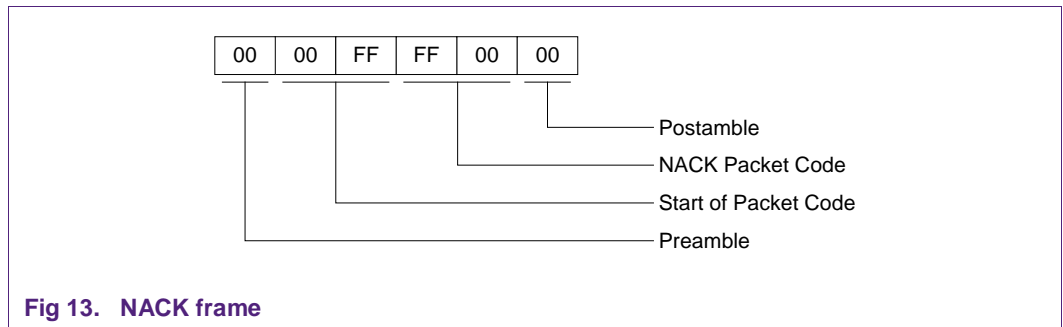


Fig 13. NACK frame

### 7.4 Error frame

The syntax error frame is used to inform the host controller that the PN533 has detected an error at the application level.

Error frame:

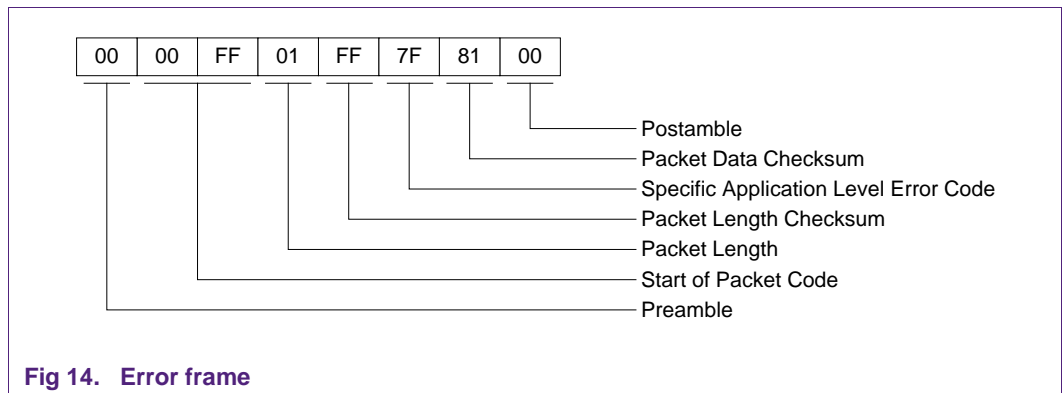


Fig 14. Error frame

### 7.5 USB communication details

The USB device interface of the PN533 is built around:

- A Control Endpoint 0 (8 bytes IN/ 8 bytes OUT),
- A BULK IN Endpoint (64 bytes),
- A BULK OUT Endpoint (64 bytes).

The command is sent by the system controller over the BULK OUT endpoint and the response is received in the BULK IN endpoint.

The host polls the BULK IN after BULK OUT has been sent.

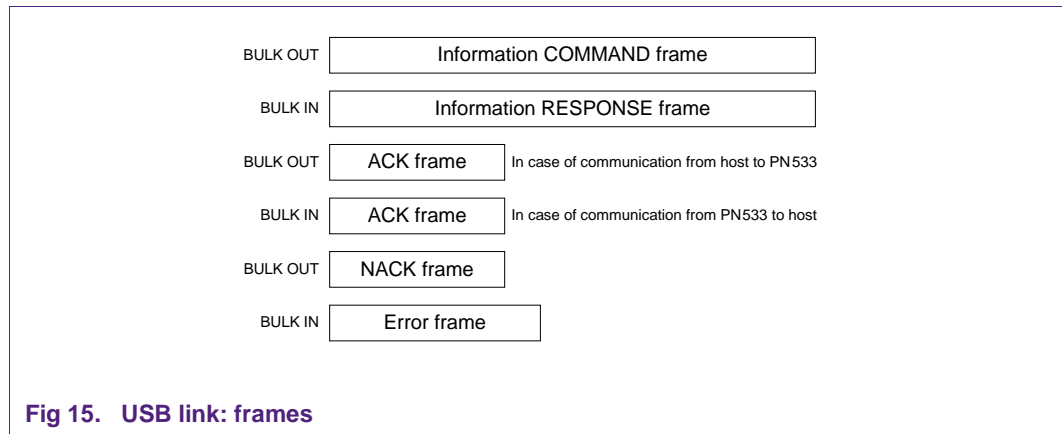


Fig 15. USB link: frames

The figure below depicts the normal scheme of communication with the USB:

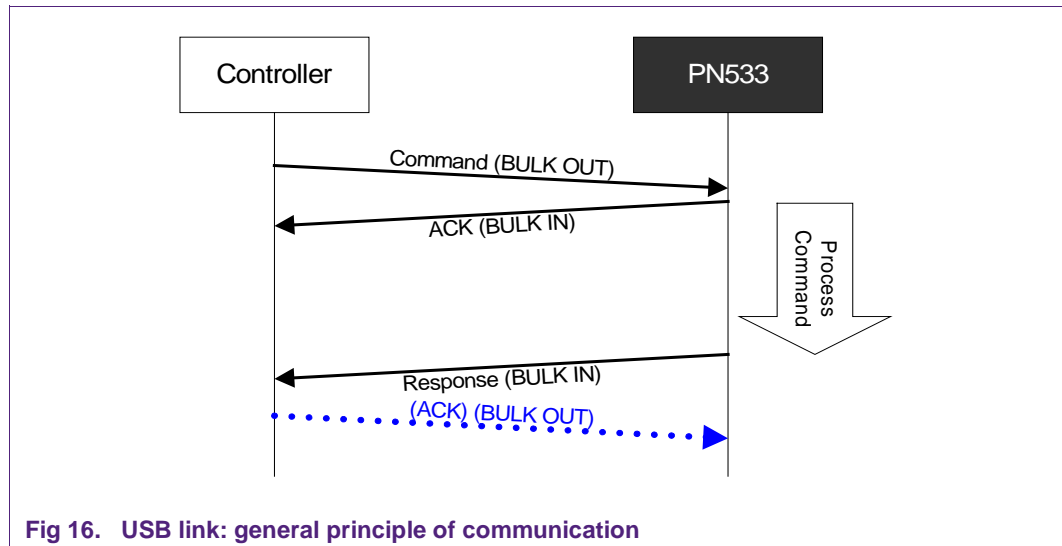


Fig 16. USB link: general principle of communication

The PN533 has to respond to the incoming command frame within 15 ms ( $T_{Max Response Time}$ : delay between the command frame and the ACK frame).

In the case the host controller does not detect an ACK frame within these 15 ms, the host controller should resend the same command frame.

## 8. I<sup>2</sup>C master interface

The I<sup>2</sup>C master interface of the PN533 is compliant with the I<sup>2</sup>C bus specification.

The PN533 is configured as master and is able to communicate with an EEPROM (address 0xA0) and a smart card reader (address 0x50).

- Serial data line (SDA), has to be connected to pin 33 of the PN533
- Serial clock line (SCL), has to be connected to pin 32 of the PN533.

External pull-up resistors have to be connected to DVDD supply (=3.3V).

### 8.1 I<sup>2</sup>C EEPROM

The PN533 can communicate with an EEPROM. Information like RF settings and USB descriptor can be embedded in it.

EEPROM transactions are performed at a frequency of 400 kHz on the I<sup>2</sup>C link.

When EEPROM transactions are performed, the TDA connected to the I<sup>2</sup>C bus has to be in Shut-down mode.

Example of I<sup>2</sup>C bus compatible serial EEPROM = BR24C02FV-W (256 Bytes).

- **Sequence of reading information on I<sup>2</sup>C EEPROM to perform enumeration:**

Step	
1	When the PN533 is supplied, it shall check if USB information is contained in the EEPROM.
2	During this step, PN533 will look for tags: 0x02, 0x03, 0x04 and 0x05 (see data organization in next chapter). Once a tag is discovered, the firmware will read the content of its associated block and save it in RAM. If the 4 blocks have been read correctly in EEPROM, this information is used as initial values after power-on, otherwise ROM code information is used as initial values after power-on.
3	PN533 shall active the USB_SoftConnect switch. The voltage of D+ will be set to '1' (see next figure).
4	The host shall detect this voltage change and it shall send a request to ask for USB device information for the enumeration phase.
5	The PN533 has to fill the IN Bulk Endpoint with its USB information.
6	The Host has to read the content of the IN Bulk Endpoint. After this reading, the PN533 should be correctly enumerated.

**Fig 17. Sequence of reading information on I<sup>2</sup>C EEPROM to perform enumeration**

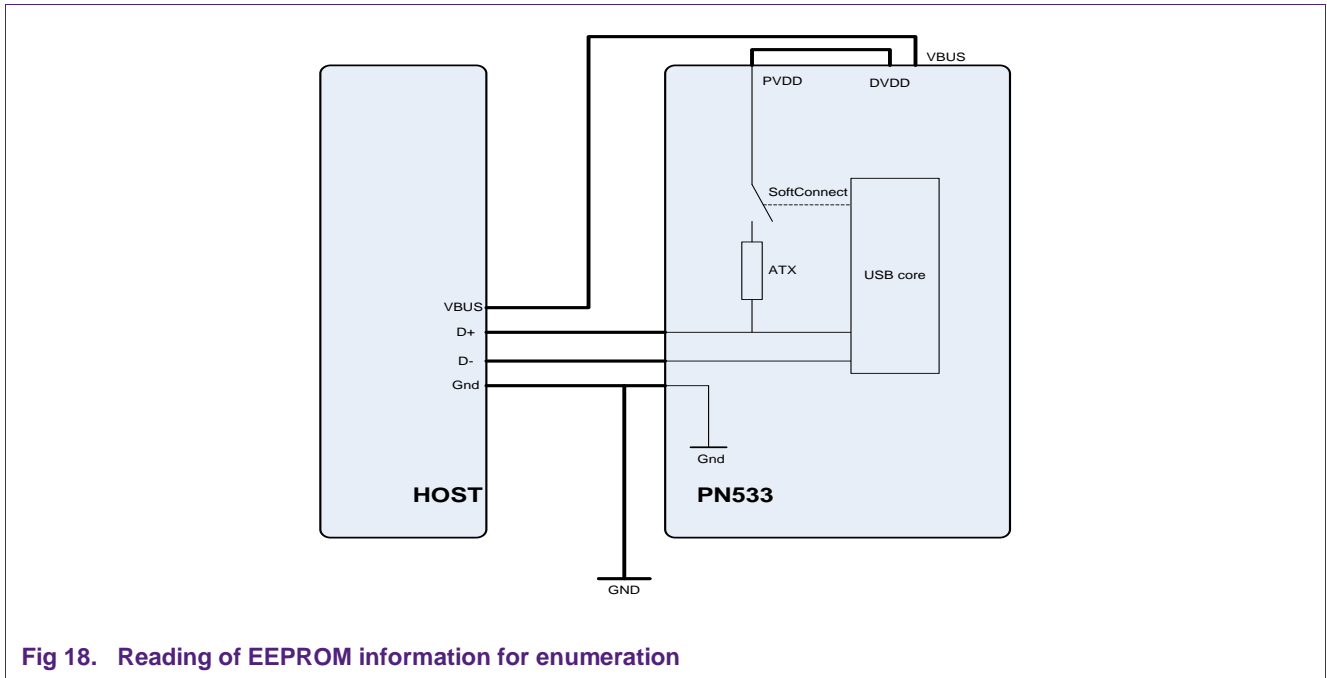


Fig 18. Reading of EEPROM information for enumeration

• **EEPROM reading/writing**

The two commands ReadRegister and WriteRegister (see user manual Ref. #1 of chapter 1) are used to read and write data into a serial I<sup>2</sup>C EEPROM.

The addresses of EEPROM registers are 0xA0 XX.

Example of writing four bytes (0x01, 0x02, 0x03, 0x04) into EEPROM (slave address: A0) at the address 0x80:

The frame from the host controller to the PN533 is:

D4	08	A0	80	01	A0	81	02	A0	82	03	A0	83	04
----	----	----	----	----	----	----	----	----	----	----	----	----	----

and the frame returned by the PN533 is:

D5	09
----	----

The PN533 takes in charge I<sup>2</sup>C master protocol:

Write I<sup>2</sup>C Start condition and Slave Address on I<sup>2</sup>C bus: 'I<sup>2</sup>C START CONDITION', A0h,  
 Write EEPROM Address where to write, and write the data: 80h, 01h, 02h, 03h, 04h  
 Write I<sup>2</sup>C Stop condition: 'I<sup>2</sup>C STOP CONDITION'

Example of reading four bytes from EEPROM (slave address: A0) at the address 0x80:

The frame from the host controller to the PN533 is:

D4	06	A0	80	A0	81	A0	82	A0	83
----	----	----	----	----	----	----	----	----	----

And the one returned by the PN533 is:

D5	07	00	01	02	03	04
----	----	----	----	----	----	----

The PN533 takes in charge I<sup>2</sup>C master protocol:

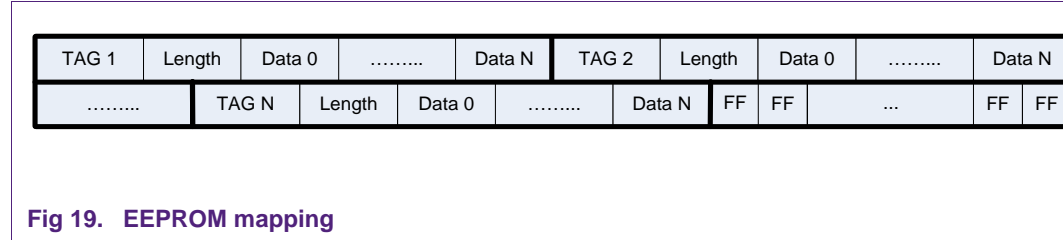
Write I<sup>2</sup>C Start condition and Slave Address on I<sup>2</sup>C line: 'I<sup>2</sup>C START CONDITION', A0h,  
 Write EEPROM Address, and the number of bytes read: 80h, 00h, 04h

'I<sup>2</sup>C START CONDITION', A1h, Read ← 01h, 02h, 03h, 04h from EEPROM  
 Write I<sup>2</sup>C Stop condition: 'I<sup>2</sup>C STOP CONDITION'

**8.1.1 EEPROM data organization**

To organize data in the EEPROM, we use the TLV system. Each group of data is preceded by its tag and the length of the data.

The EEPROM mapping is shown below:



**Fig 19. EEPROM mapping**

- The free space of EEPROM may be set to 0xFF.
- At least one 0xFF has to be present at the end of EEPROM mapping.
- Each tag has to be present only one time.
- The order of tags in the EEPROM is not important.
- Each and every tag # 2, 3, 4 and 5 has to be present.
- If one tag is missing or there is one error in the mapping of the EEPROM, then the information embedded in EEPROM is not used.

**8.1.2 List of EEPROM tags**

Tags allow the PN533 to identify which data are stored after it in the EEPROM.

The list of EEPROM tags is the following:

- 0x01 ⇔ RF settings block,
  - 106kbps communication Type A,
  - 212kbps/424kbps communication for FeliCa,
  - 106/212/424/847kbps communication Type B,
  - 212/424/847kbps communication Type A,
- 0x02 ⇔ Fixed USB descriptor block,
  - Device Descriptor,
  - Configuration Descriptor,
  - Interface Descriptor,
  - OUT endpoint Descriptor,
  - IN Endpoint Descriptor.
- 0x03 ⇔ Device ID String Descriptor,
- 0x04 ⇔ Manufacture ID String Descriptor,
- 0x05 ⇔ Configuration mode 0 String Descriptor.

Each block description can be found in the User Manual (see Ref. #1 of chapter 1).



## 8.2 I<sup>2</sup>C link to the TDA8029

The maximum SCL frequency used for transactions with the TDA8029 is less than 50 kHz.

Pin 1 of the TDA8029 has to be connected to ground to enable I<sup>2</sup>C interface.

Pin 25 is connected to VDD to enable Energy Saving Mode (ESM).

- SDA is connected to pin 32 of the TDA8029.
- SCL is connected to pin 2 of the TDA8029.

In addition to I<sup>2</sup>C bus specification, we shall use three other lines to manage Energy Saving Mode (ESM) mechanism of the TDA8029. ESM is enabled by pin #25 (P26) of the TDA8029 (see Ref. #14 of chapter 1):

- WakeUpSlave line is used to wake up the TDA8029. It has to be connected between INT1 (pin 30 of the TDA8029) and P31 (pin 29) of the PN533.
- SlaveI<sup>2</sup>C Mute line is used by the TDA8029 to indicate to the host controller either that it is ready to receive a command frame, or to send the corresponding answer, or to signal a hardware event. It has to be connected between pin 24 of the TDA8029 and P33 (pin 31) of the PN533.
- Shut-down line is used for entering in the TDA8029 shut-down mode. This mode is set when the TDA8029 SDWN\_N pin is set to 0. The only way to leave shut-down mode is when pin SDWN\_N is set to 1. The SDWN\_N pin has to be connected to P30 (pin 28) of the PN533.

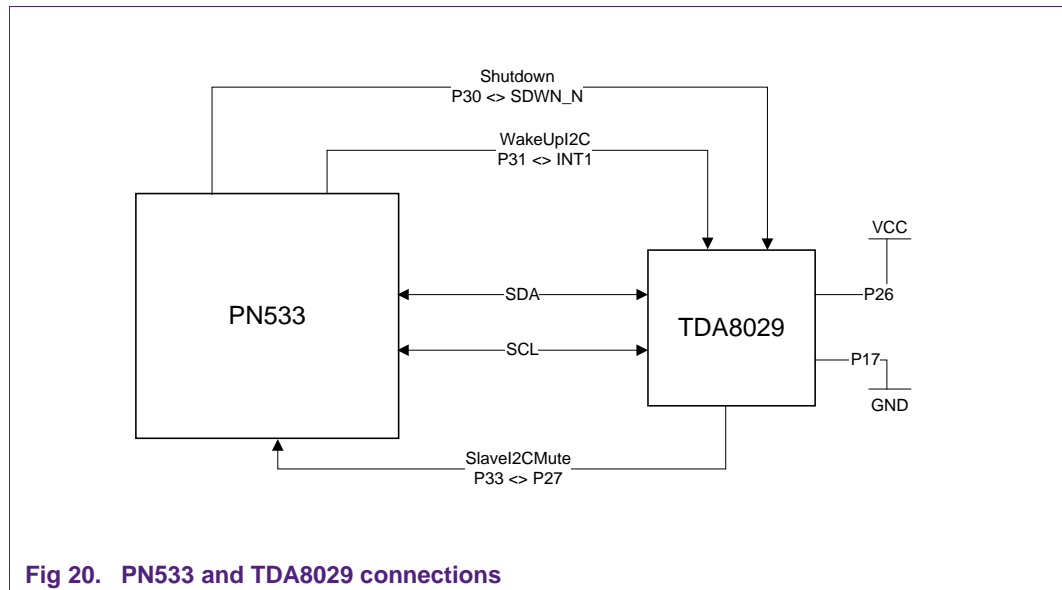


Fig 20. PN533 and TDA8029 connections

Warning: The contact reader TDA8029 is used only as a SAM reader; it cannot be EMVCo approved when Energy Saving Mode is used (clock-stop mode is not allowed).

The PN533 contactless application can be EMVCo approved.

When EEPROM transactions are performed, the TDA8029 has to be in Shut-down mode (SDWN pin has to be set to 0) in order to not disturb the TDA8029. At the end of transactions with the EEPROM, the TDA8029 has to leave shut-down mode (SDWN pin

has to be set to 1). This is done by using the command SetParameters with flag bit fTDApowered (see user manual Ref. #1 of chapter 1).

Input:

D4	12	Flags
----	----	-------

The command AlparCommandForTDA (see user manual Ref. #1 of chapter 1) is defined to access to the TDA8029 via the I<sup>2</sup>C bus.

The I<sup>2</sup>C transactions use the 'ALPAR' protocol described in the TDA8029 application note (see Ref. #14 of chapter 1).

The command AlparCommandForTDA is used to send an ALPAR command to the TDA8029, and the response is a status byte + the ALPAR response (see user manual Ref. #1 of chapter 1).

- **Examples of use are described below:**

Example of sending of 'Check Card presence' ALPAR command to TDA (slave address for command: 50h):

Input:

D4	18	ALPAR STRUCTURE IN = 60h, 00h, 00h, 09h, LRC
----	----	---

The PN533 takes in charge I<sup>2</sup>C master protocol:

Write I<sup>2</sup>C Start condition and Slave Address on I<sup>2</sup>C line: 'I<sup>2</sup>C START CONDITION', 50h,

Write ALPAR Command: 60h, 00h, 00h, 09h, LRC

Write I<sup>2</sup>C Stop condition: 'I<sup>2</sup>C STOP CONDITION'

Example of reception of 'Check Card presence' ALPAR response from the TDA (slave address for response 51h):

The PN533 takes in charge I<sup>2</sup>C master protocol:

Write I<sup>2</sup>C Start condition and Slave Address on I<sup>2</sup>C line: 'I<sup>2</sup>C START CONDITION', 51h,

Read from TDA ← 60h, 00h, 01h, 09h, 01h, LRC when card is present.

Write I<sup>2</sup>C Stop condition: 'I<sup>2</sup>C STOP CONDITION'

Output:

D4	19	Status	ALPAR STRUCTURE IN = 60h, 00h, 00h, 09h, LRC
----	----	--------	---

8.3 Typical application schematics

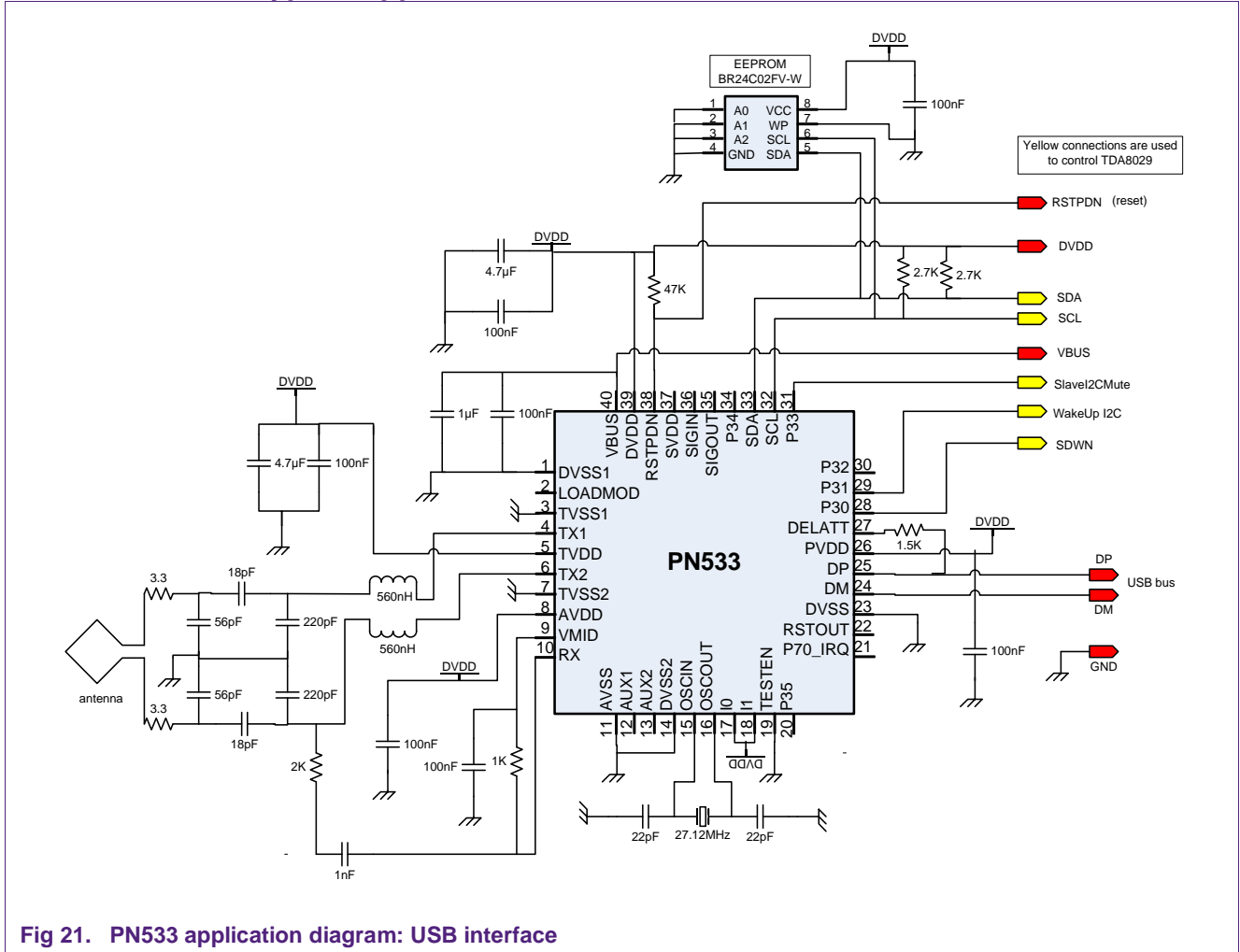


Fig 21. PN533 application diagram: USB interface

Antenna and RF design guide AN100720 gives detailed information about external components to be used with RF transmitter and receiver.

It is an USB bus powered application.

VBUS is the power supply provided by USB link.

DVDD supply is output from PN533 internal regulator and it has to be externally connected to other supply pins AVDD, TVDD and PVDD.

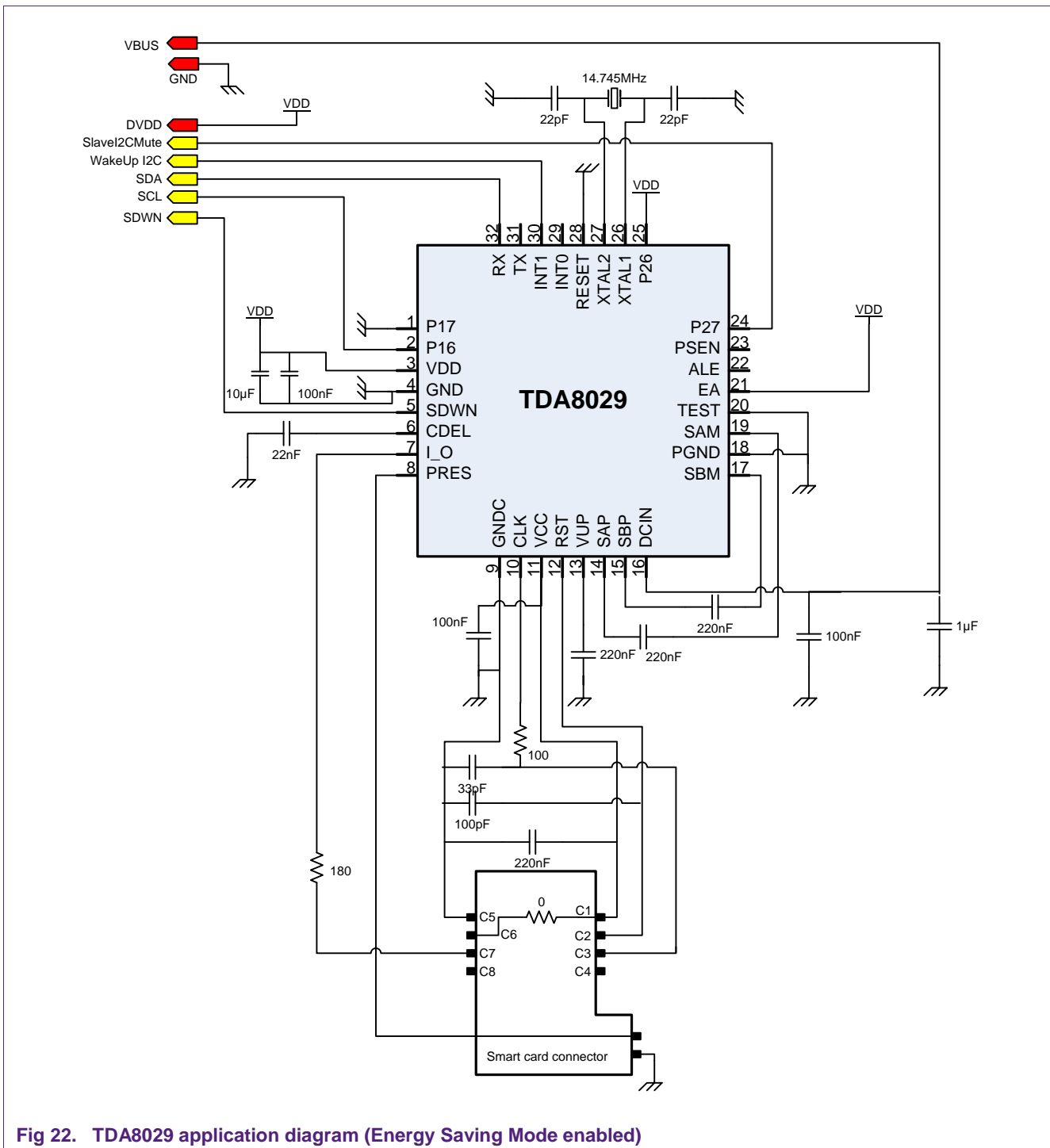


Fig 22. TDA8029 application diagram (Energy Saving Mode enabled)

DVDD output (=3.3V) from the PN533 is used as VDD supply for the TDA8029.

VBUS supply (>4.2V) from the USB bus is used as DCIN supply for the TDA8029.

DVDD cannot be used as DCIN supply because current consumption may be as high as 250mA.

## 9. PayPass/EMVCo compliancy

### 9.1 Protocol

The embedded firmware command `InActivateDeactivatePaypass` (see user manual Ref. #1 of chapter 1) allows compliancy with Paypass v1.1 and EMVCo v2.0 specifications with respect to the protocol. This command fully takes in charge Polling / Anti-Collision / Activation and PICC removal procedure with respect to the PayPass and EMVCo specifications. Complete description of these procedures can be found in the user manual.

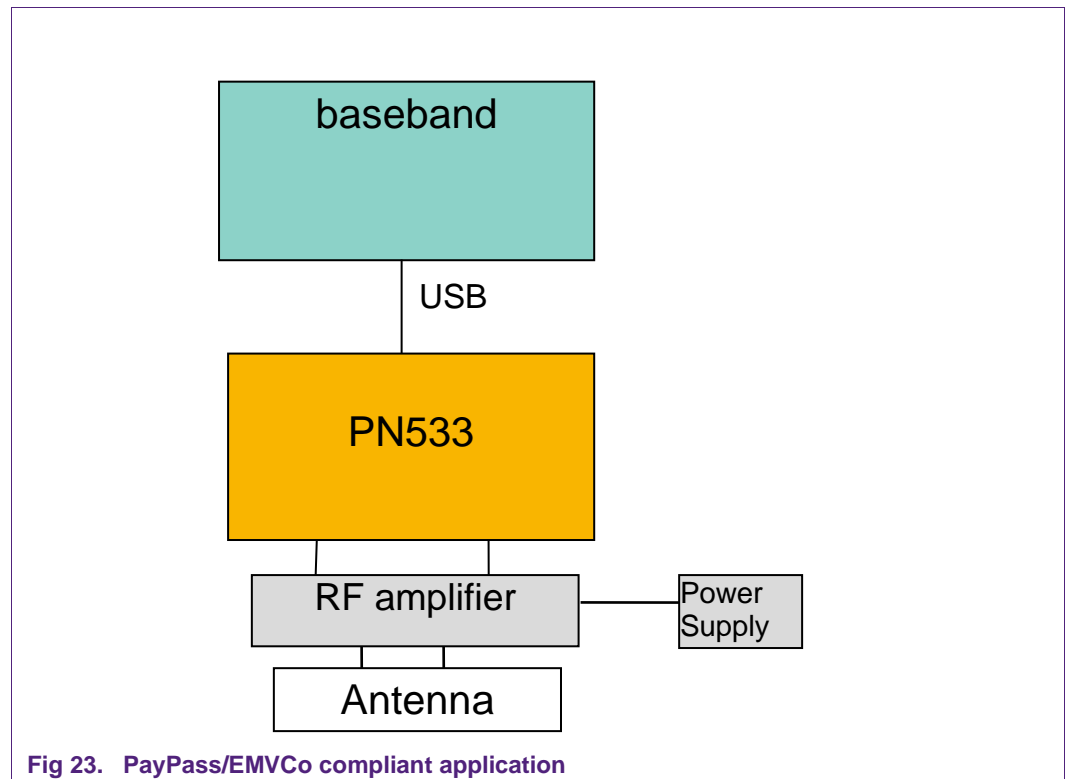
### 9.2 RF power and signal interface

The Paypass and EMVCo contactless specifications (Ref. #12 and #13 of chapter 1) define an operating volume where a minimum magnetic field strength has to be provided. This field strength cannot be achieved in all the operating volume by the transmitter of the PN533 because it is powered by a typical supply voltage of 3.3V only.

An external RF amplifier is needed. This RF amplifier is made of few discrete components and it is connected to TX1, TX2 outputs of the PN533 transmitter instead of the usual matching/ tuning antenna network.

An additional power supply is needed to power this external RF amplifier. Depending on the application environment, a minimum supply voltage of 6V may be needed to fulfil the PayPass/EMVCo specification. Typical current consumption is around 150mA. The implementation of the RF amplifier is fully described in the application note AN142520 (see Ref #7 of chapter 1).

Here is a block diagram of a complete PayPass/EMVCo compliant application:



Here is the schematic of the additional RF amplifier:

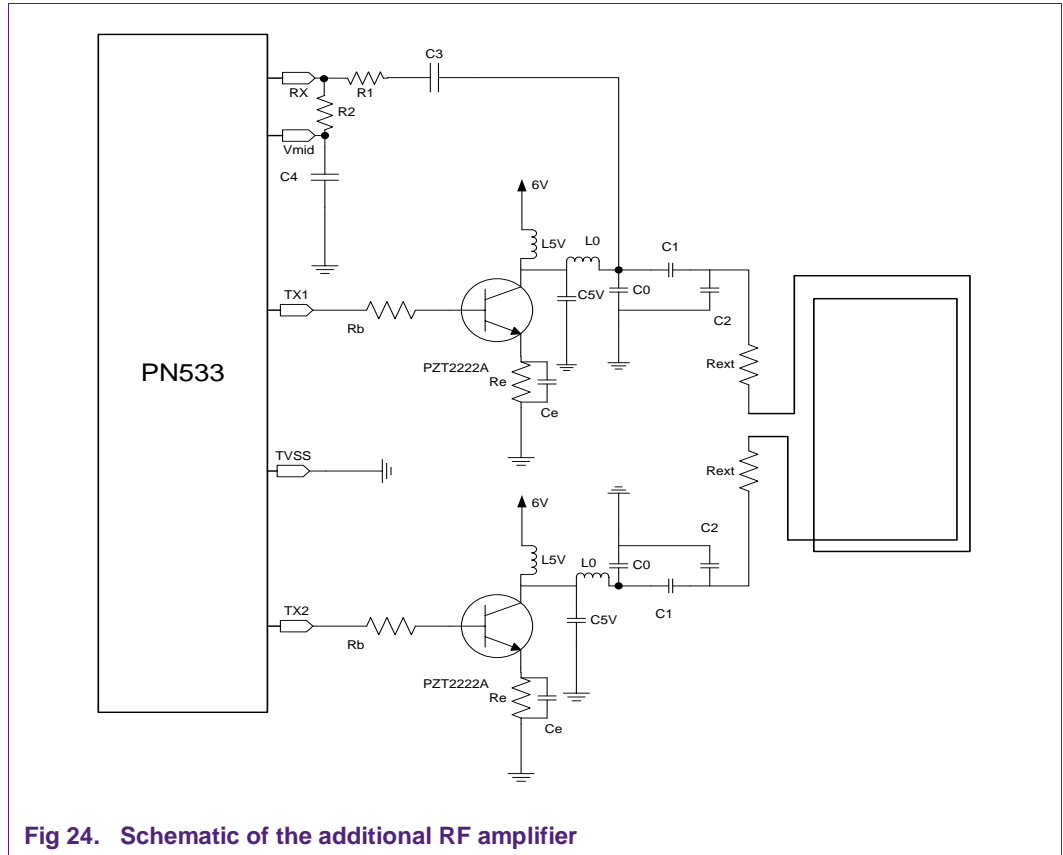


Fig 24. Schematic of the additional RF amplifier

Here is the list of components of this additional RF amplifier:

Components	Values
C <sub>0</sub>	180pF
C <sub>1</sub>	68pF
C <sub>2</sub>	280pF
C <sub>3</sub>	1nF
C <sub>4</sub>	100nF
C <sub>5V</sub>	91pF
C <sub>e</sub>	22pF
L <sub>0</sub>	560nF
L <sub>5V</sub>	1.5uH, Q>=20 @ 13.56MHz
R <sub>1</sub>	3.3kR
R <sub>2</sub>	1kR
R <sub>b</sub>	22R
R <sub>e</sub>	18R
T <sub>1</sub> , T <sub>2</sub>	PZT2222A

Fig 25. List of components of the additional RF amplifier

## 10. Radio Frequency Activation

The command `InQuartetByteExchange` (see user manual Ref. #1 of chapter 1) is a specific command used to support basic data exchanges between the PN533 and a RFA tag.

When using this command, it is assumed that the card has been first activated by an `InListPassiveTarget` command.

The command `InQuartetByteExchange` is complementary of the `InDataExchange` command or the `IncommunicateThru` command.

The `InQuartetByteExchange` works like the `InCommunicateThru` command, but the `InQuartetByteExchange` is able to manage different type of input data (coming from the PICC) as quartets (Ack / Nack), data bytes or time out.

The host has to specify the type of response that is expected back after sending the command.

The input parameters are interpreted by the PN533 to execute a RFA exchange. The PN533 sends the command and waits for the answer with a default timeout value of 51.2 ms. This timeout value can be changed by using the command `RFConfiguration`.

Complete description of these procedures can be found in the user manual.

## 11. NFC Secure

NFC Secure Standard specifies an application layer on top of NFCIP-1 interface and protocol. The NFC Secure layer protects the communication on the RF interface against eavesdropping and data modification (at the NFCIP-1 message level) by cryptographic methods. The protocol can be activated either by the Initiator or by the Target.

This layer is implemented on the application level. The PN533 provides only a pipe of exchange; all the cryptographic methods are implemented on the application layer.

During a secured DEP, the 5<sup>th</sup> bit of the NFCIP-1 Data Exchange Protocol PFB byte shall be set to 1. The rest of the bits of the PFB byte shall be identical to the ones used in the Information pdu and are used in the same way.

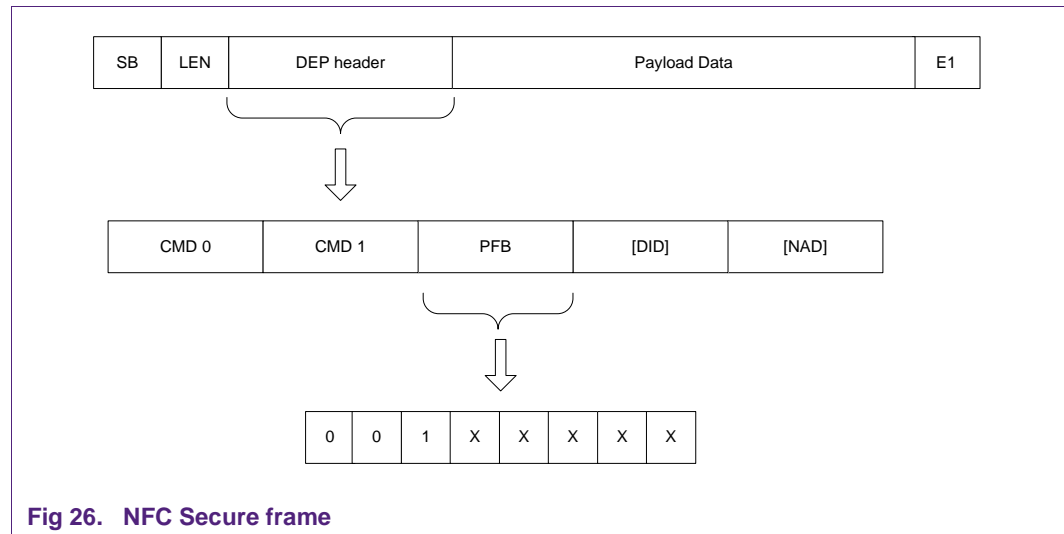


Fig 26. NFC Secure frame

See examples of Shared Secret Communication in the User Manual (see Ref. #1 in chapter 1).

For enabling the NFC Secure layer, the Host has to set the fSecure bit of the SetParameters command to 1.

In case of Shared Secret Service activated by the Initiator, the Initiator-Host activates the shared secret service with the InDataExchange command used with the 5th bit of Tg byte equal to 1, then the PN533 sets the 5th bit of PFB to 1.

If the 5th bit of PFB is set to 1, then the communication is secured and the status of the GetData or InDataExchange response is set to 0x20 (if the RF communication was successful).

Then the Target-Host is informed by the status of the GetData response that the communication has to be secure and it answers to the Initiator-Host with the SetDataSecure command.

The command TgSetDataSecure can be used by the PN533 configured as target for initiating a Secret Exchange Protocol (SEP), then the PN533 sets the 5th Bit of PFB to 1.

The command TgSetMetaDataSecure is used in case of the PN533 configured as target for Secure Exchange Protocol (SEP) if the overall amount of data to be sent cannot be transmitted in one frame (more than 262 bytes).



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.

## 12. The PN533 commands

### 12.1 Global view

The PN533 can be accessed by using directly the firmware API described in the user manual (see Ref. #1 in chapter 1) and in the following pages (interface B in the figure below). 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 below).

Note: PN51x, PNxxx, RCxxx represents other NXP NFC products. PN53x represents PN532 or PN533 product.

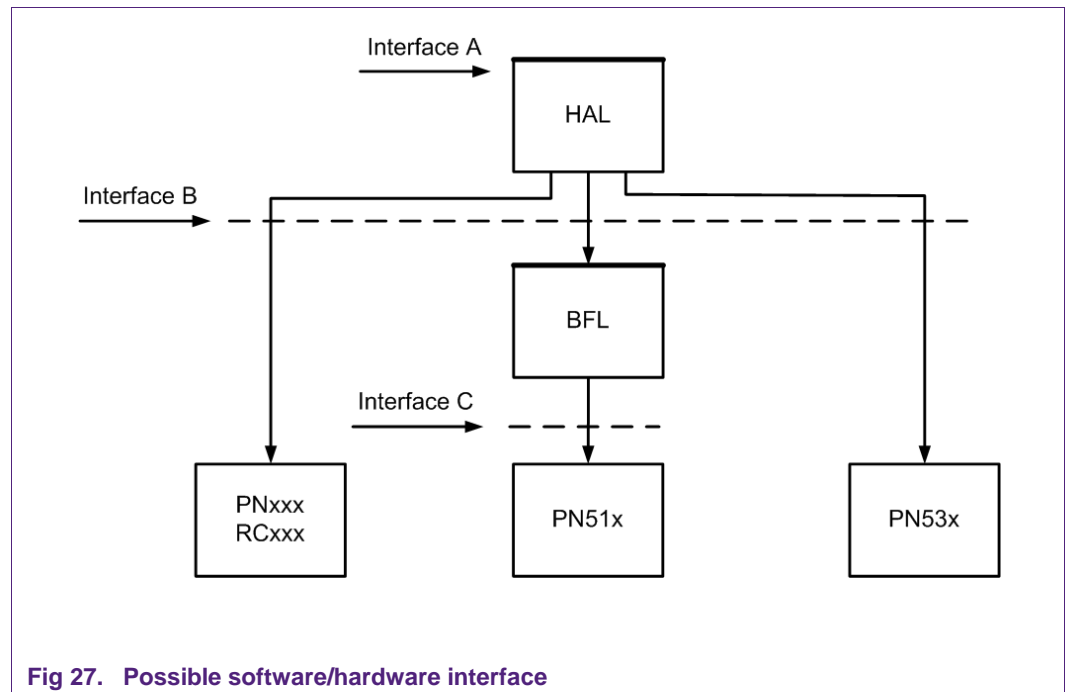


Fig 27. Possible software/hardware interface

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

### 12.2 The PN533 embedded software API: commands and errors lists

#### 12.2.1 Commands codes

Table 8. Commands codes

Command	PN533 as Initiator	PN533 as Target	Command Code
<b>Miscellaneous</b>			
Diagnose	X	X	0x00
GetFirmwareVersion	X	X	0x02
GetGeneralStatus	X	X	0x04
ReadRegister	X	X	0x06

Command	PN533 as Initiator	PN533 as Target	Command Code
WriteRegister	X	X	0x08
ReadGPIO	X	X	0x0C
WriteGPIO	X	X	0x0E
SetParameters	X	X	0x12
SAMConfiguration	X	X	0x14
PowerDown	X	X	0x16
AlparCommandForTDA	X		0x18
<b>RF communication</b>			
RFConfiguration	X	X	0x32
RFRregulationTest	X	X	0x58
<b>Initiator</b>			
InJumpForDEP	X		0x56
InJumpForPSL	X		0x46
InListPassiveTarget	X		0x4A
InATR	X		0x50
InPSL	X		0x4E
InDataExchange	X		0x40
InCommunicateThru	X		0x42
InQuartetByteExchange	X		0x38
InDeselect	X		0x44
InRelease	X		0x52
InSelect	X		0x54
InActivateDeactivatePaypass	X		0x48
<b>Target</b>			
TgInitAsTarget		X	0x8C
TgSetGeneralBytes		X	0x92
TgGetData		X	0x86
TgSetData		X	0x8E
TgSetDataSecure		X	0x96
TgSetMetaData		X	0x94
TdSetMetaDataSecure		X	0x98
TgGetInitiatorCommand		X	0x88
TgResponseToInitiator		X	0x90
TgGetTargetStatus		X	0x8A

## 12.2.2 Errors codes

**Table 9. Error codes**

Error cause	Error code
Time Out, the target has not answered	0x01
A CRC error has been detected by the CIU	0x02
A Parity error has been detected by the CIU	0x03

Error cause	Error code
During an anti-collision/select operation (ISO/IEC14443-3 Type A and ISO/IEC18092 106 kbps passive mode), an erroneous Bit Count has been detected	0x04
Framing error during MIFARE Classic operation	0x05
An abnormal bit-collision has been detected during bit wise anti-collision at 106 kbps	0x06
Communication buffer size insufficient	0x07
RF Buffer overflow has been detected by the CIU (bit BufferOvfl of the register <i>CIU_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	0x0B
Temperature error: the internal temperature sensor has detected overheating, and therefore has automatically switched off the antenna drivers	0x0D
Internal buffer overflow	0x0E
Invalid parameter (range, format, ...)	0x10
DEP Protocol: The PN533 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)	0x12
DEP Protocol, MIFARE Classic or ISO/IEC14443-4: The data format does not match to the specification. Depending on the RF protocol used, it can be: <ul style="list-style-type: none"> <li>• Bad length of RF received frame,</li> <li>• Incorrect value of PCB or PFB,</li> <li>• Invalid or unexpected RF received frame,</li> <li>• NAD or DID incoherence.</li> </ul>	0x13
MIFARE Classic: Authentication error	0x14
Target or Initiator does not support NFC Secure	0x18
I <sup>2</sup> C bus line is Busy. A TDA transaction is on going	0x19
ISO/IEC14443-3: 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 PN533 (Initiator vs. Target, unknown target number, Target not in the good state, ...)	0x27
The PN533 configured as target has been released by its initiator	0x29
PN533 and ISO/IEC14443-3B only: the ID of the card does not match, meaning that the expected card has been exchanged with another one.	0x2A
PN533 and ISO/IEC14443-3B 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

### 12.3 The PN533 various modes

This paragraph summarizes the PN533 functionalities and shows which commands are associated to them.

The PN533 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.

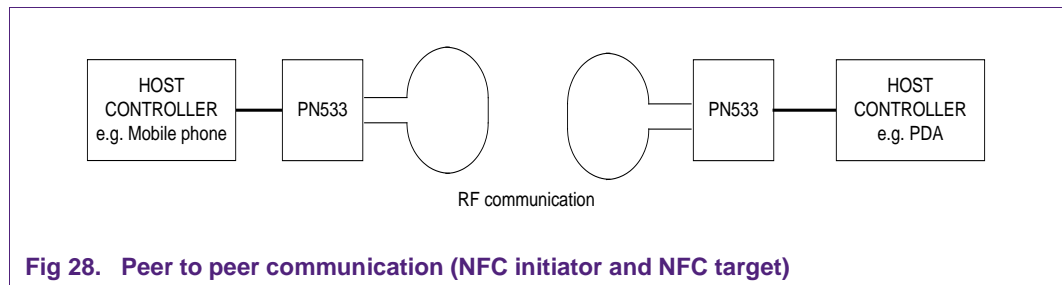


Fig 28. Peer to peer communication (NFC initiator and NFC target)

- As a reader for MIFARE Classic (MIFARE Classic protocol).

In this mode, RF communication is according to MIFARE Classic specification. the PN533 behaves as a reader for MIFARE Classic. It can communicate with MIFARE product-based cards.

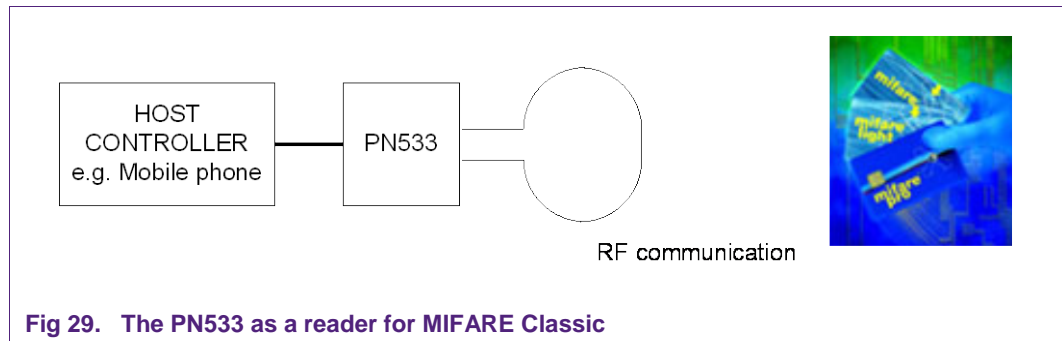


Fig 29. The PN533 as a reader for MIFARE Classic

The PN533 has been tested with MIFARE Classic 1K, MIFARE Classic 4K, MIFARE Ultralight, and MIFARE 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 PN533 behaves as an ISO/IEC 14443-4 reader. It can communicate with ISO/IEC 14443-4 cards (only ISO compliant cards are supported).

(The PN533 has been tested with CD97BX, CD light, DESFire, P5CN072 (SMX) as ISO/IEC 14443-4 (with JCOP OS))

- As a Jewel card reader

The PN533 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 PN533 has been tested with FeliCa RCS\_860 and RCS\_854

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

In this mode, the PN533 is combined with a secure smart card. An external reader sees the set the PN533 + secure smart card as a contactless card.

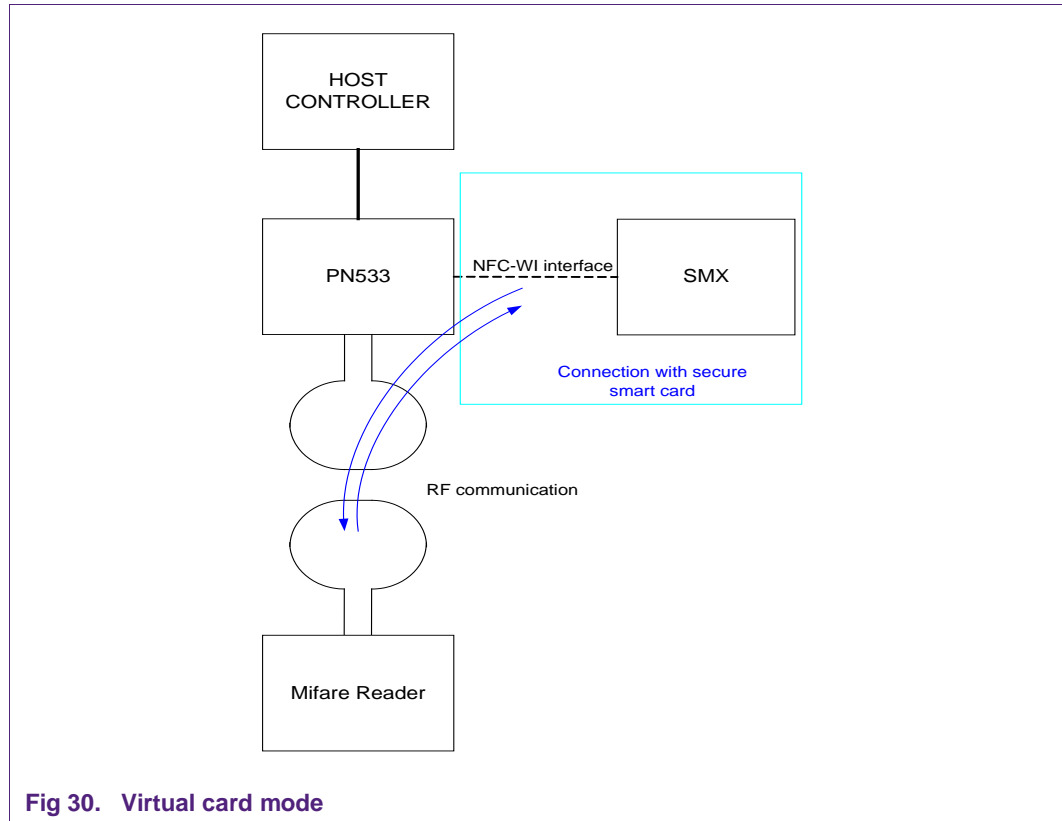


Fig 30. Virtual card mode

**RFConfiguration command**

This command, described in the user manual (see reference #1), allows changing some registers settings than can influence the RF communication. The default values are described in the user manual. The antenna 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 PN533 and another device (reader, card, or other NFC device). The PN533 executes **different RF processes**, depending on the type of communication, **but from the host controller to the PN533, same commands are used** (whatever the baudrate, the mode etc).

12.3.1 How to use the PN533 as a reader for MIFARE Classic?

Typical sequence (example):

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

This typical sequence can be performed with the following commands:

- **InListPassivTarget**, to initialize one card
- **InDataExchange**, to send MIFARE Classic commands
- **InSelect**, **InDeselect**, and **InRelease** to select, and release the card (this is optional).

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 PN533 performs automatically ISO/IEC 14443-4 activation (i.e. RATS sending). To disable automatic RATS sending, SetParameter command must be used (cf. user manual =Ref. #1).

**Table 10. SetParameter command usage to enable or disable automatic RATS sending (ISO/IEC 14443-4 mode)**

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Disable automatic sending of RATS command	12 04	Command code: <b>SetParameters</b> Automatic ATR_RES = 1	13	Response command code
<b>RATS will not be performed automatically by next InListPassiveTarget.command, even if the card indicates it supports ISO 14443-4</b>				
Enable automatic sending of RATS command (default configuration of the PN533 at power up)	12 14	Command code: <b>SetParameters</b> Automatic ATR_RES = 1 Automatic RATS = 1	13	Response command code
<b>RATS will be performed automatically by next InListPassiveTarget.command, if the card indicates it supports ISO 14443-4</b>				

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

The MIFARE Classic specific command byte Cmd may take one of the possible values:

- 60h / 61h Authentication A / Authentication B (MIFARE Classic 1K/4K)  
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 Classic 1K/4K)  
Write one data block (16 bytes) to the selected address of the card.
- A2h 4 bytes writing (MIFARE Ultralight)

Write one data block (4 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.

**Table 11. Data structure**

Byte	0	4	8	12	13	14	15
	3	7	11				
	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.

**Table 12. 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	-

**Table 13. MIFARE Classic 1K/4K cards**

*Authentication is required before any access to MIFARE Classic 1K/4K 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 Classic card documentation to have a more detailed description of the MIFARE Classic command set**

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

**Table 14. The PN533 as a reader for MIFARE Classic**

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code:	4B	Response command code
	01	<b>InListPassivTargets</b>	01	1 target detected
		Number of cards to initialize (if present in the field) is 1.	01	Target number 1
	00	Baud rate = 106 kbits/sec.	00 04	SENS_RES(2) of target 1
			08	SEL_RES(2) of target 1
			04	NFCID1 length = 4 bytes
			12 34 56 78	NFCID1 of target 1
<b>Target 1 is a MIFARE Classic 1K/4K card. The card is initialized (select).</b>				
Authenticate with MIFARE 1K/4K card	40	Command code:	41	Response command code
	01	<b>InDataExchange</b> The cmd is intended to target	00	Status = 0 (OK, no error)
	60	number 1 MIFARE Classic		
	07	cmd code (authenticate)		
	FF FF FF FF FF	MIFARE Classic address		
	FF 12 34 56 78	Default authentication key (last bytes are NFCID1 bytes)		
<b>Card1 has been waked up and authentication has been performed.</b>				
Read out card 1 memory	40	Command code:	41	Response command code
	01	<b>InDataExchange</b>	00	Status = 0 (OK, no error)
	30	The cmd is intended to target number 1	EE EE EE EE	Bytes read (example)
	04	MIFARE Classic cmd code (read 16 bytes)	EE EE EE EE	
		MIFARE Classic address	EE EE EE EE	
			EE EE EE EE	
<b>Any MIFARE commands, for example Read 16 bytes, can be sent to card 1 with InDataExchange function.</b>				
Halt card 1	44	Command code:	45	Response command code
	01	<b>InDeselect</b>	00	Status = 0 (OK, no error)
		The cmd is intended to target number 1		
<b>HALT REQ is sent with InDeselect command.</b>				

(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.



**MIFARE Plus cards**

These cards present some special features that were not known when implementing the PN533 firmware: here are some examples of communication with MIFARE Plus cards.

**Example of activation + authentication and read/write commands:**

The commands InListPassivTarget and InDataExchange are used.

At activation the MIFARE Plus card returns 7 bytes UID.

The authentication command sent by the InDataExchange **command uses only the four last bytes of UID.**

// Activate a target in passive mode

PC -> IFD: INITIATOR: LIST PASSIVE TARGETS

00 00 FF 04 FC D4 4A 01 00 E1 00

IFD -> PC: ACK

00 00 FF 00 FF 00

IFD -> PC: INITIATOR: LIST PASSIVE TARGETS EXECUTED

00 00 FF 0F F1 D5 4B 01 01 00 42 18 07 04 AB 0D

04 05 06 07 AB 00

// Authentication sector 0 (address 0 to 3)

PC -> IFD: INITIATOR: DATA EXCHANGE

00 00 FF 0F F1 D4 40 01 60 03 FF FF FF FF FF FF

04 05 06 07 78 00

IFD -> PC: ACK

00 00 FF 00 FF 00

IFD -> PC: INITIATOR: DATA EXCHANGE EXECUTED

00 00 FF 03 FD D5 41 00 EA 00

// Authentication sector 1 (address 4 to 7)

PC -> IFD: INITIATOR: DATA EXCHANGE

00 00 FF 0F F1 D4 40 01 60 07 FF FF FF FF FF FF

04 05 06 07 74 00

IFD -> PC: ACK

00 00 FF 00 FF 00

IFD -> PC: INITIATOR: DATA EXCHANGE EXECUTED

00 00 FF 03 FD D5 41 00 EA 00

```
// Read 16 bytes from block address 0x04
PC ->  IFD: INITIATOR: DATA EXCHANGE
        00 00 FF 05 FB D4 40 01 30 04 B7 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: INITIATOR: DATA EXCHANGE EXECUTED
        00 00 FF 13 ED D5 41 00 00 00 00 00 00 00 00 00
        00 00 00 00 00 00 00 00 EA 00
```

**Example of activation + a specific MIFARE Plus command:**

At activation the SAK byte indicates MIFARE Plus can also act like an ISO/IEC 14443 card: disable auto RATS to stay in the MIFARE Plus mode.

```
PC ->  IFD: SET TAMA PARAMETERS
        00 00 FF 03 FD D4 12 00 1A 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: SET TAMA PARAMETERS EXECUTED
        00 00 FF 02 FE D5 13 18 00
```

```
// Look for a 106 kbps card
PC ->  IFD \: INITIATOR: LIST PASSIVE TARGETS
        00 00 FF 04 FC D4 4A 01 00 E1 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: INITIATOR: LIST PASSIVE TARGETS EXECUTED
        00 00 FF 0F F1 D5 4B 01 01 00 42 18 07 04 66 C5
        04 05 06 07 38 00
```

The specific MIFARE Plus commands are sent by using the InCommunicateThru command without checking the CRC of response that is received.

```
// Change CRC of RxMod: disable CRC Rx
PC ->  IFD: WRITE REGISTERS
        00 00 FF 05 FB D4 08 63 03 00 BE 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: WRITE REGISTERS EXECUTED
        00 00 FF 02 FE D5 09 22 00
```

```

PC ->  IFD: INITIATOR: COMMUNICATE THRU
        00 00 FF 15 EB D4 42 A8 02 90 11 11 11 11 11 11
        11 11 11 11 11 11 11 11 11 11 11 A0 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: INITIATOR: COMMUNICATE THRU EXECUTED
        00 00 FF 04 FC D5 43 00 0A DE 00

```

**Please refer to MIFARE Plus card documentation to have a more detailed description of the MIFARE Plus command set and specific features.**

**Example of activation of a MIFARE Plus card as a ISO/IEC 14443 type A card:**

Activation is done as usual by using the InListPassiveTarget command.

If the MIFARE Plus card shows in its SAK byte that it is also compliant with ISO14443 standard, then RATS is sent automatically and the rest of communication can be done with the InDataExchange command: it is the case of SL0.

```

PC ->  IFD: INITIATOR: LIST PASSIVE TARGETS
        00 00 FF 04 FC D4 4A 01 00 E1 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: INITIATOR: LIST PASSIVE TARGETS EXECUTED
        00 00 FF 18 E8 D5 4B 01 01 00 04 20 04 6D 2A E9
        02 0C 75 77 80 02 C1 05 2F 2F 00 35 C7 9A 00
PC ->  IFD: INITIATOR: DATA EXCHANGE
        00 00 FF 16 EA D4 40 01 A8 00 90 00 00 00 00 00
        00 00 00 00 00 00 00 00 00 00 00 00 B3 00
IFD ->  PC: ACK
        00 00 FF 00 FF 00
IFD ->  PC: INITIATOR: DATA EXCHANGE EXECUTED
        00 00 FF 04 FC D5 41 00 90 5A 00

```

40 = InDataExchange

01 = Target number

A8 00 90 00 00 00 00 00 00 00 00 00 00 00 00 00 = ISO1443-4 command APDU

If the MIFARE Plus card does not show in its SAK byte (bit 6 =0) that it is also compliant with ISO14443 standard, then RATS has to be sent manually and the rest of

communication can be done with the InCommunicateThru command: it is the case of SL1.

```
// Look for the card
```

```
PC -> IFD: INITIATOR: LIST PASSIVE TARGETS
```

```
00 00 FF 04 FC D4 4A 01 00 E1 00
```

```
IFD -> PC: ACK
```

```
00 00 FF 00 FF 00
```

```
IFD -> PC: INITIATOR: LIST PASSIVE TARGETS EXECUTED
```

```
00 00 FF 0F F1 D5 4B 01 01 00 42 18 07 04 7F 23
```

```
A9 61 28 80 25 00
```

```
// send RATS
```

```
PC -> IFD: INITIATOR: COMMUNICATE THRU
```

```
00 00 FF 04 FC D4 42 E0 82 88 00
```

```
IFD -> PC: ACK
```

```
00 00 FF 00 FF 00
```

```
IFD -> PC: INITIATOR: COMMUNICATE THRU EXECUTED
```

```
00 00 FF 0F F1 D5 43 00 0C 75 77 80 02 C1 05 2F
```

```
2F 01 BC D6 B7 00
```

```
// test InCommunicateThru for sending commands to MIFARE Plus card
```

```
// Present SL3 Switch Key
```

```
// 90 03 = address of SL3 switch key, be careful it must be inverted (03 90) in the command
```

```
PC -> IFD: INITIATOR: COMMUNICATE THRU
```

```
00 00 FF 08 F8 D4 42 0A 02 70 03 90 00 DB 00
```

```
IFD -> PC: ACK
```

```
00 00 FF 00 FF 00
```

```
IFD -> PC: INITIATOR: COMMUNICATE THRU EXECUTED
```

```
00 00 FF 16 EA D5 43 00 0A 02 90 94 25 8C E5 86
```

```
EC E3 BD 36 F5 1C F8 9F 68 F6 CA 0A 00
```

**Timeout and number of retries:****Activation phase (InListPassiveTarget command)**

By default, the PN533 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, see user manual = ref. #1).

If there is no card in the field, a timeout occurs after 5 ms. Either the PN533 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 (see user manual).

**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: The PN533 is capable of handling 1 target maximum at once. It is needed to use InDeselect command to handle another card.**

12.3.2 How to use the PN533 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 initialize one card.  
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 indicating 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).

Table 15. The PN533 as an ISO/IEC 14443-4 reader. Type A card activation

Action	Command <sup>4</sup>	Command explanation	Response	Response explanation	
Scan for 1 target in the field and initialize it	4A	Command code:	4B	Response command code	
	01	InListPassivTargets	01	1 target detected	
	00	Number of cards to initialize (if present in the field) = 1 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 PN533 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	00	Status = 0 (OK, no error)
	00 B0 81 00	number 1	00 11 22 33 44 55	Response of the card
	10	ISO/IEC 14443-4 command	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 SetParameters command. Please refer to the PN533 User manual (cf. Ref. 1).

- (3) The PN533 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 16. The PN533 as an ISO/IEC 14443-4 reader. Type B card activation (timeslot method)**

Action	Command <sup>4</sup>	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code:	4B	Response command code
	01	InListPassivTargets	01	1 target detected
		Number of cards to initialize (if present in the field) = 1	01	Target number 1
	03	Baud rate = 106 kbits/sec Type B.	50 01 02 03 04 00	ATQB_RES (12 bytes)
	00	AFI	00 00 00 00 00 00	
	No other parameter : default timeslot method will be used.	01	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 example Select 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 17. The PN533 as an ISO/IEC 14443-4 reader. Type B card activation (probabilistic polling method)**

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

**Type B card is activated. The method used is the probabilistic approach.**

### Timeout and number of retries:

#### **Activation phase (InListPassiveTarget command)**

By default, the PN533 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 user manual (see Ref. #1).

If there is no card in the field, a timeout occurs after 5 ms. Either the PN533 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 PN533. The error handling and the chaining are also fully managed by the PN533.

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

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

**Note: The PN533 is capable of handling 1 target maximum at once. It is needed to use InDeselect command to handle another card.**



### 12.3.3 How to use the PN533 as a DESFire cards reader?

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

**Table 18. The PN533 as a DESFire card reader**

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

Bit 5 of SEL\_RES indicates the target is ISO/IEC 14443-4 compliant.

In that case the PN533 automatically sends the RATS<sup>(2)</sup>. ATS is indicated in the response.

40	40	40	40	Response command code
01	01	01	01	Status = 0 (OK, no error)
6A	6A	6A	6A	Response of the card

DESFire commands, for example GetApplicationID command, can be sent with InDataExchange command<sup>(3)</sup>.

40	40	40	40	Response command code
01	01	01	01	Status = 0 (OK, no error)
5A 06 00 00	5A 06 00 00	5A 06 00 00	5A 06 00 00	

#### Timeout and number of retries:

Cf. paragraph 12.3.2

12.3.4 How to use the PN533 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 initialize one card.
- **InDataExchange**, to transfer data/command bytes to the card (The PN533 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 19. The PN533 as a FeliCa reader

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code:	4B	Response command code
	01	InListPassivTargets	01	1 target detected
		Number of cards to initialize (if present in the field) = 1	01	Target number 1
	01	Baud rate = 212 kbits/sec.	12	POL_RES length
	00 FF FF	Payload field of polling request	01	Response code
	00 00		01 01 06 01 46 05	NFCID2
			C3 1A	
			04 01 4B 02 4F 49	PAD
		93 FF		

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	Data	00 FF AA BB	
	BB			

- (1) Command code and command parameters. Mandatory protocol encapsulation is not represented.
- (2) Example of FeliCa command. Not all the FeliCa cards support this command. Refer to FeliCa card specification.

**Timeout and number of retries:****Activation phase (InListPassiveTarget command)**

By default, the PN533 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 user manual (see reference #1).

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

Either the PN533 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 the user manual.

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

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

**12.3.5 How to use the PN533 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 initialize one card.
- InDataExchange, to transfer data/command bytes to the card
- InSelect, InDeselect, and InRelease to select, and release the card.

**Table 20. The PN533 as a Jewel card reader**

Action	Command <sup>4</sup>	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code:	4B	Response command code
	01	InListPassivTargets	01	1 target detected
	04	Number of cards to initialize (if present in the field) = 1	01	Target number 1
		Baud rate = 106 kbits/sec, type = Innovision Jewel	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 (example : 255 bytes, 01 to FF. Not all bytes are shown here)

**The PN533 transfers the data.**

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

**Table 21. Jewel commands**

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

## 12.3.6 How to use the PN533 in a NFC peer-to-peer communication

### 12.3.6.1 How to use the PN533 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):

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

Another typical sequence (example 2): compared to example 1, initialization 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 initialize), 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).
- InRelease to release the target.

Table 22. The PN533 as a NFC initiator (“reader”) example 1

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Initialize and activate a target	56	Command code: <b>InJumpForDep</b>	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
	00 FF FF	Payload field of polling request	44 33 22 11	(except cmd0 and cmd1 bytes)
	00 00		00 00 00 09 01 22	

Initiator has chosen the active mode. the PN533 performed automatically activation and possibly parameter selection (ATR\_REQ and possibly PSL\_REQ)<sup>(2)</sup>.

Exchange data with the card	40	Command code: <b>InDataExchange</b>	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			

PN533 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.

Deselect	44	Command code: <b>InDeselect</b>	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: <b>InSelect</b>	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: <b>InRelease</b>	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 PN533 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 23. The PN533 as a NFC initiator (“reader”) example 2

Action	Command <sup>3</sup>	Command explanation	Response	Response explanation
Scan for 1 target in the field and initialize it	4A	Command code: <b>InListPassivTargets</b>	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	Payload field of polling request	12	POL_RES length
	00 00		01	Response code
			01 FE A2 A3 A4	NFCID2
			A5 A6 A7	
			C0 C1 C2 C3 C4	PAD

Action	Command <sup>3</sup>	Command explanation	Response	Response explanation
			C5 C6 C7	
<b>The target has been initialised (in passive mode) at the requested baud rate.</b>				
Activate the target	50 01	Command code: <b>InATR</b> 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)
<b>The target has been activated.</b>				
Change parameters	4E 01 00 00	Command code: <b>InPSL</b> The cmd is intended to target number 1 New baud rate = 106 kbits/sec	4F 00	Response command code Status = 0 (OK, no error)
<b>The baud rate has been changed.</b>				
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: <b>InDataExchange</b> 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)
<b>The PN533 transfers the data according to NFCIP-1 transport protocol. Error handling, chaining, time out extension are automatically handled.</b>				
Release the target	52 01	Command code: <b>InRelease</b> The cmd is intended to target number 1	53 00	Response command code Status = 0 (OK, no error)
<b>RLS_REQ is sent with InRelease command.</b>				

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

### About PSL (Parameter Selection)

When using InJumpForDEP command, the PN533 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 PN533 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 PN533 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 0 applies. At 212 or 424 kbps, paragraph 0 applies.

Either the PN533 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 PN533 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.

#### 12.3.6.2 How to use the PN533 as a target in a NFC peer-to-peer communication

In this mode, the PN533 is configured as target, meaning it keeps waiting for an initiator command.

**The PN533 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 PN533. The PN533 transfers the response from the host to the initiator. Initialisation/activation is handled automatically by the PN533.**

Typical exchange:

- Be ready to respond to an initiator, whatever 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 PN533 as a target,
- **TgGetData**, to wait for data coming from the initiator,
- **TgSetData**, to respond to the initiator.



**Table 24. The PN533 as NFC target**

Action	Command <sup>4</sup>	Command explanation	Response	Response explanation
Be ready to respond to an initiator, whatever the mode and the baud rate	8C	Command code: TgInitAsTarget	8D	Response command code
	00	Accepted modes : 0 = all	22	Mode : passive mode, 424 kbits/s
	08 00	SENS_RES	11 D4 00 01 FE A2	
	12 34 56	NFCID1	A3 A4	Initiator command received (ATR_REQ)
	40	SEL_RES	A5 A6 A7 00 00 00	
	01 FE a2	Parameters to build POL_RES (16 bytes)	00 00	
	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 target was waiting for any initialization command. In this example, it has been initialized at 424 kbit/s in passive mode. POL\_Res and ATR\_RES have been automatically sent by the PN533**

Wait for data to be transferred to the host controller	86	Command code: TgGetData	87	Response command code
			00	Status = 0 (OK, no error)
			98 76	Data received

**The target received some data from the initiator.**

Sends data response from the host controller	8E	Command code: TgSetData	8F	Response command code
		No data sent back to the controller	00	Status = 0 (OK, no error)

**The target responded to the initiator (it can possibly send some data).**

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

#### What are default timeout values of the PN533 as a target?

WT = 09h (ATR\_RES parameter) → RWT = 154ms approx.

RTOX = 07h (Timeout extension request parameter) → RWTINT = 1078ms approx.

#### How to fill TgInitAsTarget parameters?

##### Mode (1 byte)

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

Mode = 01h: to refuse active communication mode

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

##### MIFARE ICs 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 PN533 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.

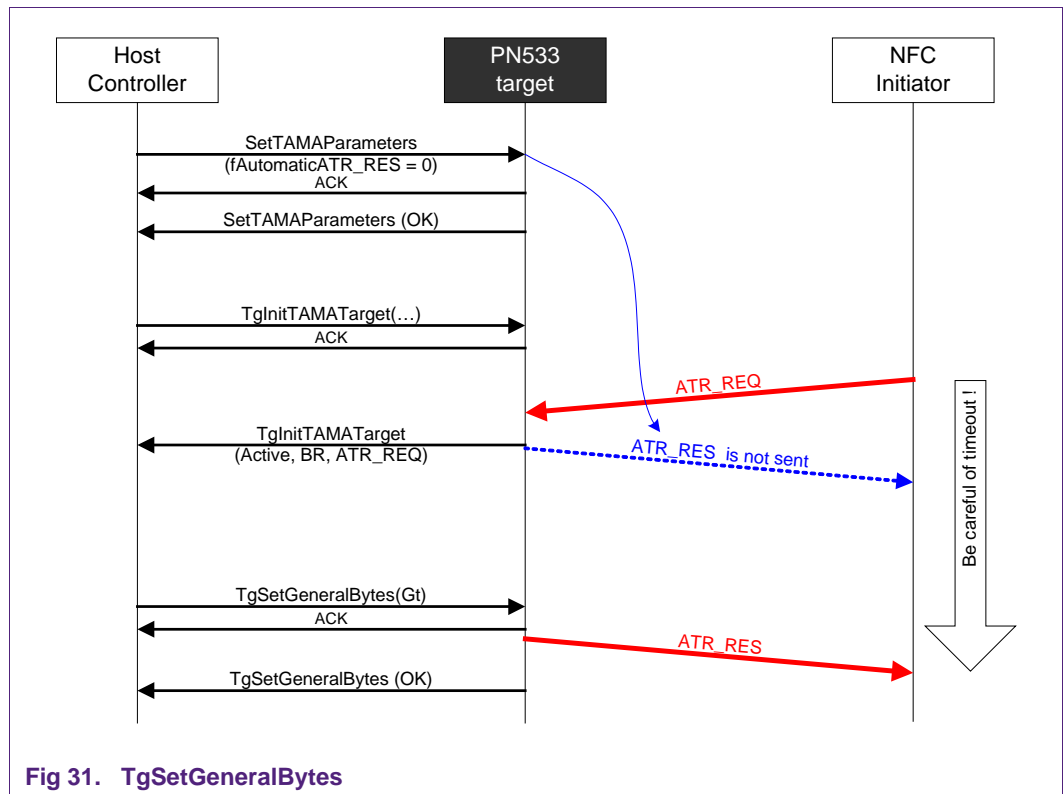


Fig 31. TgSetGeneralBytes

**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 PN533 is in ISO/IEC 14443-4 PICC emulation mode.

**12.3.6.3 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 PN533 User manual (reference #1) for detailed explanation.

#### 12.3.6.4 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 PN533, depends on several parameters:

- The RF baudrate
- The amount of data:
  - The PN533 length reduction is 0: packets size on RF interface is 64 bytes max. The time to transfer the data depends on the number of packets necessary.
  - The number of packets on host link influences the transfer time as well. The PN533 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 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) does not influence the performances.

### 12.3.7 How to use Smart connectivity (combination of the PN533 and SMX)?

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

Combining the PN533 and SMX (P5CN072) allows dealing with application that requires security such as payment applications.

The frame delay time (FDT) can be adjusted in the PN533, thanks to bit 5 of register address 0x630D (DELAY\_MF\_SO bit of Manual Rcv register. See reference 2), a WriteRegister command can be used, after SAMconfiguration.

NFC-WI interface is used.

The SMX power is supplied by the PN533 (SVDD). In case an external power supply is used, it has to be between 2.7V and 3.3V.

Commands needed to use the PN533 + SMX are:

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

#### 12.3.7.1 Virtual card mode

In virtual card mode, the (PN533 + SMX P5CN072) is seen as a contactless secure smart card. Only one command, SAMConfiguration, is needed to put the PN533+SMX in this mode.

Optionally, the PN533 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 PN533 only acts a bridge between SMX and the external reader.

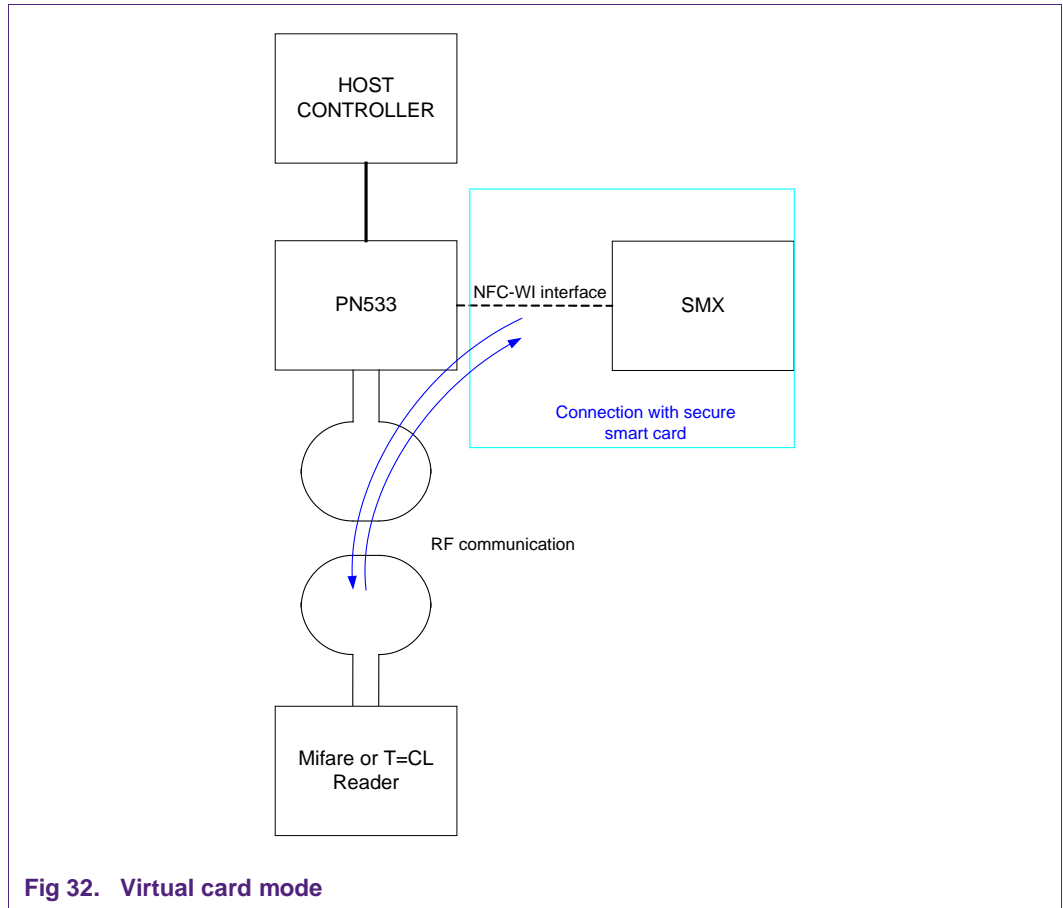


Fig 32. Virtual card mode

Depending on the first command, after initialisation, sent by the reader, the PN533+SMX will act as a MIFARE product-based card or as a T=CL card.

Table 25. The PN533 +SMX as virtual card

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Set the PN533 in virtual card mode	14	Command code:	15	Response command code
	02	<b>SAMConfiguration</b>		
	00	Virtual card mode No timeout		

The PN533 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.

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).

### 12.3.7.2 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 PN533 access SMX to check what happened.

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

The PN533 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 PN533 must be disabled.

**Table 26. The PN533 +SMX as wired ISO/IEC 1443-4 card**

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Set the PN533 in wired card mode	14 03	Command code: <b>SAMConfiguration</b> Wired card mode	15	Response command code

**The PN533 is configured in wired card mode. SMX is accessed by the PN533 as a contactless card.**

Initialize the SMX	4A	Command code:	4B	Response command code
	01	InListPassivTargets	01	1 target detected
	00	Number of cards to initialize = 1 Baud rate = 106 kbits/sec.	01	Target number 1
			04 07	SENS_RES
			28	SEL_RES
			04	NFCID1 length
			00 D7 1E 92	NFCID1
			0D 77 80 81 02 00 73 C8 40 13 00 90 00	ATS (13 bytes)

**The PN533 communicates with the SMX as with a card. If SMX indicates T=CL compliance, the PN533 automatically sends RATS command.**

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

**Table 27. The PN533 +SMX as wired MIFARE product-based card**

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Disable automatic sending of RATS command	12 04	Command code: <b>SetParameters</b> Automatic ATR_RES = 1	13	Response command code

**The PN533 is configured in wired card mode. SMX is accessed by the PN533 as a contactless card.**

Set the PN533 in wired card mode	14 03	Command code: <b>SAMConfiguration</b> Wired card mode	15	Response command code
----------------------------------	----------	---	----	-----------------------

**The PN533 is configured in wired card mode. SMX is accessed by the PN533 as a contactless card.**

Initialize the SMX	4A	Command code:	4B	Response command code
	01	InListPassivTargets	01	1 target detected
	00	Number of cards to initialize = 1 Baud rate = 106 kbits/sec.	01	Target number 1
			04 07	SENS_RES
		28	SEL_RES	

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
			04	NFCID1 length
			00 D7 1E 92	NFCID1

As automatic sending of RATS was disabled, it has not been sent by the PN533.

If SMX supports MIFARE implementation, it is now ready to answer in this mode

Read some data in the SMX memory	40	Command code: InDataExchange	41	Response command code
	01	Target number 1	00	Status = OK
	30	MIFARE Read16bytes command	00 D7 1E 92 5B 28	16 bytes read
	00	(Address 00)	04 07 00 00 00 00 41 07 11 00	

The PN533 communicates with the SMX using MIFARE product-based commands.

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



**12.3.7.3 Dual card mode**

In this mode, both the PN533 (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 28. The PN533 +SMX as dual card**

Action	Command <sup>1</sup>	Command explanation	Response	Response explanation
Set the PN533 in Dual card mode	14 04	Command code: <b>SAMConfiguration</b> Dual card mode	15	Response command code

**The PN533 is configured in dual card mode.**

Configure the PN533 as a target	8C	Command code: TgInitAsTarget	8D	Response command code
	00	Accepted modes : 0 = all	22	Mode : passive mode, 424 kbits/s
	08 00	SENS_RES	11 D4 00 01 FE A2	
	12 34 56	NFCID1	A3 A4	Initiator command received (ATR_REQ)
	40	SEL_RES	A5 A6 A7 00 00 00 00 00	
	01 FE a2 a3 a4 a5 a6 a7	Parameters to build POL_RES (16 bytes)		
	c0 c1 c2 c3 c4 c5 c6 c7			
	FF FF			
	AA 99 88	NFCID3t (10 bytes)		
	77 66 55 44 33 22 11			
	00 00	Length of Gt and Tk		

**SMX (P5CN072) and the PN533 can respond to a reader only after TgInitAsTarget command has been sent.**

## 13. Annex: Demoboard schematic

---

### Recommendations

The USB interface is configured by I0 and I1 pins connected to DVDD.

Standard operating mode is configured at power-on reset by pins P70\_IRQ and P35.

Recommended capacitors:

- On TVDD: 4.7  $\mu$ F in parallel with 100 nF; ceramic X7R type
- On DVDD: 4.7  $\mu$ F in parallel with 100 nF; ceramic X7R type
- On AVDD: 100 nF; ceramic X7R type
- On PVDD: 100 nF; ceramic X7R type
- Antenna matching and tuning components: use COG capacitors. Concerning the antenna design and tuning, please refer to the application note “NFC Transmission Module Antenna and RF Design Guide” AN100720.

See also application note AN10683\_1 about PN533 demo board for more information.

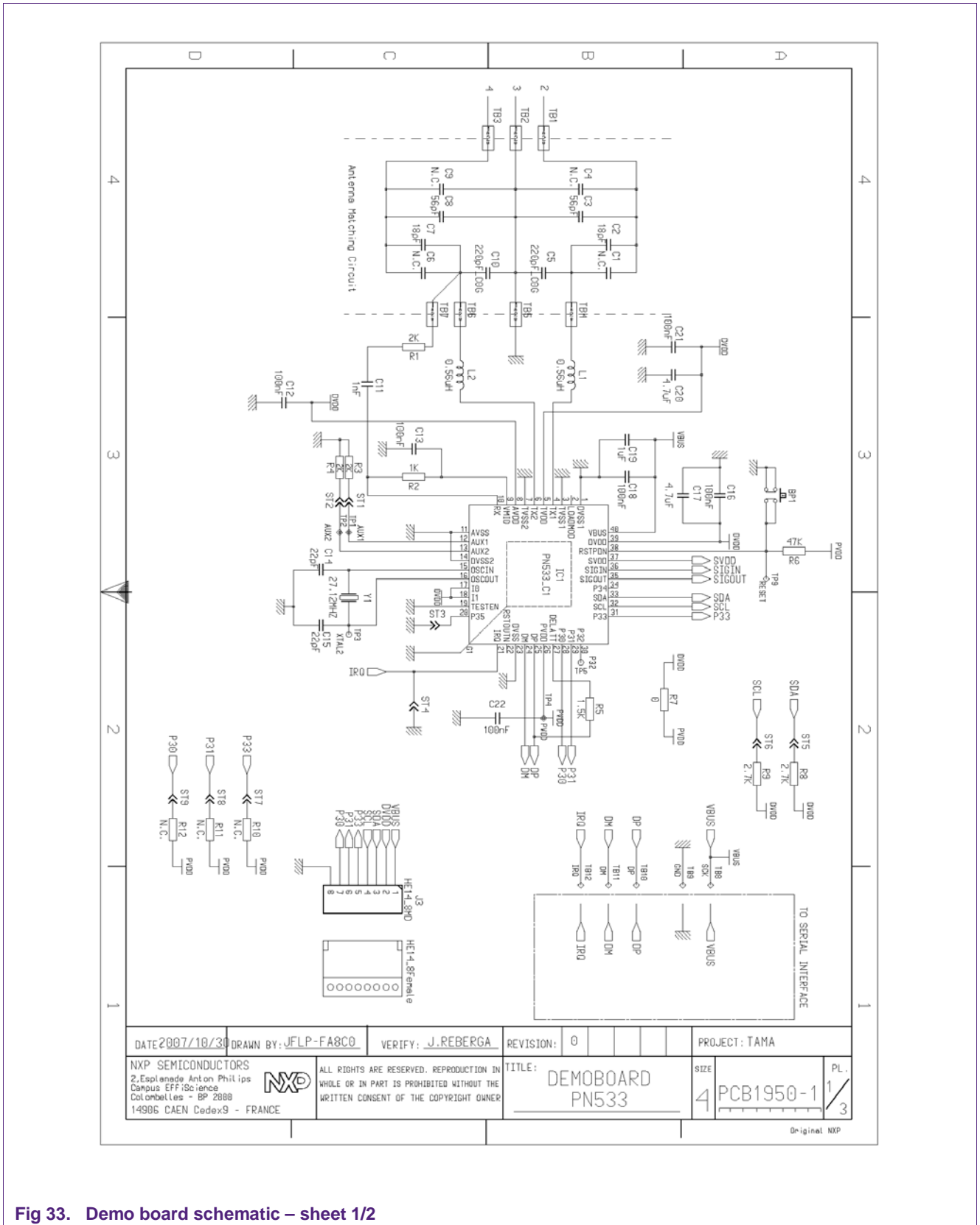


Fig 33. Demo board schematic – sheet 1/2

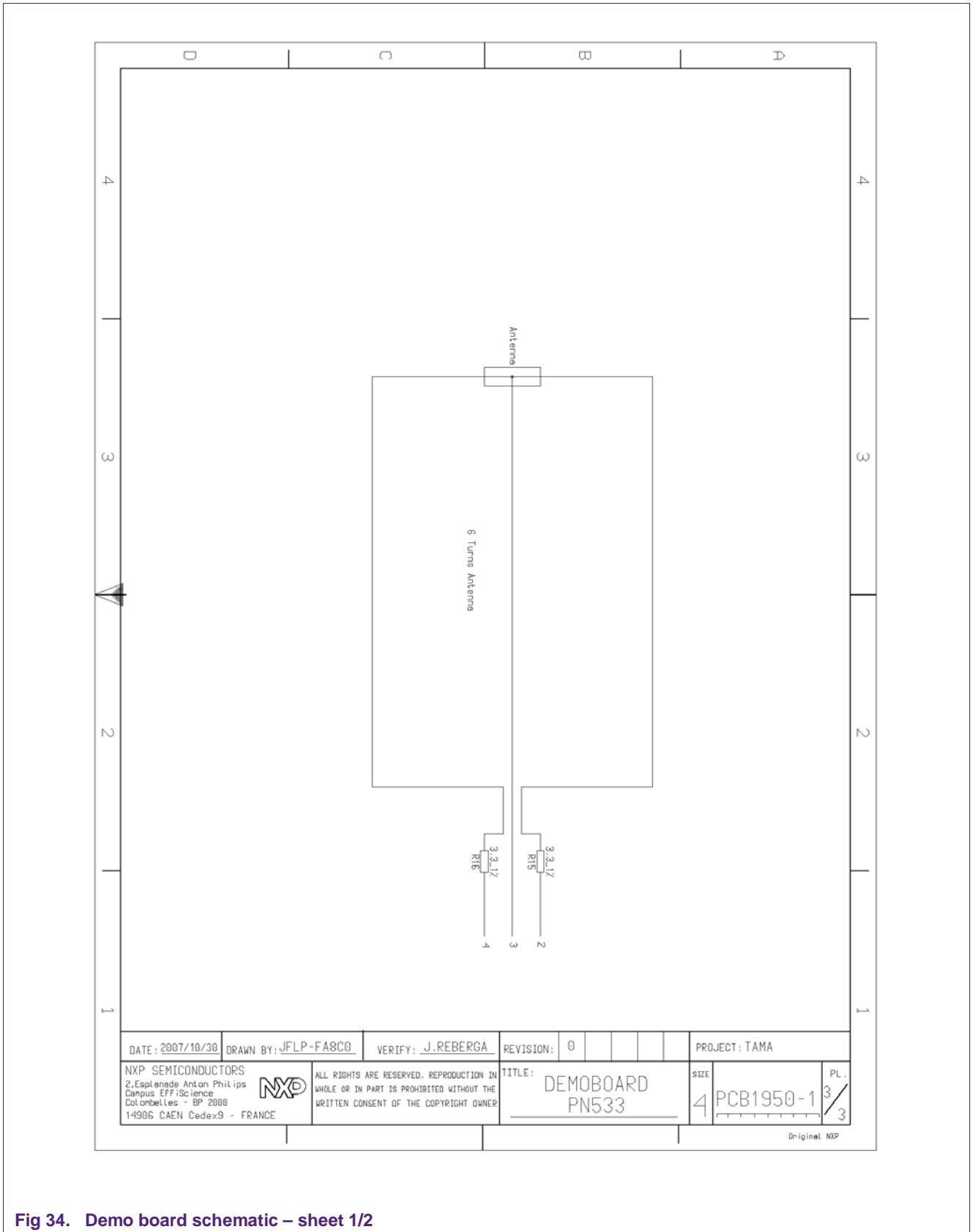


Fig 34. Demo board schematic – sheet 1/2

## 14. Abbreviations

**Table 29. Abbreviations**

Acronym	Description
CIU	Contactless Interface Unit
DEP	Data Exchange Protocol
EMV	Europay MasterCard Visa
HSU	High Speed UART
I <sup>2</sup> C	Inter Integrated Circuit
NFC	Near Field Communication
NFC-WI	NFC Wired Interface
PCD	Proximity Coupling Device
PCR	Power, Clock and Reset controller
PICC	Proximity Integrated Circuit Card
PMU	Power Management Unit
RFA	Radio Frequency Activation
SMX	NXP SmartMX (Memory Extension): secure smart card controller

## 15. References

- [1] **PN533 User Manual** - UM0801\_03.pdf
- [2] **PN5331B3HN data sheet** - PN5331B3HN\_DS.pdf
- [3] **ISO/IEC 14443-2** - Identification cards – Contactless integrated circuit(s) cards Proximity card(s) - Part 2: RF power and signal interface
- [4] **ISO/IEC 14443-3** - Identification cards – Contactless integrated circuit(s) cards - Proximity card(s) - Part 3: Initialization and anti-collision
- [5] **ISO/IEC 14443-4** - Identification cards – Contactless integrated circuit(s) cards - Proximity card(s) - Part 4: Transmission protocol
- [6] **ISO/IEC 18092** - Near Field Communication - Interface and Protocol (NFCIP-1)
- [7] **AN144511** NFC Transmission Module Antenna and RF Design Guide - AN144511.pdf
- [8] **AN142520** - RF amplifier for NXP contactless NFC reader ICs
- [9] **USB** - Universal Serial Bus Specification
- [10] **I<sup>2</sup>C** - I<sup>2</sup>C bus Specification
- [11] **DESFire cards specification** - Document number: 075041.pdf
- [12] **MIFARE cards specification** - [https://www.nxp.com/products/identification-and-security/mifare-ics:MC\\_53422](https://www.nxp.com/products/identification-and-security/mifare-ics:MC_53422)
- [13] **MasterCard Paypass specification Paypass** - ISO/IEC 14443 Implementation Specification V1.1
- [14] **EMVCo** - EMV Contactless Communication Protocol Specification V2.0

## 16. Legal information

### 16.1 Definitions

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

### 16.2 Disclaimers

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

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

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

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

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

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

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

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

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

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

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

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

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

### 16.3 Licenses

#### ICs with DPA Countermeasures functionality



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

#### Purchase of NXP ICs with NFC technology

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

### 16.4 Trademarks

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

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

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

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

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

**DESFire** — is a trademark of NXP B.V.

## 17. List of figures

Fig 1.	Simplified system view .....	3
Fig 2.	PN533 Block diagram .....	7
Fig 3.	USB bus powered .....	8
Fig 4.	HOST powered application .....	9
Fig 5.	Two power supplies application .....	10
Fig 6.	USB description of the PN533 .....	12
Fig 7.	PN533 application diagram: USB interface.....	13
Fig 8.	PN533 application diagram: HSU interface.....	15
Fig 9.	Normal exchange between host controller and the PN533 .....	18
Fig 10.	Information frame .....	18
Fig 11.	ACK frame .....	19
Fig 12.	Extended Information frame.....	19
Fig 13.	NACK frame.....	20
Fig 14.	Error frame.....	20
Fig 15.	USB link: frames .....	21
Fig 16.	USB link: general principle of communication.	21
Fig 17.	Sequence of reading information on I <sup>2</sup> C EEPROM to perform enumeration .....	22
Fig 18.	Reading of EEPROM information for enumeration .....	23
Fig 19.	EEPROM mapping.....	24
Fig 20.	PN533 and TDA8029 connections.....	25
Fig 21.	PN533 application diagram: USB interface.....	27
Fig 22.	TDA8029 application diagram (Energy Saving Mode enabled) .....	28
Fig 23.	PayPass/EMVCo compliant application .....	29
Fig 24.	Schematic of the additional RF amplifier.....	30
Fig 25.	List of components of the additional RF amplifier .....	30
Fig 26.	NFC Secure frame .....	31
Fig 27.	Possible software/hardware interface .....	33
Fig 28.	Peer to peer communication (NFC initiator and NFC target) .....	36
Fig 29.	The PN533 as a reader for MIFARE Classic...36	
Fig 30.	Virtual card mode.....	37
Fig 31.	TgSetGeneralBytes.....	59
Fig 32.	Virtual card mode.....	62
Fig 33.	Demo board schematic – sheet 1/2 .....	67
Fig 34.	Demo board schematic – sheet 1/2 .....	68

## 18. List of tables

Table 1.	Configuration modes.....	5
Table 2.	IC correspondence.....	5
Table 3.	TX framing and TX speed in RFfieldON configuration .....	5
Table 4.	Interface selection.....	11
Table 5.	USB VID/PID.....	13
Table 6.	HSU baud rates .....	14
Table 7.	Possible issue for thePN533.....	16
Table 8.	Commands codes .....	33
Table 9.	Error codes .....	34
Table 10.	SetParameter command usage to enable or disable automatic RATS sending (ISO/IEC 14443-4 mode).....	38
Table 11.	Data structure .....	39
Table 12.	MIFARE Ultralight cards.....	39
Table 13.	MIFARE Classic 1K/4K cards .....	39
Table 14.	The PN533 as a reader for MIFARE Classic...	40
Table 15.	The PN533 as an ISO/IEC 14443-4 reader. Type A card activation .....	46
Table 16.	The PN533 as an ISO/IEC 14443-4 reader. Type B card activation (timeslot method).....	47
Table 17.	The PN533 as an ISO/IEC 14443-4 reader. Type B card activation (probabilistic polling method).....	47
Table 18.	The PN533 as a DESFire card reader .....	49
Table 19.	The PN533 as a FeliCa reader .....	50
Table 20.	The PN533 as a Jewel card reader.....	52
Table 21.	Jewel commands .....	52
Table 22.	The PN533 as a NFC initiator (“reader”) example 1 .....	54
Table 23.	The PN533 as a NFC initiator (“reader”) example 2 .....	54
Table 24.	The PN533 as NFC target.....	57
Table 25.	The PN533 +SMX as virtual card.....	62
Table 26.	The PN533 +SMX as wired ISO/IEC 1443-4 card.....	63
Table 27.	The PN533 +SMX as wired MIFARE product-based card .....	63
Table 28.	The PN533 +SMX as dual card .....	65
Table 29.	Abbreviations .....	69



## 19. Contents

<b>1.</b>	<b>Introduction</b> .....	<b>3</b>		
<b>2.</b>	<b>PN533 configuration modes</b> .....	<b>4</b>	12.3.4	How to use the PN533 as a FeliCa reader? .....50
2.1	Standard mode.....	5	12.3.5	How to use the PN533 as a Jewel cards reader? .....52
2.2	PN512 emulation mode.....	5	12.3.6	How to use the PN533 in a NFC peer-to-peer communication .....53
2.3	RFfieldON Mode .....	5	12.3.6.1	How to use the PN533 as an initiator in a NFC peer-to-peer communication.....53
<b>3.</b>	<b>PN533 block diagram</b> .....	<b>7</b>	12.3.6.2	How to use the PN533 as a target in a NFC peer-to-peer communication.....56
<b>4.</b>	<b>Power distribution</b> .....	<b>8</b>	12.3.6.3	Transfer of large amount of data .....59
4.1	USB bus powered application .....	8	12.3.6.4	Effective throughput .....
4.2	Host powered application (single source) .....	9	12.3.7	How to use Smart connectivity (combination of the PN533 and SMX)?.....61
4.3	Host powered application (two sources) .....	10	12.3.7.1	Virtual card mode .....
<b>5.</b>	<b>Interfaces with the host controller</b> .....	<b>11</b>	12.3.7.2	Wired card mode .....
5.1	Interface selection .....	11	12.3.7.3	Dual card mode .....
5.2	USB interface .....	12	<b>13.</b>	<b>Annex: Demoboard schematic</b> .....
5.3	HSU interface.....	14	<b>14.</b>	<b>Abbreviations</b> .....
<b>6.</b>	<b>USB 2.0 compliancy</b> .....	<b>16</b>	<b>15.</b>	<b>References</b> .....
<b>7.</b>	<b>PN533 host link protocol</b> .....	<b>18</b>	<b>16.</b>	<b>Legal information</b> .....
7.1	Standard frame .....	18	16.1	Definitions.....
7.2	Extended frame.....	19	16.2	Disclaimers.....
7.3	NACK frame.....	20	16.3	Licenses .....
7.4	Error frame .....	20	16.4	Trademarks .....
7.5	USB communication details .....	21	<b>17.</b>	<b>List of figures</b> .....
<b>8.</b>	<b>I<sup>2</sup>C master interface</b> .....	<b>22</b>	<b>18.</b>	<b>List of tables</b> .....
8.1	I <sup>2</sup> C EEPROM.....	22	<b>19.</b>	<b>Contents</b> .....
8.1.1	EEPROM data organization .....	24		
8.1.2	List of EEPROM tags .....	24		
8.2	I <sup>2</sup> C link to theTDA8029.....	25		
8.3	Typical application schematics.....	27		
<b>9.</b>	<b>PayPass/EMVCo compliancy</b> .....	<b>29</b>		
9.1	Protocol.....	29		
9.2	RF power and signal interface.....	29		
<b>10.</b>	<b>Radio Frequency Activation</b> .....	<b>31</b>		
<b>11.</b>	<b>NFC Secure</b> .....	<b>31</b>		
<b>12.</b>	<b>The PN533 commands</b> .....	<b>33</b>		
12.1	Global view.....	33		
12.2	The PN533 embedded software API: commands and errors lists.....	33		
12.2.1	Commands codes .....	33		
12.2.2	Errors codes.....	34		
12.3	The PN533 various modes.....	36		
12.3.1	How to use the PN533 as a reader for MIFARE Classic?.....	38		
12.3.2	How to use the PN533 as a T=CL reader (ISO/IEC 14443-4) .....	46		
12.3.3	How to use the PN533 as a DESFire cards reader? .....	49		

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