UM10463

PR533 Contactless Interface Controller

Rev. 1.3 — 11 January 2018
232013

User manual
COMPANY PUBLIC

Document information

Info	Content
Keywords	PR533, FW V3.60, CCID, PCSC, APDU
Abstract	This document describes the firmware V3.60 embedded in the chip PR533.

Revision history

Rev	Date	Description
1.3	20180111	Public release, no content change
1.2	20141110	Description of changes V360 – V370
1.1	20120704	Update chapter specific implementation and known limitations
1.0	20120605	First release

Contact information

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

1. Document purpose

This document describes the firmware functionalities of the PR533 chip.

The protocol used on the different possible physical links (USB or HSU) is defined as well as the framing and the possible host commands.

It describes also the global behavior of the PR533 device.

2. General presentation of the PR533

The embedded firmware and the internal hardware support the handling of the host controller protocol for USB and HSU interfaces.

The host controller protocol is defined in chapter §10 (p.25).

The firmware of the PR533 supports the following operating modes:

- PCD mode for FeliCa (212 kbps & 424 kbps), ISO/IEC14443 Type A & B (from 106 kbps to 847 kbps), MIFARE (106 kbps), B' cards (106 kbps), picoPass tag (106kBps) and Innovision Jewel cards (106 kbps);
- Initiator passive mode (from 106kbps to 424kbps) can be supported through the PC/SC transparent mode;
- The PR533 manages an I2C master interface.

  The PR533 is configured as master and is able to communicate with external EEPROM (address 0xA0) and with an external Contact smartcard reader (TDA8029).
- PR533 in PCD mode is compliant with EMV contactless specification V2.0.1 and BSI requirements.

In this document:

PR533 refers to PR533/V3.6

3. Version changes

3.1 PR533/C360 to PR533/C370

The following table lists and describes the changes that applied between version C360 and C370.

This version update only contains FW change. There is no update of the Hardware part.

Table 1. Changes C360/C370

Item	Description
EMVCo compliancy	C360 was compliant with EMVCo 2.1. C370 is compliant with EMVCo 2.2. !! PR533 C370 is not compliant with latest EMVCo specification V.2.3.1 !!
APDU Timing	Optimization of the transfer of APDU in C370: No more buffer transfer to send of RF the data received from Host interface → Optimization of transaction time
Authentication MIFARE 4k	Problem to access blocks over 1k has been solved. When accessing a block > 1k, a key enable the read/write of 16 blocks (and 4 blocks when less than 1k). The version C360 only allows to read/write 4 blocks.
RF field Reset when abort	In C360, when the PR533 was in the mode HSU and transparent mode, an abort command on host interface causes a RF Reset. This has been fixed in C370: no RF reset is seen in this case
Max HSU Response Frame	In C360 HSU mode, in case of a response frame of more than 259 bytes, the last byte was not sent. C370 now allows responses larger than 259 bytes.
Dual Card Management	Cards with an SAK showing more than one card (e.g. JCOP emulating MIFARE with SAK = 0x28) were not supported by PR533/C360: the card could be accessed in ISO14443-4 mode, but MIFARE emulation couldn't be used. This is corrected in C370 which can access MIFARE or ISO14443-4 card from such a dual card.
TDA timing	When the PR533 communicates with the TDA8029 over I2C, sometimes the timeout was reached even if the TDA8029 was about to send a message. In C370, the timeout has been adjusted to match with the delay in the worst case.
EEPROM parameters	In version C360, the parameters from EEPROM are not taken into account when the PR533 boots up. This has been fixed in C370: the parameters from the EEPROM are correctly applied when the PR533 starts.

4. Configuration Modes

4.1 Introduction

The PR533 has 3 possible modes that can be chosen by using two GPIOs during the reset phase of the IC:

Table 2. Configuration modes

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

4.2 Configuration modes transition

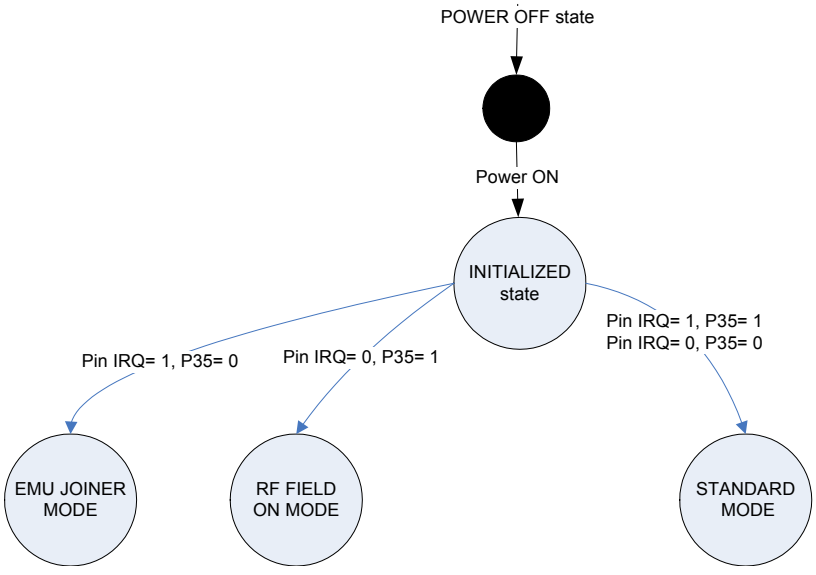


Fig 1. PR533 configuration modes

4.3 Standard Mode

This is the default mode of the PR533.

The description of this mode is detailed in this document starting from chapter §6 (p.8).

4.4 PN512 emulation mode

In this test mode, the PR533 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.

Then, the PR533 can be easily interfaced with the PN512 dedicated host controller software, as e.g. **Joiner PC Serial**.

The link used is RS232 at 9600 bauds[1]. It is not possible to change the value of the baud rate; the SerialSpeedReg register is not emulated.

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. The bit IRQInv in the register CommIEnReg has no effect (see [Datasheet]).

4.5 RFfieldON Mode

In this mode, the PR533 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/SIC_CLK and random data bytes are continuously sent.

In this mode, the temperature sensor is not activated, so that test 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/SIC_CLK** (pin #34)
MIFARE - 106 kbps	1	1
	0	0
FeliCa - 212 kbps	0	1
FeliCa - 424 kbps	1	0

1. [1] The RS232 link used here is the standard UART, not the High Speed UART. Consequently, in this mode the PR533 must be interconnected with P30 (pin#24) for the RS232_RX line and P31 (pin#31) for the RS232_TX line.

5. PR533 State Machine

5.1 Introduction

The device can switch between 5 different states when it is in STANDARD mode (see §5.3, p.7). Its behavior (functional and power consumption) will depend on the current state.

The 5 possible states are:

- Standby state,
- Suspend state (or Power Down state),
- ATD state,
- Transparent state,
- Reader state.

The transition between these states is explained hereafter.

5.2 Standby state

The Standby is the starting state after reset in HSU; the PR533 stays in this state unless it is commanded to go into any other mode.

Table 4. Transition from Standby state

Initial State	Action	Final State
Standby	USB: Receive RF Field ON apdu (see §0, p.78) HSU: N/A	**ATD**
Standby	Receive InActivateDeactivate apdu (see §13.2.9, p.86)	**Reader**
Standby	Receive the Open session command (see §13.2.6, p.54)	**Transparent**
Standby	USB : The USB link enters in suspend state HSU: Receive the HSU_Powerdown apdu (see §13.2.15, p. 95)	**Suspend**

After reset, the PR533 stays in Normal mode regarding the CPU, and the contactless interface Is switched OFF.

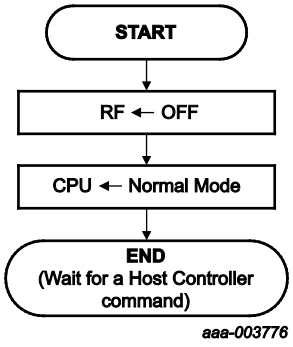


Fig 2. Standby state

5.3 Suspend state

The Suspend is the preferred state to save power consumption in USB and in HSU.

For USB link, PR533 enters in suspend mode when the link is in suspend state (according to reference §7.1.7.6 of [USB]).

For HSU link, PR523 enters in this state when the Power_Down apdu is received (see §13.2.15, p.95).

Table 5. Transition from Suspend state

Initial State	Action	Final State
Suspend	USB: link state exits from suspend ; HSU: Depend on Wake-up condition set (see §13.2.15, p. 95);	**Previous state**

In addition of the CPU and contactless front-end state, PR533 manage GPIOs configuration port in order to reduce the power consumption.

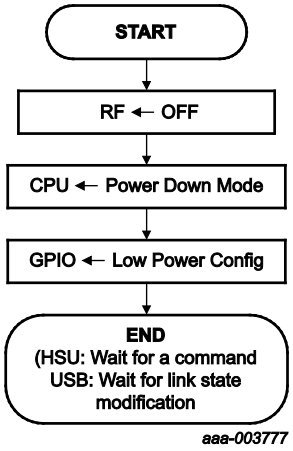


Fig 3. Suspend state

5.4 ATD State

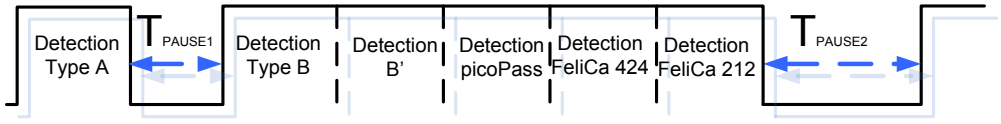
5.4.1 Description

This mode is used to activate a card or check its presence automatically. This mode is only applicable with the USB link.

Lastly, this is the initial state when a device boots with USB link.

The behaviour of the PR533 will depend on card presence:

- If no PICC is activated, this task handles the detection loop. This detection loop consists of several Reader phases and pauses[2] which can be enabled or disabled (except the phase Pause2 that cannot be disabled):
  - Reader type A (ISO14443-A and Jewel);
  - Pause 1
  - Reader ISO14443-4B;
  - Reader B';
  - Reader picoPass (only ISO14443-2B);
  - Reader FeliCa 424;
  - Reader FeliCa 212;
  - Pause 2



This detection sequence is fixed. As soon as a card is activated the PR533 shares the ATR of the card with the Host and start the presence check mechanism;

- If a card is activated, the PR533 will check the presence of the card. As soon as the card is removed from the reader, it will share this event with the Host and restart the detection loop;

5.4.2 Customization

The Automatic Tag Discovery is customizable.

The following points can be modified:

- Number of activation per technology;
- PAUSE2 duration;
- Some phases can be disabled;
- ATD can be also stopped.

All these actions can be performed with the **`Manage Reader`** command (see §13.2.8, p. 65).

2. [2] Pause means that the RF field is switched OFF

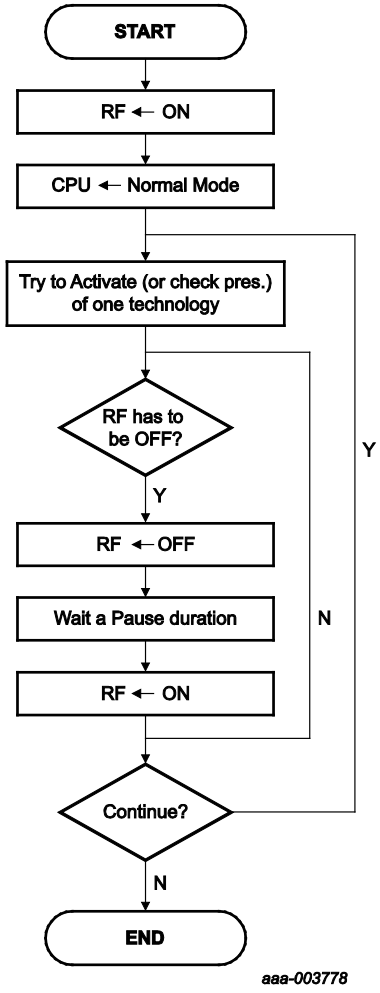


Fig 4. ATD mode

Table 6. Transition from ATD state

Initial State	Action	Final State
ATD	Receive RF Field OFF apdu (see §0, p.78)	**Standby**
ATD	Receive InActivateDeactivate apdu (see §13.2.9, p.86)	**Reader**
ATD	Receive the Open session command (see §13.2.6, p.54)	**Transparent**
ATD	The USB link enters in suspend state	**Suspend**

5.4.3 Reader state

This mode is the one used when the host application tries to activate a card without the ATD.

The **`InActivateDeactivateCard`** command uses this mode (see §13.2.9, p.86) to activate cards. In addition the verification of the card presence has to be done with the **`Manual Check Presence`** command (see §13.2.7.5, p.61).

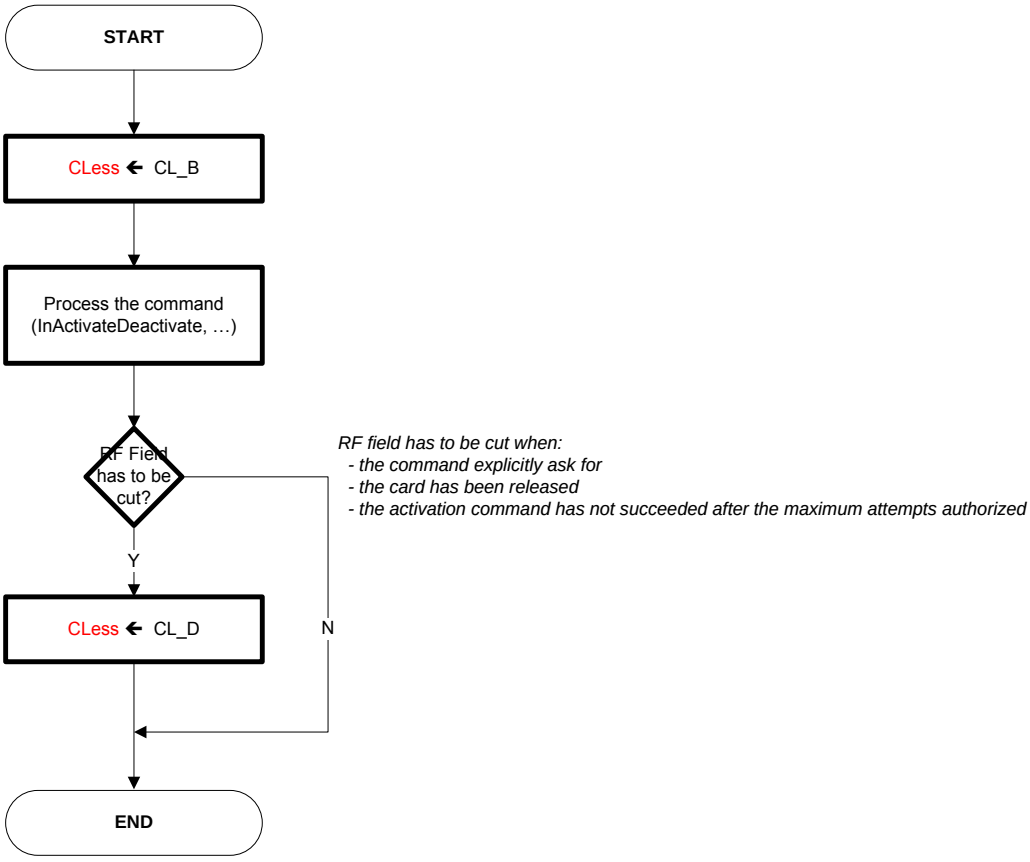


Fig 5. Reader state

Table 7. Transition from Reader state

Initial State	Action	Final State
Reader	Receive RF Field OFF apdu (see §0, p.78)	**Standby**
Reader	USB: Receive RF Field ON apdu (see §0, p.78) HSU: N/A	**ATD**
Reader	Receive the Open session command (see §13.2.6, p.54)	**Transparent**
Reader	USB : The USB link enters in suspend state HSU: Receive the HSU_Powerdown apdu (see §13.2.15, p. 95)	**Suspend**

5.5 Transparent state

This state is specified in the PCSC specification (see [PCSC3-sup2]). In this state, the Host application takes the control of the CL Reader.

Table 8. Transition from Transparent state

Initial State	Action	Final State
Transparent	USB: Receive the Open session command (see §13.2.6, p.54)	**ATD**
Transparent	HSU: Receive the Open session command (see §13.2.6, p.54)	**Standby**
Transparent	USB : The USB link enters in suspend state HSU: Receive the HSU_Powerdown apdu (see §13.2.15, p. 95)	**Suspend**

5.6 How to reduce power consumption

The design of the firmware embedded in the PR533 takes care of power consumption, in a sense that it minimizes the overall power consumption.

The configuration of the device can also decrease this power consumption.

We listed hereafter some of these parameters (see §13.2.8, p.65):

- RF_{pause} duration in the ATD state can be increased decreased;
- The number of retry during activation in Reader mode can be decreased;
- The external LED management can be disabled;
- Some reader phases can be disabled if there are useless.

6. PR533 architecture

6.1 Simplified block diagram

The PR533 can be connected to the host with either the USB link and the HSU link.

The device can control several interface:

- External LEDs;
- Contact card Reader (TDA8029);
- External EEPROM;
- RF communication.

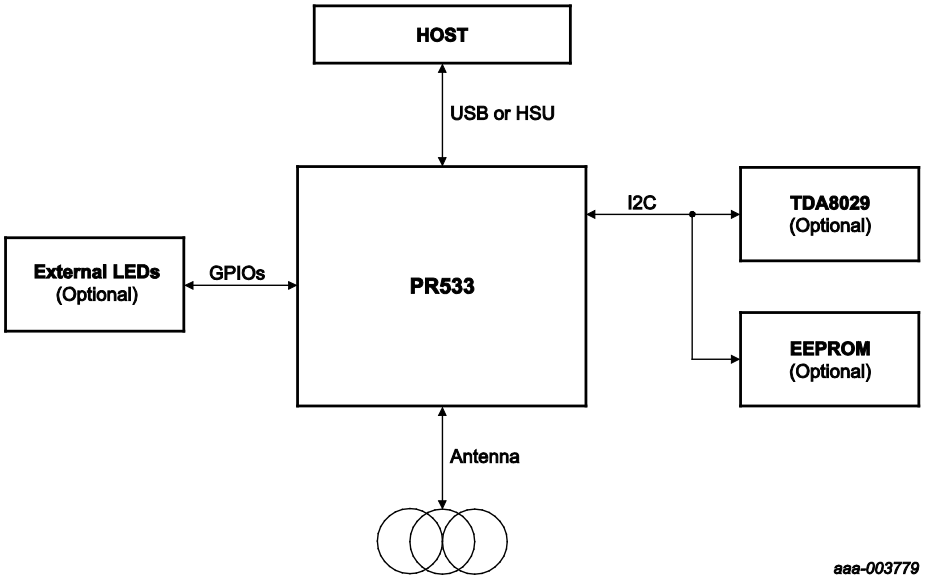


Fig 6. PR533 Simplified block diagram

6.2 Software architecture (only for USB)

The PR533 implements the CCID specification on the USB link. So Host applications use the PC/SC API to exchange frame with the device.

The figure below depicts how the system works.

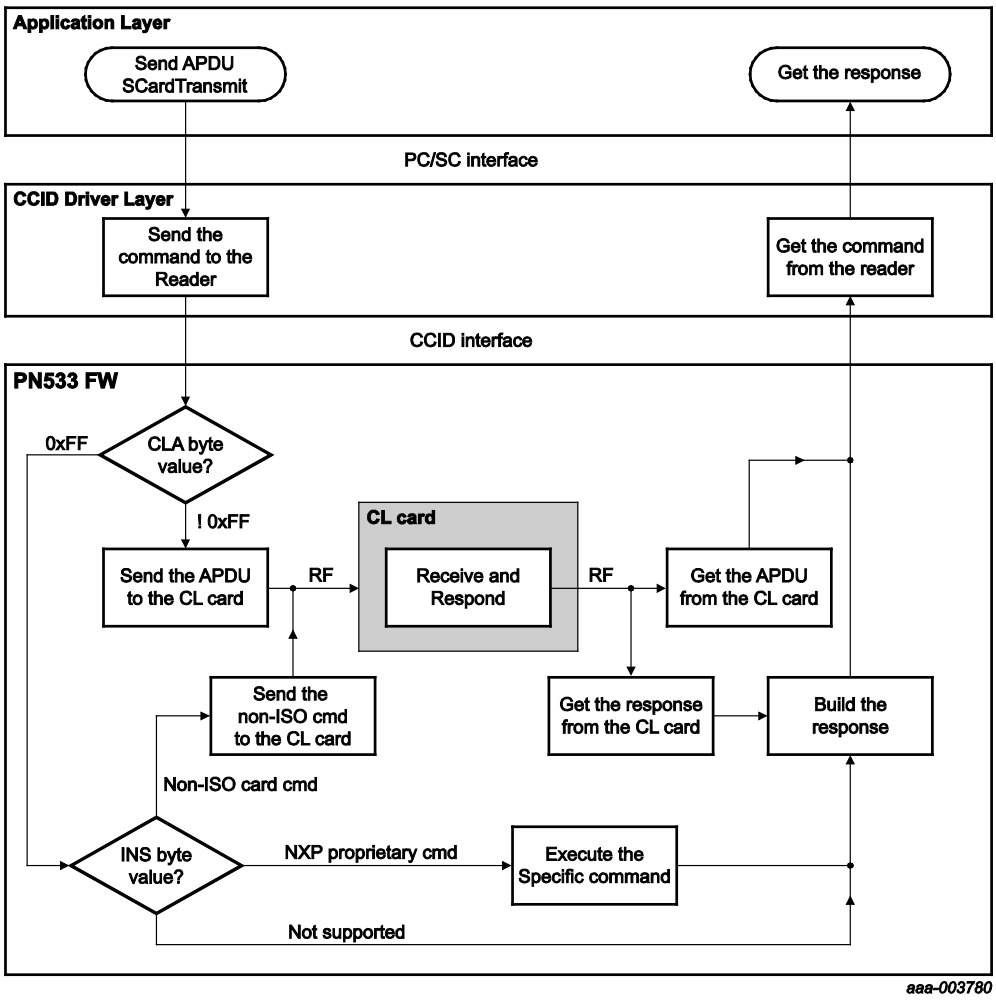


Fig 7. PR533 FW architecture

7. I2C master interface

The I2C master interface of the PR533 is compliant with the I2C bus specification (see reference document [I²C]).

The PR533 is configured as master on this I2C link and it's able to communicate with an external EEPROM (address 0xA0) and with the TDA8029 (contact card reader, address 0x50).

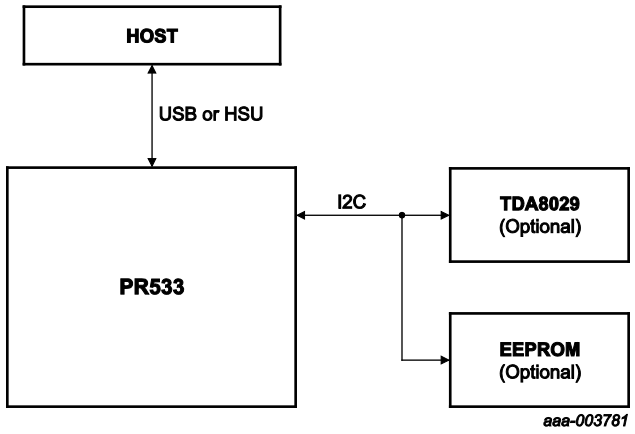


Fig 8. PR533 links overview

As specified in the I2C-bus specification, only two lines have to be used for managing the serial link between EEPROM and the host controller:

A serial data line (SDA) has to be connected on pin 33 of the PR533

And a serial clock line (SCL) has to be connected on pin 32 of the PR533.

The SCL frequency is set to 400 kHz for the transaction with the EEPROM.

7.1 External EEPROM mapping

7.1.1 EEPROM data organization

The TLV system is used for organizing data in EEPROM. Each block of data is preceded by its tag and its length.

The EEPROM mapping is shown below:

Fig 9. EEPROM mapping

The remaining free bytes at the end of the EEPROM have to be set to 0xFF. At least one 0xFF has to be present at the end of EEPROM mapping.

Each block of data has to be present only one time in EEPROM. The sequence of these blocks is not important.

If one of Tags 3, 4, 5, 6 and 7 is present then all these Tags have to be present.
If there is one error in the EEPROM mapping, then all these information are not used. In this case PR533 will boot with the default settings located in ROM code.

7.1.2 List of EEPROM tags

Tags allow PR533 to identify each block of data in the EEPROM. The list of EEPROM tags is the following:

- 0x01 ⇔ RF settings block,
- 0x02 ⇔ ATD configuration
- 0x03 ⇔ Fixed USB descriptor block,
- 0x04 ⇔ String Descriptor 0.
- 0x05 ⇔ Manufacturer ID String Descriptor,
- 0x06 ⇔ Device ID String Descriptor,
- 0x07 ⇔ Serial number String Descriptor,

7.1.2.1 RF settings blck description

This block contains settings of RF front-end for all supported RF protocols.

Byte #	Register	Size
0 … 10	*106kbps Type A* *(See Table 8, p.80)*	11
11 … 18	*212/424kbps Type FeliCa* *(See Table 9, p. 82)*	8
19 … 21	*Type B* *(See Table 10, p.83)*	3
22 … 30	*Up to 212kBps Type A* *(See Table 11, p. 84)*	9

7.1.2.2 ATD configuration block description

This block sets the behavior of the ATD. These parameters are detailed in the **`ATD_Configuration`** command (see Table 7, p.79).

Byte #	Field	Size
0	*Reader phase state*	1
1	*Pause duration*	1
2	*Max Retry Activation*	1

7.1.2.3 Fixed USB descriptor block description

This blocks contains the overall USB descriptor excepts the String descriptors.

Byte #	Field	Size
0 … 17	*Device Descriptor* *(Default: see Table 9, P.27)*	18
18 … 26	*Configuration Descriptor* *(Default: Table 10, p.27)*	9
27 …	*Interface Descriptor* *(Default: Table 12, p.28)*	9
35 … 88	*CCID class Descriptor* *(Default: Table 11, p.28)*	54
89 … 95	*BULK-OUT Endpoint Descriptor* *(Default: Table 13, p.28)*	7
96 … 102	*BULK-IN Endpoint Descriptor* *(Default: Table 14, p.29)*	7
103 … 109	*INT-IN Endpoint Descriptor* *(Default: Table 15, p.29)*	7

7.1.2.4 Device ID String Descriptor description

- String Descriptor (Device ID):

Byte #	Field	Size
0	*bLength*	1
1	*bDescriptor Type*	1
2	*bString* *(Unicode encoded string)*	N[3]

7.1.2.5 Manufacturer ID String Descriptor description

- String Descriptor (Manufacturer ID):

Byte #	Field	Size
0	*bLength*	1
1	*bDescriptor Type*	1
2	*bString* *(Unicode encoded string)*	N[4]

7.1.2.6 String Descriptor 0 description

- String Descriptor 0 (Specifying languages supported by the device):

Byte #	Field	Size
0	*bLength*	1
1	*bDescriptor Type*	1
2	*wLANGID[0]*	2

7.1.2.7 Serial String Descriptor description

- String Descriptor (Serial number):

Byte #	Field	Size
0	*bLength*	1
1	*bDescriptor Type*	1
2	*bString* *(Unicode encoded string)*	N[5]

3. [3] Due to RAM limitation the value of N has to be defined within 0 and 28 bytes.
4. [4] Due to RAM limitation the value of N has to be defined within 0 and 28 bytes.
5. [5] Due to RAM limitation the value of N has to be defined within 0 and 28 bytes.

7.2 I2C TDA8029

In addition to I2C specification, we shall use three other lines to manage Energy Saving Mode mechanism of the TDA8029 (refer to [AN10207-4]):

- **WakeUpSlave** line is used to wake up the TDA8029. It must be connected between INT1 (pin 30 of the TDA8029) and P31 (pin 29) of the PR533.
- **SlaveI2CMute** 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 must be connected between pin 24 of the TDA8029 and P33 (pin 31) of the PR533
- **Shut-down** line is used for entering 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 to set the pin SDWN_N to 1.

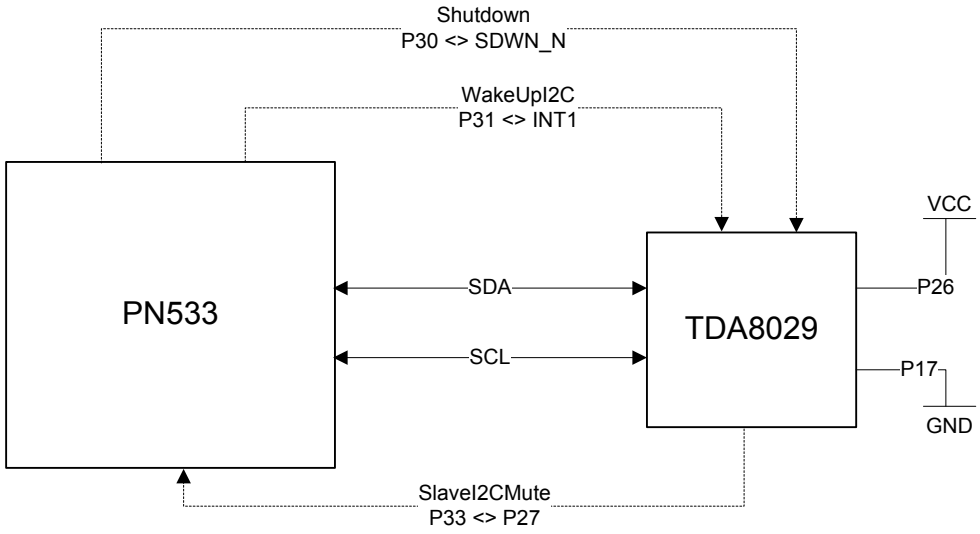


Fig 10. PR533 and TDA connection

One dedicated command is defined to access to the TDA8029 through the PR533.

- Refer to **`TDA_Communication`** command (§13.2.14, p. 93).

7.3 Read data in EEPROM

The PR533 is able to fetch a modified USB descriptor and modified RF settings from the external EEPROM during the startup of the IC (see §8.1.2, p.18).

Moreover, the host is able to read and write data in EEPROM through two specific commands.

Reading data in EEPROM

Refer to **`Manage Reader`** command (§13.2.8, p.65).

Writing data in EEPROM

Refer to **`Manage Reader`** command (§13.2.8, p.65).

.

8. External LED management

The reader can manage two external LEDs. They are connected as showed in the figure below.

By default, LEDs management is enabled in the FW. The host application can disable it by using the command **Manage Reader** (see §13.2.8, p.65).

LEDs are managed with respect to the following description:

Device Power-up:

When the PR533 is plugged, LED1 is turned ON and LED2 is turned OFF.

PR533 is enumerated:

If the device is well enumerated by the computer, LED1 is turned OFF and LED2 is turned ON.

Host application is connected:

If an application succeeds to establish a PC/SC-connection to the driver and a tag is in the RF field (SHARED or EXCLUSIVE modes), the LED2 starts blinking and the LED1 remains in OFF state.

No card is removed or Host application is disconnected:

If the card is removed or if the application call the disconnect routine of PC/SC API, LED2 stops flashing and stays ON while the LED1 stays OFF.

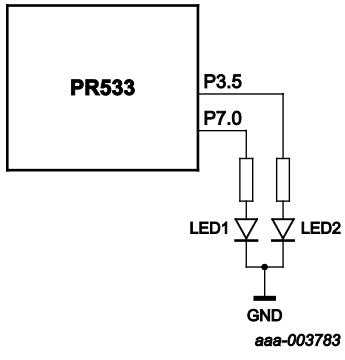


Fig 11. LEDs connection to the PR533 device

9. Host controller Interface

9.1 General points

9.1.1 Introduction

The system host controller communicates with the PR533 by using the USB or the HSU link. The protocol between the host controller and the PR533, on top of this physical link is the CCID protocol (see [CCID]). It's described in §12, p.43.

9.1.2 Possible links

Only one interface can be used to communicate between the system controller and the PR533. The choice of the interface is done by the firmware during the boot sequence and it depends on the level on the **I0**-**I1** pins (Interface modes lines, see [Datasheet]).

	Interface Selection pin	
	I0	**I1**
HSU	0	0
	0	1
USB (self powered)	1	0
USB (bus powered)	1	1

Fig 12. Host Interface selection

9.1.3 USB interface

9.1.3.1 PR533 USB model

The figure below shows an USB modeling of the PR533:

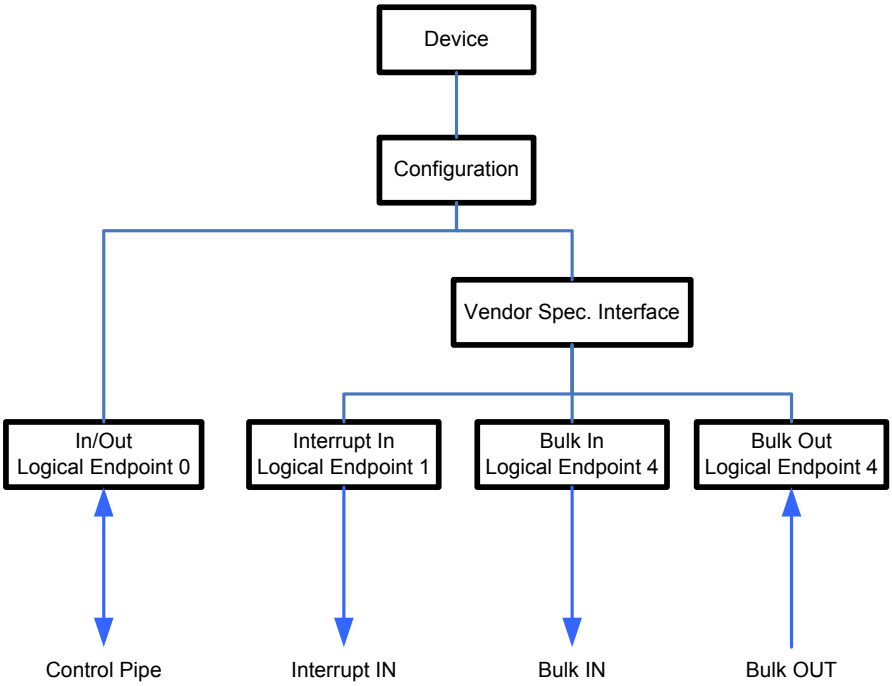


Fig 13. USB description of PR533

PR533 uses 3 endpoints which are part of the Vendor Specific Interface in addition to two mandatory default endpoints control 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

Logical endpoint 4, **bulk out**:

- This endpoint performs transfers to supply data to the PR533

Logical endpoint 4, **bulk in**:

- This endpoint performs transfers to retrieve data from the PR533

Logical endpoints 1, **interrupt in**:

- This endpoint notifies the Host that a card is detected by the PR533 or has been removed from the PR533 in ATD mode

Logical endpoints 2 and 3, **interrupt in**:

- Not used

9.1.3.2 Default USB descriptors

USB descriptors report the attributes of an USB device. They are data structures with a fixed format defined in the document Universal Serial Bus Specification.

The default descriptors of the PR533 are listed below:

Table 9. Device Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	12h	18 bytes of descriptor length
1	bDescriptorType	1	01h	**Device descriptor**
2	bcdUSB	2	0200h	--
4	bDevice Class	1	00h	--
5	bDevice Subclass	1	00h	--
6	bDevice Protocol	1	00h	--
7	bMax Packet Size	1	08h	--
8	idVendor	2	1FC9h	--
10	idProduct	2	010Bh	--
12	bcdDevice	2	0100h	--
14	iManufacturer	1	01h	--
15	iProduct	1	02h	--
16	iSerialNumber	1	03h	--
17	bNumConfigurations	1	01h	--

Table 10. Configuration Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	09h	9 bytes of descriptor length
1	bDescriptorType	1	02h	**Configuration descriptor**
2	wTotalLenght	2	005Dh	--
4	bNumInterfaces	1	01h	--
5	bConfigurationValue	1	01h	--
6	iConfiguration	1	00h	--
7	bmAttributes	1	80h	--
8	bMaxPower	1	32h	--

Table 11. Smart Card Device Card Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	36h	Size of this descriptor
1	bDescriptorType	1	21h	**Functional descriptor**
2	bcdUSB	2	0110h	CCID Spec. 1.1 compliant
4	bMaxSlotIndex	1	00h	1 slot
5	bVoltageSupport	1	07h	5V, 3V and 1.8V supported
6	dwProtocol	4	00000003h	Protocols T=0 and T=1
10	dwDefaultClock	4	00000E65h	--
14	dwMaxClock	4	000034F8h	--
18	bNumClockSupported	1	00h	--
19	dwDataRate	4	000026B5h	--
23	dwMaxDataRate	4	000CF080h	848 kbps
27	bNumDataRatesSupported	1	00h	--
28	dwMaxIFSD	4	000000FEh	254 bytes
32	dwSynchProtocols	4	00000000h	--
36	dwMechanical	4	00000000h	--
40	dwFeatures	4	0004047Eh	Short & Extended APDU
44	dwMaxCCIDMessageLength	4	00000111h	273 bytes
48	bClassGetResponse	1	FFh	Echo the class value
49	bClassEnvelope	1	FFh	Default class value 0
50	wLcdLayout	2	0000h	No LCD support
52	bPinSupport	1	00h	No pin support
53	bMaxCCIDBusySlots	1	01h	Only one slot

Table 12. Interface Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	09h	9 bytes of descriptor length
1	bDescriptorType	1	04h	**Interface descriptor**
2	bInterfaceNumber	1	00h	--
3	bAlternateSetting	1	00h	--
4	bNumEndpoints	1	03h	3 endpoints available
5	bInterfaceClass	1	0Bh	Smart card class device
6	bInterfaceSubClass	1	00h	Subclass code
7	bInterfaceProtocol	1	00h	Protocol code
8	iInterface	1	00h	--

Table 13. Endpoint 4 Descriptor IN

Offset	Field	Size	Value	Description
0	bLength	1	07h	7 bytes of descriptor length
1	bDescriptorType	1	05h	**Endpoint descriptor**
2	bEndpointAddress	1	84h	Physical Ept #4, type IN
3	bmAttributes	1	02h	BULK endpoint
4	wMaxPacketSize	2	0040h	64 bytes of max. packet size
6	bInterval	1	04h	4 ms

Table 14. Endpoint 4 Descriptor OUT

Offset	Field	Size	Value	Description
0	bLength	1	07h	7 bytes of descriptor length
1	bDescriptorType	1	05h	**Endpoint descriptor**
2	bEndpointAddress	1	04h	Physical Ept #4, type OUT
3	bmAttributes	1	02h	BULK endpoint
4	wMaxPacketSize	2	0040h	64 bytes of max. packet size
6	bInterval	1	04h	4 ms

Table 15. Endpoint 1 Descriptor INTERRUPT IN

Offset	Field	Size	Value	Description
0	bLength	1	07h	7 bytes of descriptor length
1	bDescriptorType	1	05h	**Endpoint descriptor**
2	bEndpointAddress	1	81h	Physical Ept #1, type IN
3	bmAttributes	1	03h	INTERRUPT endpoint
4	wMaxPacketSize	2	0008h	8 bytes of max. packet size
6	bInterval	1	80h	128 ms

Remote WakeUp

The PR533 is an USB device and it does not support Remote WakeUp functionality.

9.1.4 HSU interface

Refer to the PR533 data sheet (see [Datasheet]).

HSU interface default configuration is:

Data bit : 8 bits,
Parity bit : none,
Stop bit : 1 bit,
Baud rate : 115 200 bauds,
Data order : LSB first.

Table 16. Pin used for HSU interface

	PR533 Pin number
HSU_RX	25
HSU_TX	24

9.2 Host controller communication protocol

9.2.1 Frames structure

Communication between the host controller and the PR533 is performed through frames, which respects the CCID specification (see [CCID]).

9.2.1.1 Frame format on HSU link

With the HSU, CCID frame are encapsulated within specific format to ensure the reliability of this link. This format is applicable for incoming and out coming frame.

The amount of data that can be exchanged using this frame structure is limited to 274 bytes.

The structure of this frame is the following:

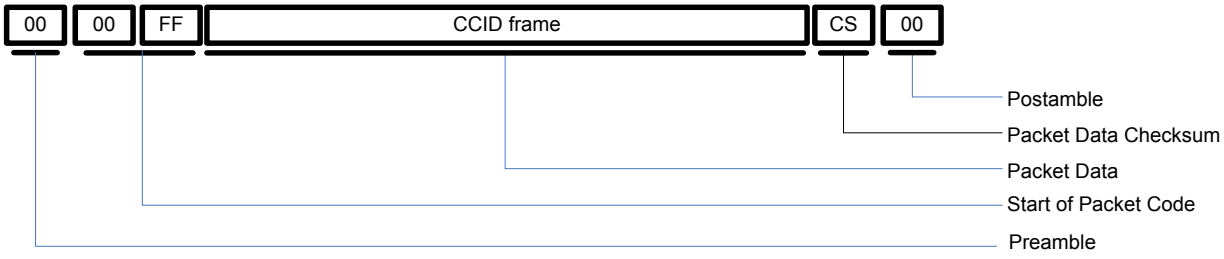


Fig 14. Normal information frame

- **START CODE** 2 bytes (0x00 and 0xFF),
- **DATA** The CCID frame like in USB,
- **DCS** 1 Data Checksum DCS byte that satisfies the relation: CCID_D0 + … + CCID_Dn + DCS = 0x00,
- **POSTAMBLE** 1 byte (0x00).

9.2.1.2 ACK frame on HSU link

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

This frame may be used from the PR533 to the host controller to indicate that the previous frame has been successfully received.

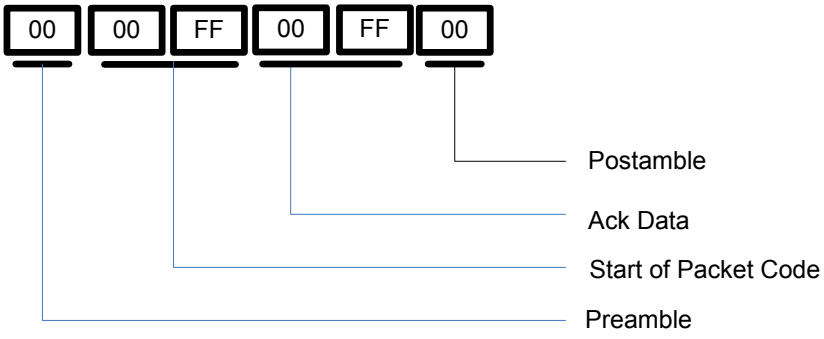


Fig 15. Ack frame

9.2.1.3 Frame format on USB link

With the USB link no additional byte is added. PR533 link respects CCID requirements (see [CCID]).

9.2.2 Dialog structure

The following chapters explain the dialog structure, whatever the physical link used.

The host controller is always the master of the complete exchange:

- It sends a command to the PR533,
- The PR533 acknowledges the command (only with HSU link),
- The PR533 executes the command,
- The PR533 sends back the corresponding answer to the host controller,

9.2.2.1 Data link level

a) Successful exchange at data link level

The figure below describes a normal exchange:

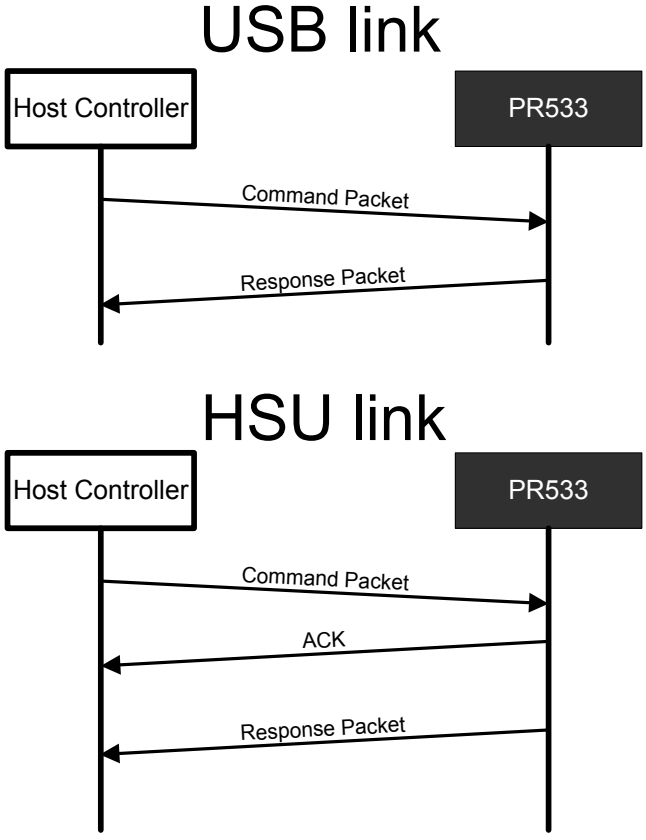


Fig 16. Data link level: normal exchange

b) Error at data link level, from host controller to PR533 with HSU link

When an error is detected by the PR533 at the data link level, it does not do anything.

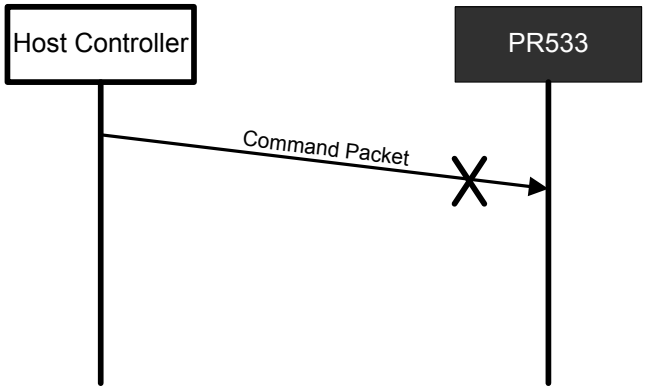


Fig 17. Data link level: error from the host controller to the PR533

The following errors are considered by the PR533 as data link level errors:

- Framing error (stop bit is at logic level 0);
- CS error;
- Timeout error:
  The PR533 detects a timeout error if the complete frame is not received within a time interval corresponding to four times the duration of a 256-bytes length frame with the current baud rate used. The timeout detection starts after the reception of the Start bytes.

  Thus the timeout values for all the possible baud rates are:

Table 17. HSU timeout values

Baud Rate	1-byte duration (µs)	256-bytes duration (ms)	Timeout value (ms)
9 600	1 041,7	266,7	1 067
19 200	520,8	133,3	533
38 400	260,4	66,7	267
57 600	173,6	44,4	178
115 200	86,8	22,2	89
230 400	43,4	11,1	44
460 800	21,7	5,6	22
921 600	10,9	2,8	11
1 288 000	7,8	2,0	8

9.2.2.2 Application level

a) Successive exchanges

The host controller sends a new command after having received the answer of the previous one.

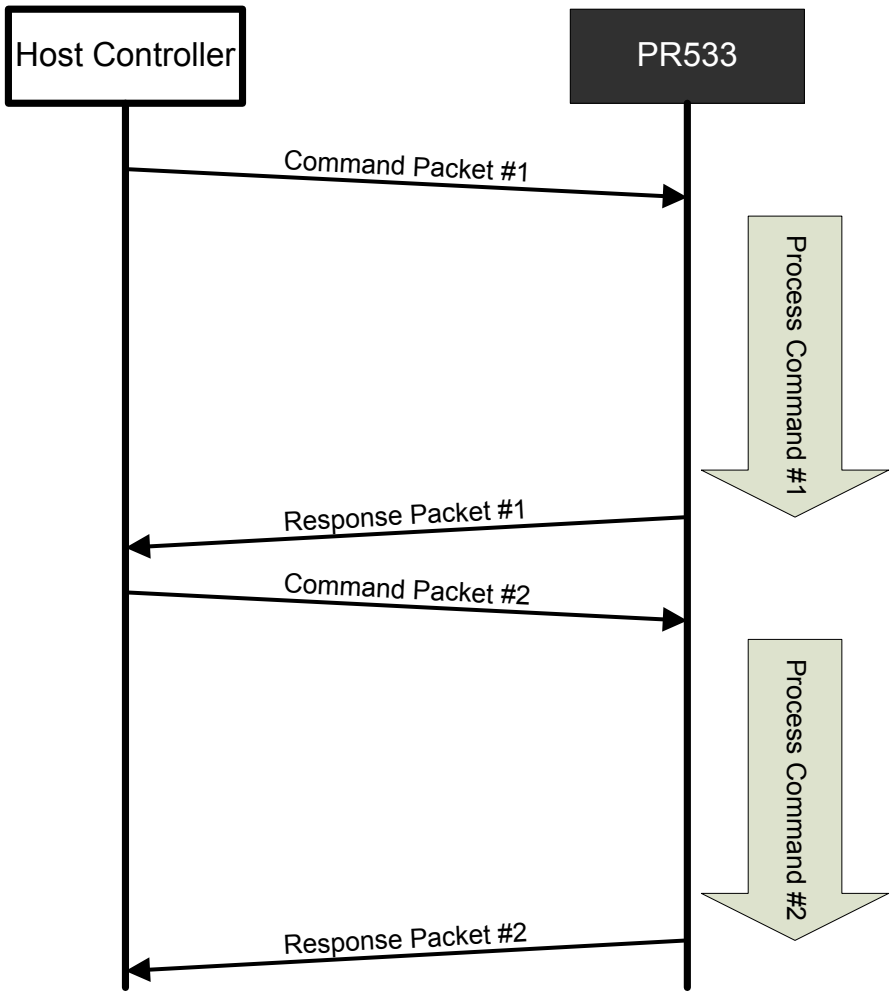


Fig 18. Application level: Successive exchanges

Note: Only the USB view is depicted.

b) Abort

The host controller can force the PR533 to abort an ongoing process. The behavior is different from HSU to USB.

In USB mode:

If the PR533 receives a command before having answered to the previous one, it stops the current process and start processing the new command received. It will send only the response to the last command.

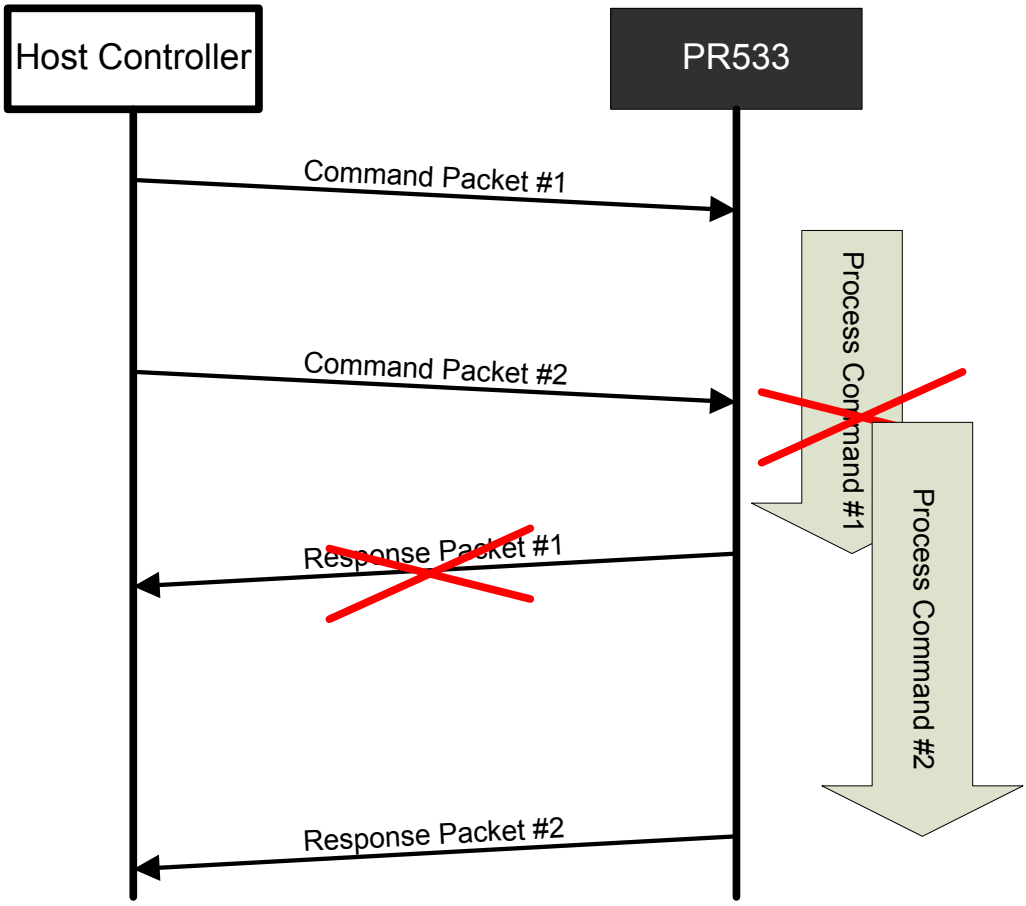


Fig 19. Application level: Abort a command and process a new one in USB mode

In HSU mode:

In HSU, the host can abort the current command by sending an ACK frame.

Actually any frame will abort the current command, but a command frame used as abort will not be executed.

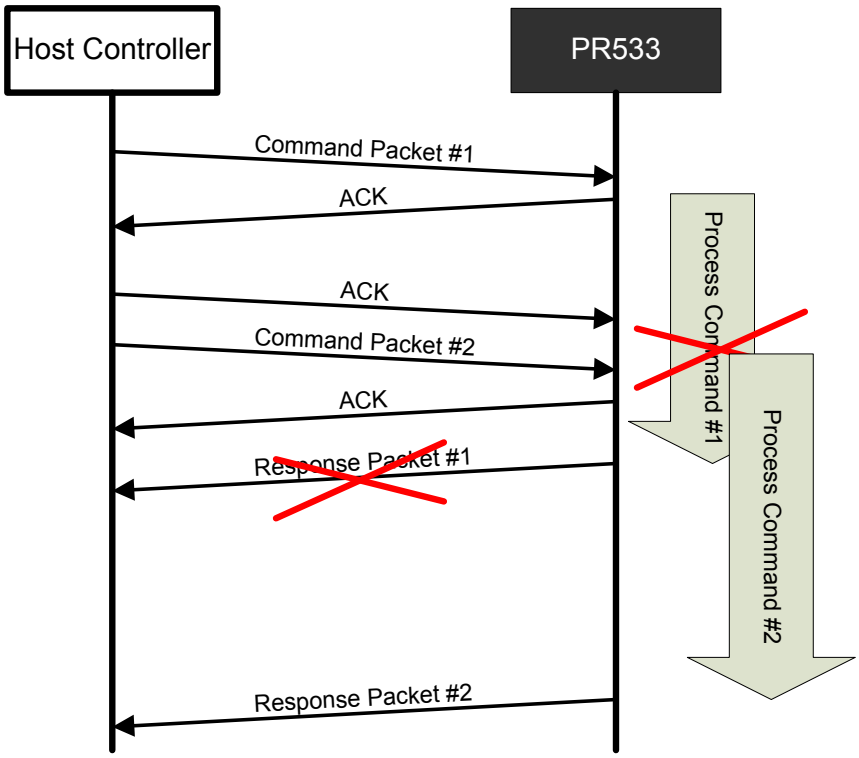


Fig 20. Application level: Abort a command and process a new one in HSU

c) Error at application level

When the PR533 detects an error at the application level, it sends back the specific “Syntax Error frame” to the host controller.

An application level error may be due to one of the following reasons:

- Unknown **Command Code** sent by the host controller in the command frame,
- Unexpected frame length,
- Incorrect parameters in the command frame.

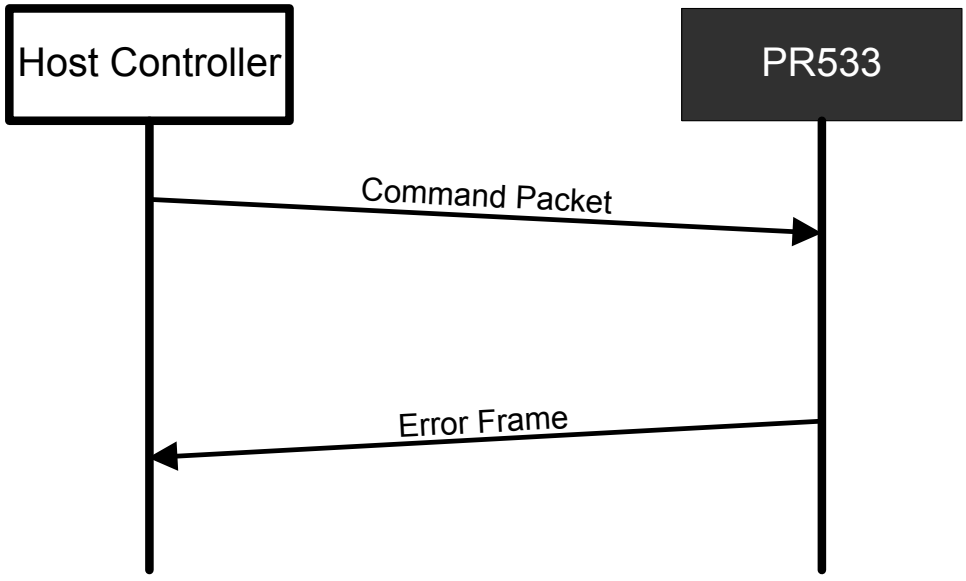


Fig 21. Application level: Error detected

Note: Only the USB view is depicted.

9.2.3 USB communication details

The USB device interface of the PR533 is built around:

- A Control Endpoint 0 (8 bytes IN/ 8 bytes OUT),
- An INTERRUPT IN Endpoint (8 bytes),
- 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.

The frames used when communicating with the USB are exactly the same as defined in the previous paragraphs §10.2.1.

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

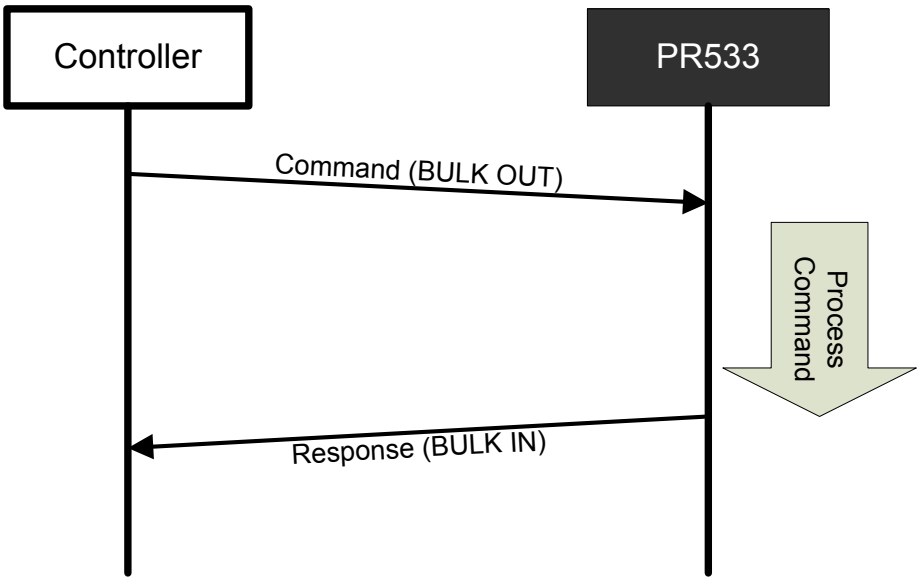


Fig 22. USB link: general principle of communication

9.2.4 HSU communication details

The HSU interface of the PR533 is a full duplex serial port capable of communicating with a host controller with a baud rate up to 1.288 Mbaud.

The PR533 receives the host controller command on its HSU_RX pin and transmits the response to the host controller on its HSU_TX pin.

The frames used when communicating with the HSU are exactly the same as defined in the previous paragraphs §10.2.1.1.

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

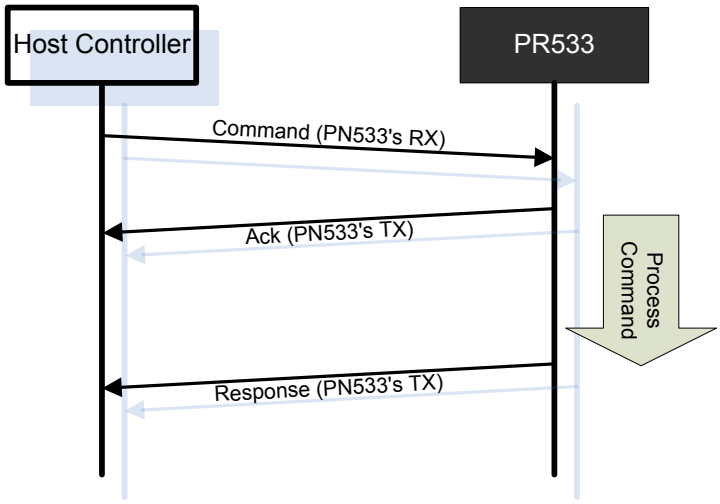


Fig 23. HSU link: general principle of communication

10. PC/SC Interface

10.1 Introduction

PC/SC interface is a specification for contact and contactless smartcard into the computer eco-system. When the PR533 is plugged with the USB link, the device is controlled using this interface on top of the USB driver.

10.2 Supported operation systems for USB link

The PR533 is compliant with the CCID transport protocol. It does not need a native driver. The device uses the driver from the operating system!

The PR533 supports the following Microsoft Windows version:

- Microsoft Windows 2000;
- Microsoft Windows XP (32 & 64 bits);
- Microsoft Windows 2003 Server (32 & 64 bits);
- Microsoft Windows 2008 Server (32 & 64 bits);
- Microsoft Windows Vista (32 & 64 bits);
- Microsoft Windows 7 (32 & 64 bits);

The PR533 supports the following OS through the PCSC-Lite driver[6]:

- GNU/Linux using libusb 1.0.x and later;
- Mac OS Leopard (1.5.6 and newer);
- Mac OS Snow Leopard (1.6.X);
- Solaris;
- FreeBSD;

6. [6] See http://pcsclite.alioth.debian.org/ccid.html

10.3 PC/SC API routines

The table below lists API routines supported by USB drivers.

Table 1. PC/SC API routines support

API routines	WinScard[7]	PCSC-Lite[8]
SCardEstablishContext	X	X
SCardReleaseContext	X	X
SCardIsValidContext	X	X
SCardListReaders	X	X
SCardListReaderGroups		X
SCardConnect	X	X
SCardReconnect	X	X
SCardDisconnect	X	X
SCardBeginTransaction	X	X
SCardEndTransaction	X	X
SCardTransmit	X	X
SCardControl	X	X
SCardStatus	X	X
SCardGetStatusChange	X	X
SCardCancel	X	X
SCardSetTimeOut		X
SCardGetAttrib	X	X
SCardSetAttrib	X	X

7. [7] See http://msdn.microsoft.com/EN-US/library/aa374731.aspx#smart_card_functions
8. [8] See http://pcsclite.alioth.debian.org/pcsc-lite/pcsc-lite.html

10.3.1 Enable ESCAPE IOCTL

Under Windows, this IOCTL is mandatory for **`SCardControl`** routines. It can be enable under the OS to allow some operations.

To enable the IOCTL, the key EscapeCommandEnable must be created and set to 1 under the following location of the Windows registry:

- For windows XP:
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB\VID_1FC9&PID_010B\3.60\Device Parameters
- For Windows 7:
  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB\VID_1FC9&PID_010B\3.60\Device Parameters\WUDFUsbccidDriver

The full procedure to enable this IOCTL is described in the PCSCTool User Manual.

10.4 ATR construction

The ATR construction is described in the PCSC specification (see §3.1.3.2.3 of [PCSC3] and [PCSC3-sup1][PCSC3-sup1]).

11. CCID transport protocol

11.1 Introduction

The PR533 uses the CCID protocol on USB link but also on HSU link. This protocol is compliant with USB Smart Card Device Class (see [CCID]).

11.2 Supported CCID feature

PR533 does not support all features of CCID specification.

You can find below the list of CCID commands. The column “Supports” informs if the command is supported or not on the PR533.

If the command is not supported the status in the response is set to command not supported.

Table 2. CCID Command set

Command	bMessage Type	Supports	See
PC_to_RDR_Icc_PowerOn	0x62	Yes	[CCID] §6.1.1
PC_to_RDR_Icc_PowerOff	0x63	Yes	[CCID] §6.1.2
PC_to_RDR_GetSlotStatus	0x65	Yes	[CCID] §6.1.3
PC_to_RDR_XfrBlock	0x6F	Yes	[CCID] §6.1.4
PC_to_RDR_GetParameters	0x6C	Yes	[CCID] §6.1.5
PC_to_RDR_ResetParameters	0x6D	No	[CCID] §6.1.6
PC_to_RDR_SetParameters	0x61	No	[CCID] §6.1.7
PC_to_RDR_Escape	0x6B	Yes	[CCID] §6.1.8
PC_to_RDR_IccClock	0x6E	Yes	[CCID] §6.1.9
PC_to_RDR_T0APDU	0x6A	No	[CCID] §6.1.10
PC_to_RDR_Secure	0x69	No	[CCID] §6.1.11
PC_to_RDR_Mechanical	0x71	No	[CCID] §6.1.12
PC_to_RDR_Abort	0x72	Yes	[CCID] §6.1.13
PC_to_RDR_SetDataRateAndClockFrequency	0x73	No	[CCID] §6.1.14

12. List of APDUs

12.1 Introduction

The PR533 receives an ISO7816-4 compliant APDU within a CCID frame.

With respect to PCSC specification (see [PCSC3]), the reader interprets this APDU.

12.2 APDU supported

Following chapters will give the description of each command.

This description contains:

- ⇨ The frame structure, including the type and the amount of data:
  - That the host application has to deliver to the PR533 (***Input***),
  - That the PR533 returns to the host application (***Output***).
  - When existing, the possible error causes (***Error Status Word***),
  - A description of the process attached to the command (***Description***).

For Input and Output data, optional bytes are written into square brackets (**[ ]**).

Table 3. Command set

PCSC Standard commands	INS	See
Load_Key	0x82	p.45
General_Authenticate	0x86	p.46
Get_Data	0xCA	p.47
Read_Binary	0xB0	p.48
Update_Binary	0xD6	p.51
Transparent_Mode	0xC2	p.54
NXP proprietary commands	**INS**	**See**
Diagnose	0xE0	p.56
Manage Reader	0xE1	p.65
InActivateDeactivateCard	0xE2	p.86
Mifare_Decrement	0xF0	p.88
Mifare_Increment	0xF1	p.89
Mifare_Transfer	0xF2	p.90
Mifare_Restore	0xF3	p.91
TDA_Communication	0xF4	p.93
HSU_Config	0xF5	p.95

12.2.1 Load_Key

The **Load key** command will load MIFARE keys in the PR533 RAM memory. These keys are used by the General Authenticate command (see §13.2.2, p.46).

Note: This command will load data in a volatile memory. So if the PR533 is turned OFF, the key will be lost. Moreover if the MIFARE card is deactivated (or lost) the MIFARE keys are automatically deleted.

Input:

Command	Class	INS	P1	P2	Lc	Data In
Load Keys	0xFF	0x82	Key Structure	Key Number	0x06	Key

- **Key Structure** : See [PCSC3] p.28;
- **Key Number** : Only 2 keys can be stored by the PR533[9];
  0x00 and 0x01 are allowed;
  Other values are RFU;
- **Key** : is an array of 6 bytes containing the key;

Output:

Response	
SW1	SW2

- **SW1 SW2** : See [PCSC3] table 3-12;

Example:

⇨ C-APDU: **FF 82 00 01 06 FF FF FF FF FF FF**

This command APDU loads the card key number 01 "FF FF FF FF FF FF" into the volatile memory of the PR533. The key is 6-bytes long.

⇦ R-APDU: **90 00**

The response APDU is 90 00 when the above C-APDU is executed successfully.

9. [9] Due to xRAM limitation in the PR533, we limit the number of key to 2.

12.2.2 General_Authenticate

The **`General Authenticate`** command will perform the Authenticate sequence on a MIFARE card.

This command is applicable on the following cards: MIFARE Mini, MIFARE 1k and MIFARE 4k.

Input:

Command	Class	INS	P1	P2	Lc	Data In
General Authenticate	0xFF	0x86	0x00	0x00	0x05	Data

- **Data** : 5 bytes

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0x01	Address MSB	Address LSB	Key Type	Key Number

- **Address** : represents the block number;
- **Key Type** : is the type of key to be used;
  - o **0x60:** MIFARE KEY_A;
  - o **0x61:** MIFARE KEY_B;
- **Key Number** : is the key number to be used for the authentication; It corresponds to the key number set with the Load Key command (see §13.2.1, p.45);

Output:

Response	
SW1	SW2

- **SW1 SW2**: See [PCSC3] table 3-14;

Example:

⇨ C-APDU: **`FF 86 00 00 05 01 00 04 60 01`**

This command performs an authentication on the block 4 using MIFARE key_A and the key number 1.

⇦ R-APDU: **`90 00`**

12.2.3 Get_Data

The **`Get Data`** command will retrieve information card information.

Input:

Command	Class	INS	P1	P2	Le
Get Data	0xFF	0xCA	See Below	0x00	0x00

The behavior of the PR533 will depend on the type of card which is activated. It will also depend on the P1 parameter. The following table specifies the content of the Data Out array:

P1	Data Out content
0x00	Serial Number of cards: • ISO14443-A: UID; • ISO14443-B: PUPI; • FeliCa: IDm; • JEWEL: RID; • B': DIV; • picoPass: SN;
0x01	Only for ISO14443-4 cards: • Type A: Historical bytes from ATS; • Type B INF field of ATTRIB resp;

Output:

Response		
[Data Out]	SW1	SW2

- **Data Out** : Response of the card;
- **SW1 SW2**: See [PCSC3] table 3-9;

Example:

- Get the UID of an ISO14443-4A card (MIFARE DESFire)

⇨ **`FF CA 00 00 00`**

⇦ **`04 4F 22 21 70 1C 80 90 00`**

12.2.4 Read_Binary

The **`Read Binary`** command will read data from a contact-less card. This card has to be activated before!

This command is applicable only for contactless storage cards (MIFARE, FeliCa, picoPass and Jewel/Topaz cards).

Input:

Command	Class	INS	P1	P2	[Lc]	[Data In]	[Le]
Read Binary	0xFF	0xB0	Address MSB	Address LSB	[Length Sent]	[Data]	[Length Expected]

The command frame will depend on the type of the activated card.

- **MIFARE Family:**

Command	CLA	INS	P1	P2	Le
UL READ 16	0xFF	0xB0	0x00	0x00 to 0x15	0x10
Classic 1k READ 16	0xFF	0xB0	0x00	0x00 to 0x3F	0x10
Classic 4k READ 16	0xFF	0xB0	0x00	0x00 to 0xFF	0x10

- **Picopass Family:**

Command	CLA	INS	P1	P2	Le
PicoPass cards	0xFF	0xB0	0x00	See [PICOPASS]	0x08

- **JEWEL Family:**

Command	CLA	INS	P1	P2	Le
READ ALL	0xFF	0xB0	0x00	0x00	0x00
RID	0xFF	0xB0	0x00	0x00	0x06
READ	0xFF	0xB0	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0x07	0x01
READ 4	0xFF	0xB0	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0x01	0x04
READ 8	0xFF	0xB0	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0xFF	0x08

READ SEG	0xFF	0xB0	0x00	Seg. Addr.: 0x00 to 0x03	0x80

- **FELICA Family:**

Command	CLA	INS	P1	P2	Lc	Data In
CHECK	0xFF	0xB0	0x00	0x00	Length of Data In	See below

o **Data In param:**

Element	Length	Data In content
Number of service	0x01	From 0x01 to 0x10.
Service code list	2 * Number of service(s)	
Number of blocks	0x01	From 0x01 to 0x0C
Block List	2 (or 3) * nb of blocks	

Output:

Response		
[Data Out]	SW1	SW2

- **Data Out** : Response of the card;
- **SW1 SW2** : See [PCSC3] table 3-21;

For all type except FeliCa:

In case of APDU framing error or RF transmission error, the standard PC/SC output (SW1 SW2) is returned.

For FeliCa only:

If no above APDU error occurs, then the two specific status flags contained in the FeliCa response are handled, which may modify the output in the following way:

- Case 1 : No error (i.e. status-flag1 = status-flag2 = 0)

  Number of blocks (1B) + Block data (16xNumber of blocks) + SW (2B)
- Case 2 : Error (i.e. status-flag1 != 0 and status-flag2 != 0)

  Status-Flags (2 bytes) + 6F08 (2 bytes), where SW1=6F, SW2=08 meaning the access to the targeted FeliCa card has not been or may not be granted.

Example:

- ⇨ MIFARE READ 16 bytes from block 05
  - ⇨ C-APDU: **FF B0 00 05 10**
  - ⇦ R-APDU: **94 D5 B0 46 6B 2A 4F B9 94 D5 B0 46 93 6C 93 6C 90 00**
- ⇨ PicoPass READ 8 bytes from block 01
  - ⇨ C-APDU: **FF B0 00 01 08**
  - ⇦ R-APDU: **12 FF FF FF F9 9F FF BC 90 00**
- ⇨ Jewel
  - ⇨ READ byte 1 from block 1 : **FF B0 01 01 01**
  - ⇨ READ4 bytes 4~7 from block 2 : **FF B0 02 01 04**
  - ⇨ READ8 block 2 : **FF B0 02 00 08**
- ⇨ FeliCa CHECK
  - ⇨ Read in RO access mode 2 blocks 0x00 to 0x01 from service 0 using 2-byte block numbering mode: **FF B0 00 00 08 01 0B 00 02 80 00 80 01**
  - ⇦ Response from FeliCa card with the 1st byte indicating the number of the blocks read: **02 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 90 00**

12.2.5 Update_Binary

The **`Update Binary`** command will try to write data in the activated card.

This command is applicable only for contactless storage cards (MIFARE, FeliCa, picoPass and Jewel/Topaz cards).

Input:

Command	Class	INS	P1	P2	Lc	Data In
Update Binary	0xFF	0xD6	Address MSB	Address LSB	Length	Data

The behavior of the PR533 will depend on the type of card which is activated.

- **MIFARE Family:**

Command	CLA	INS	P1	P2	Lc	Data In
UL WRITE 4	0xFF	0xD6	0x00	0x00 to 0x15	0x04	Data
Classic 1k WRITE 16	0xFF	0xD6	0x00	0x00 to 0x3F	0x10	Data
Classic 4k WRITE 16	0xFF	0xD6	0x00	0x00 to 0xFF	0x10	Data

- **Picopass Family:**

Command	CLA	INS	P1	P2	Lc	Data In
WRITE 8	0xFF	0xD6	0x00	See [PICOPASS]	0x08	Data

- **JEWEL Family:**

Command	CLA	INS	P1	P2	Lc	Data In
WRITE1-E	0xFF	0xD6	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0x07	0x01	Data
WRITE4-E	0xFF	0xD6	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0x01	0x04	Data
WRITE8-E	0xFF	0xD6	Block n°: 0x00 to 0xFF	Bytes Off.: 0x00 to 0xFF	0x08	Data

- **FELICA Family:**

Command	CLA	INS	P1	P2	Lc	Data In
UPDATE	0xFF	0xD6	0x00	0x00	Length	Data

o **Data In param:**

Element	Length	Data In content
Number of service	0x01	From 0x01 0x10.
Service code list	2 * Number of service	
Number of blocks	0x01	From 1 to 8
Block List	2 (or 3) * nb of blocks	
Block Data	16 * nb of blocks	

In case of APDU framing error or RF transmission error, the standard PC/SC output (SW1 SW2) is returned.

If no above APDU error occurs, then the two specific status flags contained in the FeliCa response are handled, which may modify the output in the following way :

- Case 1 : No error (i.e. status-flag1 = status-flag2 = 0)

  SW1 + SW2 (2 bytes) (i.e. the standard output)

- Case 2 : Error (i.e. status-flag1 != 0 and status-flag2 != 0)

  Status-Flags (2 bytes) + 6F08 (2 bytes), where SW1=6F, SW2=08 meaning the access to the targeted FeliCa card has not been or may not be granted.

Output:

Response	
SW1	SW2

- **SW1 SW2** : See [PCSC3] table 3-21;

Example:

- ⇨ MIFARE UL WRITE 4 bytes in block 06
  - ⇨ C-APDU: **FF D6 00 06 04 01 02 03 04**
  - ⇦ R-APDU: **90 00**
- ⇨ PicoPass WRITE 8 bytes in block 03
  - ⇨ C-APDU: **FF D6 00 03 08 01 02 03 04 05 06 07 08**
  - ⇦ R-APDU: **90 00**

    Note: The response 6F 01 means RF transaction timeout because the card is probably protected by keys.
- ⇨ Jewel WRITE
  - ⇨ WRITE byte 1 of block 2 :

    FF D6 02 01 01 66
  - ⇨ WRITE bytes 4~7 of block 2 :

    FF D6 02 01 04 11 22 33 44
  - ⇨ WRITE 8 bytes in block 2 :

    FF D6 02 00 08 11 22 33 44 55 66 77 88
- ⇨ FeliCa UPDATE
  - ⇨ Write blocks 0x0C, 0x0D of service 0 using 2-byte block numbering mode:

    FF D6 00 00 28 01 09 00 02 80 0C 80 0D 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20
  - ⇦ Response from FeliCa card: **FF 71 6F 08**, where the two FeliCa status flags FF71 indicate excessive writes to the FeliCa card (the memory has been written more than 0xFFFF times, and processing continues as this is not considered as an error). The PC/SC status code warns the access to the targeted FeliCa card may not be granted.

12.2.6 Transparent_Mode

The **`Transparent Mode`** command sets the device in manual mode. In this specific mode, the application can control the PR533. The ATD is stopped and the PR533 waits for a command from the reader.

Input:

Command	Class	INS	P1	P2	Lc	Data In
Transparent Mode	0xFF	0xC2	0x00	See Below	Length	See below

P2	Data In	Description
0x00	Data	Manage session command See [PCSC3-sup2] §4
0x01	Data	Transparent exchange command See [PCSC3-sup2] §5
0x02	Data	Switch Protocol command See [PCSC3-sup2] §6

Output:

Response		
[Data Out]	SW1	SW2

- **Data Out** : Response of the card;
- **SW1 SW2**: See [PCSC3-sup2] table 3;

Example:

- Start a transparent session

  ⇨ **FF C2 00 00 02 81 00**

  ⇦ **C0 03 00 90 00 90 00**

- Turn on the RF field

  ⇨ **FF C2 00 02 04 8F 02 00 04**

  ⇦ **C0 03 00 90 00 8F 01 00 90 00**

- Switching Protocol to Iso14443-3A

  ⇨ **FF C2 00 02 04 8F 02 00 03**

  ⇦ **C0 03 00 90 00 8F 01 00 90 00**

- Transparent Exchange command containing two BER-TLV data objects (Tag 90 for appending CRC, Tag 95 for transmitting & receiving data) sent to a MIFARE UL card for reading 16 bytes from block address 0A:

  ⇨ **FF C2 00 01 08 90 02 00 00 95 02 30 0A**

  ⇦ **C0 03 00 90 00 92 01 00 96 02 00 00 97 10 55 55 55 55 55 55 55 55 00 00 00 00 00 00 00 00 90 00**

12.2.7 Diagnose

The Diagnose command is designed for test purpose. You can find the description of this command below.

Input:

Command	CLA	INS	P1	P2	See
Com. Line Test	0xFF	0xE0	0x00	0x00	p.57
ROM Test	0xFF	0xE0	0x00	0x01	p.58
RAM Test	0xFF	0xE0	0x00	0x02	p.59
Polling Test Target	0xFF	0xE0	0x00	0x03	p.59
Check Presence	0xFF	0xE0	0x00	0x04	p.61
Antenna Self Test	0xFF	0xE0	0x00	0x05	p.63
RF Regulation Test	0xFF	0xE0	0x00	0x06	p.64

Output:

Response		
[Data Out]	SW1	SW2

Data Out : Depends on the command;

SW1 SW2: See [PCSC3-sup2] p.32;

12.2.7.1 Diagnose - Communication Line test

This test performs communication line test between host controller and the PR533. The input “Data” is entirely echoed back in the response packet.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Com. Line Test	0xFF	0xE0	0x00	0x00	m	Data

- **Lc** : this parameter is RFU; the APDU length has to be defined between 4 and 261 bytes;

Output:

Response		
Data Out	SW1	SW2

- **Data Out = Data In**

Example:

⇨ 5 bytes com. Line test:

⇨ C-APDU: **FF E0 00 00 00 01 02 03 04**

⇦ R-APDU: **00 01 02 03 04 90 00**

12.2.7.2 Diagnose - ROM test

This test checks the ROM data integrity.

It computes the 8 bits checksum of the overall ROM content and compares it to the checksum stored in the ROM.

If the sum of the two checksums are not zero, an error is raised. The stored checksum is the 2's complement of the calculated checksum of the masked ROM code.

Input:

Command	CLA	INS	P1	P2	Lc
ROM Test	0xFF	0xE0	0x00	0x01	0x00

Output:

Response		
Data Out	SW1	SW2

- **Data Out**: 3 bytes as below
  - Byte 0 : 0x00 ➔ OK, 0xFF ➔ NOK
  - Byte 1 : stored checksum
  - Byte 2 : calculated checksum

Example:

⇨ C-APDU: **FF E0 00 01 00**

⇨ R-APDU: **00 09 F7 90 00**

12.2.7.3 Diagnose - RAM test

This test checks the RAM data integrity (976 bytes in xRAM and 128 bytes in iDATA).

Input:

Command	CLA	INS	P1	P2	Lc
RAM Test	0xFF	0xE0	0x00	0x02	0x00

Output:

Response		
Data Out	SW1	SW2

- **Data Out**: 1 byte
  - Byte 0 : 0x00 ➔ OK, 0xFF ➔ NOK

Example:

⇨ C-APDU: **FF E0 00 02 00**

⇨ R-APDU: **00 90 00**

12.2.7.4 Diagnose - Polling test target

The purpose of this test is to check the RF performance in FeliCa reader mode.

The PR533 tries to activate a FeliCa card 128 times on the RF field. If a FeliCa polling response is received, the activation is succeeded.

During this test, the analog settings used are those defined in command **`RFConfiguration`** within the item n°7 (§13.2.8, p.65).

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Polling Test Target	0xFF	0xE0	0x00	0x03	0x01	Data

- **Data In** :
  - Byte 0: 0x01 ➔ 212 kbps
    0x02 ➔ 424 kbps

Output:

Response		
Data Out	SW1	SW2

- **Data Out** :
  - Byte 0 0x00 <= number of failures <= 0x80

Example:

⇨ Polling test FeliCa 424kbps:

⇨ C-APDU: **`FF E0 00 03 01 02`**

⇨ R-APDU: **`00 90 00`**

12.2.7.5 Diagnose - Manual Presence Check

This test can be used to ensure that a card is still in the RF field. It can be used if the ATD is enabled or disabled (see §0, p.78).

Input:

Command	CLA	INS	P1	P2	Lc
Check Presence	0xFF	0xE0	0x00	0x04	0x00

Output:

Response	
SW1	SW2

Description :

- In case of ISO/IEC14443-4 card, a R(NACK) block is sent to the card and it is expected to receive either a R(ACK) block or the last I-Block. In that case, the test is declared as successful (ISO/IEC14443-4 card is still in the RF field).
- In case of MIFARE UL card, a read command is sent to the card, and it is expected to receive a successful answer. In that case, the test is declared as successful.
- In case of MIFARE Classic card, two cases have to be considered :
    - If PR533 is not authenticated, we stored the MIFARE Classic UID. Then we poll for MIFARE Classic card. Once poll is done, PR533 has to compare the UID discovered and the UID of the remote device for which presence check is running. If there is no difference between UID, the test is declared as successful.
    - If PR533 is authenticated, PR533 has to know for which block number it is authenticated. Then PR533 send a read command on this block number. It is expected to receive a successful answer. In that case, the test is declared as successful.
- In case of FeliCa card, PR533 sends a SENSF_REQ command to the card. It is expected to receive a successful answer. In that case, the test is declared as successful.
- In case of JEWEL card, PR533 sends a RID command to the card. It is expected to receive a successful answer. In that case, the test is declared as successful.
- In case of B' card, PR533 sends a ATTRIB command to the card. It is expected to receive a successful answer (RR frame). In that case, the test is declared as successful.

- In case of picoPass card, PR533 asks the Serial Number of the card. It is expected to receive a successful answer (same SN than the stored one). In that case, the test is declared as successful.

In case of no or incorrect response, the response SW1 SW2 reports the status of the transaction.

12.2.7.6 Antenna Self test

This test checks the continuity of the transmission paths of the antenna. The RF is switched off at the end of this test.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Antenna Self Test	0xFF	0xE0	0x00	0x05	0x01	Data

- **Data in** :
  - Byte 0 : It contains one byte containing the threshold used for antenna detection (applied register Andet_Control @0x610C; see [Datasheet]).

.

7	6	5	4	3	2	1	0
Andet_bot	Andet_up	Andet_ithl[1;0]		Andet_ithh[1;0]			Andet_en

Output:

Response		
Data Out	SW1	SW2

- **Data Out** :
  - one byte containing test result that is relative to the used threshold. Application has to analyze the returned test result.
    - 0x00 for success
    - 0x80 for a too low power consumption detected
    - 0x40 for a too high power consumption detected

Example:

⇨ **FF E0 00 05 01 27**, with low current consumption threshold to be detected 0x20 -> 29mA, and high current consumption threshold to be detected: 011b -> 76mA

⇨ **00 90 00**, where 00 means test successful compared to the selected thresholds.

12.2.7.7 RF Regulation test:

This command executes the radio regulation in a loop till interruption by the protection timer. The later is set to 60s by default. The RF is switched off at the end of this test.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
RF Regulation Test	0xFF	0xE0	0x00	0x06	0x01	Data

- **Data in** :
  - Byte 0 : TxMode
    - It defines the bit rate and the framing to use for data transmission.

.

7	6	5	4	3	2	1	0
RFU	TxSpeed			RFU	RFU	TxFraming	
	• 000: 106kBps • 001: 212kBps • 010: 424kBps • 011: 847kBps					00 : MIFARE 10 : FeliCa	

Output:

Response	
SW1	SW2

Example:

⇨ Set a protection timer of 6s: **FF E1 05 00 02 02 58**

⇨ Response ok: **90 00**

⇨ Lancer RF regulation test for MIFARE 424kbps: **FF E0 00 06 01 20**

⇨ Response after about 6s: **6F 28**, meaning process aborted by the protection timer

12.2.8 Manage Reader

The **Manage Reader** enables the host application customize some parameters of the PR533.

Input:

Command	CLA	INS	P1	P2	See
GetFwVersion	0xFF	0xE1	0x00	0x00	p.66
Read GPIO	0xFF	0xE1	0x01	0x00	p.67
Write GPIO	0xFF	0xE1	0x01	0x01	p.68
Read Register	0xFF	0xE1	0x02	0x00	p.68
Write Register	0xFF	0xE1	0x02	0x01	p.71
SetParameter	0xFF	0xE1	0x03	0x00	p.72
Get Status	0xFF	0xE1	0x03	0x01	p.74
RF configuration	0xFF	0xE1	0x04	See Below	p.76
Protection Timer	0xFF	0xE1	0x05	0x00	p.85

Output:

Response		
[Data Out]	SW1	SW2

Data Out : Depends on the command;

SW1 SW2: See [PCSC3-sup2] p.32;

12.2.8.1 GetFwVersion

The device will send back the version of Firmware embedded in the PR533.

Input:

Command	CLA	INS	P1	P2	Le
GetFwVersion	0xFF	0xE1	0x00	0x00	0x04

Output:

Response		
Data Out	SW1	SW2

- **Data Out** : 4 bytes
  - Byte 0 : Version of the IC (default 0x33);
  - Byte 1 : Version of the hardware (default 0x03);
  - Byte 2 : Major revision of the firmware (default 0x60);
  - Byte 3 : Minor revision of the firmware.

Example:

⇨ C-APDU: **FF E1 00 00 04**

⇨ R-APDU: **33 03 60 11 90 00**

12.2.8.2 Read GPIO

The PR533 reads the value for each port and returns the information to the host controller.

Input:

Command	CLA	INS	P1	P2	Le
Read GPIO	0xFF	0xE1	0x01	0x00	0x03

Output:

Response		
Data Out	SW1	SW2

- **Data Out** : 3 bytes
  - The field **P3** contains the state of the GPIO located on the P3 port :

0	0	P35	P34	P33	P32	P31	P30
		5	4	3	2	1	0

  - The field **P7** contains the state of the GPIO located on the P7 port :

0	0	0	0	0	rfu	rfu	P70
					2	1	0

  - The field **I0I1** is reserved for future use **:**

0	0	0	0	0	0	rfu	Rfu
						1	0

Description:

The GPIOs may be used with the following limitations of usage:

- P32 corresponds to the pin P32_INT0.

  P32 can be used as standard GPIO and is therefore not used as external interrupt trigger.
- P33 corresponds to the pin P33_INT1.

  P33 can be used as standard GPIO and is therefore not used as external interrupt trigger.
- P71 and P72 are used as GPIO.
- I0 and I1 are not used

12.2.8.3 Write GPIO

The PR533 applies the value for each port that is validated by the host controller (bit **Val** of each port).

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Write GPIO	0xFF	0xE1	0x01	0x01	0x02	Data

- **Data In** :
  - The field **P3** contains the value to apply to the GPIO located on the P3 port :

Val	nu	P35	P34	rfu	P32	rfu	rfu
7	6	5	4	3	2	1	0

  - The field **is reserved for future use** :

Val	rfu	rfu	rfu	rfu	rfu	rfu	P70
7	6	5	4	3	2	1	0

Output:

Response	
SW1	SW2

Description:

For each port that is validated (bit **Val** = 1), all the bits are applied simultaneously. It is not possible for example to modify the state of the port P32 without applying a value to the ports P34 and P35.

As for the command **ReadGPIO** (see § 0), the GPIO may be used with the following limitations of usage:

- **P32** corresponds to the pin P32_INT0. It can be used as standard GPIO and is therefore not used as external interrupt trigger.
- **P34** can be used as standard GPIO.

Example:

The host controller wants to:

o set P32,

o reset P34 and P35,

o leave P70 unchanged.

The frame from host controller to PR533 is:

```
FF E1 01 01 02 84 00
```

The answer from PR533 is:

```
90 00
```

12.2.8.4 Read Register

This command is used to read the content of one or several internal registers of the PR533 (located either in the SFR area, in external EEPROM or in the XRAM memory space).

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Read Register	0xFF	0xE1	0x02	0x00	n*2	Data

- Lc : Length of the Data In is n*2 where n is the number of the registers to read
- Data In : List of addresses to read

$ADR1_H$	$ADR1_L$	…	$ADRn_H$	$ADRn_L$

$ADR1_H$ $ADR1_L$ First address (High and Low bytes)

$ADRn_H$ $ADRn_L$ n_{th} address (High and Low bytes)

Output:

Response		
Data Out	SW1	SW2

- Data Out : List of n values read in the chip;

VAL1	…	VALn

VAL1 Value read in the register located at address ADR1

VALn Value read in the register located at address ADRn

Description:

For each address ADR, the PR533 performs a reading operation in the register (located either in the SFR area or in the XRAM memory space) or in the external EEPROM at address ADR. Then the value is returned in the VAL parameter.

The table below shows information which is readable by ReadRegister Command.

ADR	Readable information
0x0000 to 0x03C7	XRam memory
0xA000 to 0xA0FF	EEPROM (128 or 256 bytes): (0xA0 00 address corresponds to 0x00 of the EEPROM)
0xFF80 to 0xFFFF	SFR (64 bytes)

Table 4. Read Register: memory mapping

- SFR registers.

The host controller has to set the High Byte of the address to 0xFF, the real address of the register is given by the low byte. The list of the SFR registers accessible for the host controller is configured in the firmware. The firmware gives access control to the following SFR registers:

Table 5. List of SFR registers

Address	Register	Address	Register	Address	Register
0x87	PCON	0xA3	FSIZE	0xD7	P5
0x9A	RWL	0xA8	IEN0	0xE8	IEN1
0x9B	TWL	0xAB	HSU_STATUS	0xF4	P7CFGA
0x9C	FIFOFS	0xAC	HSU_CONTROL	0xF5	P7CFGB
0x9D	FIFOFF	0xAD	HSU_PRESCALER	0xF7	P7
0x9E	SFF	0xAE	HSU_COUNTER	0xF8	IP1
0x9F	FIT	0xB0	P3	0xFC	P3CFGA
0xA1	FITEN	0xB8	IP	0xFD	P3CFGB
0xA2	FDATA	0xD1	CL_COMMAND		

- XRAM memory mapped registers

The complete address is given by the high and low bytes of address.
(see [Datasheet]).

Example:

The host controller reads the register *RX_THRESHOLD* located at address 0x6308, the *IEN1* register (SFR) located at address 0xFFE8 and the EEPROM at the address 0xA020:

⇨ C-APDU: **FF E1 02 00 06 63 08 FF E8 A0 20**

⇨ R-APDU: **85 00 DA 90 00**

12.2.8.5 WriteRegister

This command is used to overwrite the content of one or several internal registers of the PR533 (located either in the SFR area or in the XRAM memory space) or the content of one or several bytes of the EEPROM.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Write Register	0xFF	0xE1	0x02	0x01	n*3	Data

- **Lc** : Length of the Data In is n*3 where n is the number of the registers to write
- **Data In** : List of the addresses and values to write;

$ADR1_H$	$ADR1_L$	VAL1	…	$ADRn_H$	$ADRn_L$	VALn

- $ADR1_H$ $ADR1_L$ First address (High and Low bytes),
- VAL1 First value to be written,
- $ADRn_H$ $ADRn_L$ n^{th} address (High and Low bytes),
- VALn n^{th} value to be written.

Output:

Response	
SW1	SW2

Description:

For each address **ADR**, the PR533 performs a writing operation of the value **VAL** in the register or XRAM or EEPROM byte located at address **ADR**.

The WriteRegister memory mapping is the same as the one for ReadRegister (see Table 4 & Table 5).

Example:

The host controller writes 0x85 in the register *RX_THRESHOLD* located at address 0x6308, 0x00 in the *IEN1* register (SFR) located at address 0xFFE8 and 0x DA in the EEPROM at the address 0xA020:

⇨ C-APDU: **FF E1 02 01 09 63 08 85 FF E8 00 A0 20 DA**

⇨ R-APDU: **90 00**

Warning:

The behavior of the PR533 may be altered by this command. This command is only recommended for debug purposes.

12.2.8.6 SetParameter

This command is used to set internal parameters of the PR533, and then to configure its behavior regarding different cases.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
SetParameter	0xFF	0xE1	0x03	0x00	0x01	Data

- **Data In** :

RFU	RFU	5	4	3	RFU	1	0

- bit 0: fNADUsed	Use of the NAD information in case of ISO/IEC14443-4 PCD configuration *(Default value: 0)*.
- bit 1: fCIDUsed	Use of the CID information in case of ISO/IEC14443-4 PCD configuration *(Default value: 0)*.
- bit 2: RFU	**Must** be set to 0.
- bit 3: TDApowered	Power ON/OFF the TDA connected to the PR533 *(Default value: 0)***.**
- bit 4: fAutoRATS	Automatically send RATS to ISO type A card *(Default value: 1)*.
- bit 5: fAutoBrNegotiation	Automatic baudrate negotiation is enabled for ISO14443-4 type A and B cards *(Default value: 1)*.
- bit 6: RFU	**Must** be set to 0.
- bit 7: RFU	**Must** be set to 0.

Output:

Response	
SW1	SW2

Description:

fNADUsed (ISO/IEC14443-4 PCD mode):

By default, the PR533 does not use the NAD byte in the Transport Protocol, so the host controller must set this flag in order to use NAD.

In reception mode, the NAD value received by the PR533 will be transmitted to the PR533 host controller.

fDIDUsed (ISO/IEC14443-4 PCD mode):

By default, the PR533 does not use the CID byte for ISO/IEC14443-4 PCD in the Transport Protocol (not multi-target configuration). So the host controller must set this flag in order to use CID. In that case, CID value itself is completely handled internally by the firmware.

The CID has a fixed value of 0x01 when fDIDUsed is set to 1.

fTDApowered:

When the flag fTDApowered is set to one, the PR533 shall wake-up the TDA8029. The PR533 set the SHUTDOWN pin of the TDA to one.

When the flag fTDApowered is set to zero, the PR533 shall shutdown the TDA8029. The PR533 set the SHUTDOWN pin of the TDA to zero.

Refer to **`I2C TDA8029`** chapter (see §8.2, p.22) and the Host command: **`TDA_Communication`** (§13.2.14, p.93) to have more details on this functionality.

fAutoRATS:

If this flag is set to 1, when the PR533 activates a type A card which is compliant with ISO, the PR533 will send the RATS automatically.

If this flag is 0, the PR533 will never send RATS after receiving a SAK, even if the card is ISO compliant.

fAutoBrNegotiation:

By default, this bit is set to one for ISO/IEC14443-4 protocol. The PR533 automatically negotiates the maximum buadrate to use with the activated card via PPS command for type A card and ATTRIB for type B card.

If the user does not want to use this feature of the ISO/IEC14443-4 protocol with a card that is ISO/IEC14443-4 compliant, this flag must be set to 0.

12.2.8.7 Get Status

This command allows the host controller to know at a given moment the complete situation of the PR533.

Input:

Command	CLA	INS	P1	P2	Le
Get Status	0xFF	0xE1	0x03	0x01	0x05

Output:

Response		
Data Out	SW1	SW2

- **Data Out**: 5 bytes as below

Table 6. GetStatus

Byte #	Field	Description
Byte 0	Error	**Err** is an error code corresponding to the latest internal error detected by the PR533
Byte 1	NbTg	**NbTg** is the target number currently hold by the PR533 . It has to be equal to 0 for target absent or 1 for target present.
Byte 2	[BrRx]	• Bit rate in reception o 0x00 : 106 kbps o 0x01 : 212 kbps o 0x02 : 424 kbps o 0x03 : 847 kbps
Byte 3	[BrTx]	• Bit rate in transmission o 0x00 : 106 kbps o 0x01 : 212 kbps o 0x02 : 424 kbps o 0x03 : 847 kbps
Byte 4	[Type]	• Modulation type o 0x00: Innovision Jewel o 0x01 : ISO14443-A o 0x02: ISO14443-B o 0x03 : FeliCa 212/424kBps o 0x04: ISO14443-4A o 0x05: ISO14443-4B o 0x06: ISO14443-3B Prime o 0x07: PicoPass

Example:

⇨ C-APDU: **FF E1 03 01 05**

⇨ R-APDU: **00 01 02 02 04 90 00**

The response from PR533 indicates:

- o 00 : no internal error currently detected by PR533
- o 01 : one target detected in the RF field
- o 02 02 : baudrate is 424 kbps in both Tx & Rx directions
- o 04 : a tag of ISO14443-4A is activated

12.2.8.8 RFConfiguration

This command is used to configure the different settings of the PR533.

Input:

Command	CLA	INS	P1	P2	See
ISO14443-4 cards max baud rate	0xFF	0xE1	0x04	0x00	p.77
RF Field	0xFF	0xE1	0x04	0x01	p.78
ATD configuration	0xFF	0xE1	0x04	0x02	p.78
Analog settings for Type A 106kBps	0xFF	0xE1	0x04	0x0A	p.80
Analog settings for FeliCa 212/424kBps	0xFF	0xE1	0x04	0x0B	p.82
Analog settings for Type B	0xFF	0xE1	0x04	0x0C	p.83
Analog settings for ISO14443-4 from 212 to 847kBps	0xFF	0xE1	0x04	0x0D	p.84

Output:

Response	
SW1	SW2

ISO14443-4 cards max baud rate:

RFConfiguration allows setting a limit for the automatic baud rate negotiation. Default value is 0x22.

Command	CLA	INS	P1	P2	Lc	Data In
ISO14443-4 cards max baud rate	0xFF	0xE1	0x04	0x00	0x01	Data

- ⇨ **Lc**: 0x01
- ⇨ **Data In** :

Value	Description
0x00	106Kbps in both directions
0xX1	212Kbps from PCD to PICC
0xX2	424Kbps from PCD to PICC
0xX3	848Kbps from PCD to PICC
0x1X	212Kbps from PICC to PCD
0x2X	424Kbps from PICC to PCD
0x3X	847Kbps from PICC to PCD
Others	RFU

- ⇨ **Data Out**: Nothing
- ⇨ **Example**:

  Set the max baudrate to 848kbps in both directions: **`FF E1 04 00 01 33`**

RF Field

RFConfiguration allows switching **on** or **off** the RF field immediately and in the same time stopping or starting the ATD mechanism.

Command	CLA	INS	P1	P2	Lc	Data In
RF Field	0xFF	0xE1	0x04	0x01	0x01	Data

- ⇨ **Lc**: 0x01
- ⇨ **Data In**:
  - o If Data In value is set to 0x00, the ATD is stopped and the RF Field is turned OFF
  - o If Data In value is set to 0x01, the RF Field is turned ON and the ATD is started
  - o All other values are RFU
- ⇨ **Data Out**: Nothing

ATD configuration

This entry allows customizing the Automatic Tag Discovery wheel.

Command	CLA	INS	P1	P2	Lc	Data In
ATD configuration	0xFF	0xE1	0x04	0x02	0x05	Data

- ⇨ **Lc:** 0x05
- ⇨ **Data In:** See below
- ⇨ **Data Out:** Nothing
- ⇨ **Example**:

  Enabling only the Type A phase:

  FF E1 04 02 05 01 1E 01 01 03

  Enabling Type A, PAUSE1 and Type B' phases:

  FF E1 04 02 05 0B 1E 01 01 03

  For more details, refer to §6.4.

Table 7. ATD configuration

<table>
<tr><th>Byte #</th><th>Register</th><th>Description</th><th>Default</th></tr>
<tr><td>Byte 1</td><td>Reader Phase state</td><td>Indicates the status of the phases:
<table><tr><th>Bit</th><th>Technology</th></tr><tr><td>0</td><td>Type A</td></tr><tr><td>1</td><td>Pause_1</td></tr><tr><td>2</td><td>Type B</td></tr><tr><td>3</td><td>Type B’</td></tr><tr><td>4</td><td>Type picoPass</td></tr><tr><td>5</td><td>Type FeliCa 424</td></tr><tr><td>6</td><td>Type FeliCa 212</td></tr><tr><td>7</td><td>Pause_2</td></tr></table>0 -> disabled
1 -> enabled</td><td>0x7F</td></tr>
<tr><td>Byte 2</td><td>Pause Duration</td><td>Define the RFpause duration (value * 10ms)</td><td>0x1E</td></tr>
<tr><td>Byte 3</td><td>Max Retry Activation</td><td>Define the number of times that the PR533 will retry to activate a card.</td><td>0x01</td></tr>
<tr><td>Byte 4</td><td>Ext. LED Management</td><td>Enable or Disable the automatic external LED management:
- 0: Disable;
- 1: Enable;
- Others: RFU;</td><td>0x01</td></tr>
<tr><td>Byte 5</td><td>Type B SoF/EoF</td><td>Indicates the status of the phases:
<table><tr><th>Bit</th><th>Technology</th></tr><tr><td>0</td><td>EoFSoFAdjust</td></tr><tr><td>1</td><td>EoFSoFWidth</td></tr><tr><td>2</td><td>RFU</td></tr><tr><td>3</td><td>RFU</td></tr><tr><td>4</td><td>PreScaler * 2</td></tr><tr><td>5</td><td>RFU</td></tr><tr><td>6</td><td>RFU</td></tr><tr><td>7</td><td>RFU</td></tr></table>0 -> disabled
1 -> enabled</td><td>0x03</td></tr>
</table>

Remark:

- After reader phase state is modified, this byte only takes effect for the next card activation.

Analog settings for Type A 106kBps

This entry is used to modify the analog settings that the PR533 will use for the baud rate 106kbps type A.

Command	CLA	INS	P1	P2	Lc	Data In
Analog settings for Type A 106kBps	0xFF	0xE1	0x04	0x0A	0x0B	Data

When using this command, the host controller has to provide an array of 11 bytes in the **`Data In`** field.

- ⇨ **Lc**: 0x0B
- ⇨ **Data In**: See below
- ⇨ **Data Out**: Nothing

Table 8. Analog settings for the baudrate 106 kbps type A

Byte #	Register	Default values
Byte 1	CIU_RFCfg	0x59
Byte 2	CIU_GsNOn	0xF4
Byte 3	CIU_CWGsP	0x3F
Byte 4	CIU_ModGsP	0x11
Byte 5	CIU_DemodRfOn *when own RF is On*	0x4D
Byte 6	CIU_RxThreshold	0x85
Byte 7	CIU_DemodRfOff *when own RF is Off*	0x61
Byte 8	CIU_GsNOff	0x6F
Byte 9	CIU_ModWidth	0x26
Byte 10	CIU_MifNFC	0x62
Byte 11	CIU_TxBitPhase	0x87

Note:

Actually, there is only one **Demod** register which defines a setting used by the reader in reception only. But depending on the RF condition, two different settings can be used for this register:

- **CIU_Demod when own RF is On** defines a setting when its RF field is on during a reception i.e. initiator passive mode,
- **CIU_Demod when own RF is Off** defines a setting when its RF field is off during a reception i.e. initiator active mode.

It is the same case for the **GsN** register:

- **CIU_GsnOn** defines a setting to be used by a reader (or a target) when the RF field is on.
- **CIU_GsnOff** defines a setting to be used by a reader (or a target) when the RF field is off.

Analog settings for FeliCa 212/424kBps

The entry is used to choose the analog settings that the PR533 will use for baud rates 212/424kbps.

Command	CLA	INS	P1	P2	Lc	Data In
Analog settings for FeliCa 212/424kBps	0xFF	0xE1	0x04	0x0B	0x08	Data

When using this command, the host controller has to provide an array of 8 bytes in the Data In field.

- **Lc**: 0x08
- **Data In**: See below
- **Data Out**: Nothing

Table 9. Analog settings for the baudrate 212/424 kbps

Byte #	Register	Default values
Byte 1	CIU_RFCfg	0x69
Byte 2	CIU_GsNOn	0xFF
Byte 3	CIU_CWGsP	0x3F
Byte 4	CIU_ModGsP	0x11
Byte 5	CIU_DemodRfOn *when own RF is On*	0x41
Byte 6	CIU_RxThreshold	0x85
Byte 7	CIU_DemodRfOff *when own RF is Off*	0x61
Byte 8	CIU_GsNOff	0x6F

Note:

Actually, there is only one **CIU_Demod** register which defines a setting used by the reader in reception only. But depending on the RF condition, two different settings can be used for this register:

- **CIU_Demod when own RF is On** defines a setting when its RF field is on during a reception i.e. initiator passive mode,
- **CIU_Demod when own RF is Off** defines a setting when its RF field is off during a reception i.e. initiator active mode.

Analog settings for Type B 106kBps

The entry is used to choose the analog settings that the PR533 will use for the reader type B modulation at 106kBps.

Command	CLA	INS	P1	P2	Lc	Data In
Analog settings for Type B	0xFF	0xE1	0x04	0x0C	0x03	Data

When using this command, the host controller has to provide an array of 3 bytes in the Data In field.

- **Lc**: 0x03;
- **Data In**: See below;
- **Data Out**: Nothing;

Table 10. Analog settings for the type B

Byte #	Register	Default values
Byte 1	CIU_GsNOn	0xFF
Byte 2	CIU_ModGsP	0x12
Byte 3	CIU_RxThreshold	0x85

Except for these three specific settings, the 8 remaining analog settings are the same as the Analog settings for Type A 106kBps.

Analog settings for ISO14443-4 from 212 to 847kBps

The entry is used to choose the analog settings that the PR533 will use for the baud rates 212/424/847 kbps with ISO/IEC14443-4 cards.

Command	CLA	INS	P1	P2	Lc	Data In
Analog settings for ISO14443-4 from 212 to 847kBps	0xFF	0xE1	0x04	0x0D	0x09	Data

When using this command, the host controller has to provide an array of 9 bytes in the **`Data In`** field.

- **Lc**: 0x09;
- **Data In**: See below;
- **Data Out**: Nothing;

Table 11. Analog settings for the baudrate 212/424 and 847 kbps with ISO/IEC14443-4

Byte #	Register	Default values	Baudrate
Byte 1	CIU_RxThreshold	0x85	212 kbps
Byte 2	CIU_ModWidth	0x13	
Byte 3	CIU_MifNFC	0x8A	
Byte 4	CIU_RxThreshold	0x85	424 kbps
Byte 5	CIU_ModWidth	0x09	
Byte 6	CIU_MifNFC	0xB2	
Byte 7	CIU_RxThreshold	0x85	847 kbps
Byte 8	CIU_ModWidth	0x04	
Byte 9	CIU_MifNFC	0xDA	

Except for these three specific registers (CIU_RxThreshold, CIU_ModWidth and CIU_MifNFC), the 8 remaining analog registers are the same as the Analog settings for Type A 106kBps.

12.2.8.9 Protection Timer

This timer is started when a command is received. It is stopped when a response is generated by the PR533. The purpose of this timer is to unlock the WinScard API. It avoids that the PR533 performs endlessly one specific action, because it will not share the response. When it is elapsed the PR533 will generate a response on the USB link.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Protection Timer	0xFF	0xE1	0x05	0x00	0x02	Data

- Data In:
  - Time-out : This value specifies the time-out with steps of 10ms (default 6000, i.e. 60s);

***Notes**:*

- This timer is working only if USB link is used.
- The minimum duration is 2 seconds (i.e. Data=200)
- The maximum duration is about 10 minutes (i.e. Data=64000)

Output:

Response	
SW1	SW2

Example: Set a 10s protection timer

⇨ C-APDU: **FF E1 05 00 02 03 E8**

⇨ R-APDU: **90 00**

12.2.9 InActivateDeactivateCard

The **`InActivateDeactivateCard`** command is designed to put the PR533 in one specific reader mode. It is recommended to disable the ATD mechanism before executing this command (see the example below).

Input:

The behavior of the PR533 will depend on the following parameters:

Command Activate	CLA	INS	P1	P2	Lc
Type A ***(JEWEL and ISO14443 A)***	0xFF	0xE2	0x00	0x00	0x00
ISO14443 B	0xFF	0xE2	0x00	0x01	0x00
FeliCa 212kBps	0xFF	0xE2	0x00	0x02	0x00
FeliCa 424kBps	0xFF	0xE2	0x00	0x03	0x00
B'	0xFF	0xE2	0x00	0x04	0x00
picoPass	0xFF	0xE2	0x00	0x05	0x00
EMVco2.0.1	0xFF	0xE2	0x00	0x06	0x00

Command Deactivate	CLA	INS	P1	P2	Lc
InRelease	0xFF	0xE2	0x01	0x00	0x00
InReleaseEMVco	0xFF	0xE2	0x01	0x01	0x00

- **InRelease:**

  Releasing a card means that the host controller has finished the communication with it, so the PR533 erases all the information relative to it. This command is used whatever the target type and its current state (initialized, activated, deselected) is.

  The process depends on the type of card.

Table 12. InRelease RF actions

Card Type	Action
ISO/IEC14443-4 Type A card	• Send **DESELECT**
ISO/IEC14443-4 Type B card	• Send **DESELECT**
MIFARE card	• Send **HLTA**
FeliCa card	• No action

Card Type	Action
Innovision Jewel tag	• No action
B' card	• No action
picoPass tag	• No action

After this call, the PR533 automatically switched off its RF Field.

- **InReleaseEMVco:**

  The reader will perform a specific release sequence compliant with EMVco standard (see [EMVco]).

Output:

Response		
[Data Out]	SW1	SW2

- **Data Out** : Response of the card;
- **SW1 SW2** : See §15;

Example:

- Switch off the RF field & the ATD mode:

  ⇨ **FF E1 04 01 01 00**

  ⇦ **90 00**

- Activate an ISO14443A card:

  ⇨ **FF E2 00 00 00**

  ⇦ **90 00**

- Send a presence check command:

  ⇨ **FF E0 00 04 00**

  ⇦ **90 00**

- Deactivate the activated card:

  ⇨ **FF E2 01 00 00**

  ⇦ **90 00**

12.2.10 Mifare_Decrement

The **`MIFARE Decrement`** command will perform the Decrement MIFARE command.

Input:

Command	Class	INS	P1	P2	Lc	Data In
Mifare Decrement	0xFF	0xF0	Address MSB	Address LSB	0x04	Data

The behavior of the PR533 will depend on the type of card which is activated.

- ***MIFARE Family:***

Command	CLA	INS	P1	P2	Lc	Data In
Classic 1k Decrement	0xFF	0xF0	0x00	0x00 to 0x3F	0x04	Data
Classic 4k Decrement	0xFF	0xF0	0x00	0x00 to 0xFF	0x04	Data

- **Data:** Array of 4 bytes coded LSB first.

Output:

Response	
SW1	SW2

- **SW1 SW** : See below

SW1	SW2	Meaning
0x63	0x00	**No information**

Example: see 13.2.13

12.2.11 Mifare_Increment

The **`MIFARE Increment`** command will perform the INCREMENT Mifare command.

Input:

Command	Class	INS	P1	P2	Lc	Data In
Mifare Increment	0xFF	0xF1	Address MSB	Address LSB	0x04	Data

The behavior of the PR533 will depend on the type of card which is activated.

- ***MIFARE Family:***

Command	CLA	INS	P1	P2	Lc	Data In
Classic 1k Increment	0xFF	0xF1	0x00	0x00 to 0x3F	0x04	Data
Classic 4k Increment	0xFF	0xF1	0x00	0x00 to 0xFF	0x04	Data

- **Data:** Array of 4 bytes coded LSB first.

Output:

Response	
SW1	SW2

- **SW1 SW** : See below

SW1	SW2	Meaning
0x63	0x00	**No information**

Example: see 13.2.13

12.2.12 Mifare_Transfer

The **`MIFARE Transfer`** command will perform the TRANSFER Mifare command.

Input:

Command	Class	INS	P1	P2	Lc
Mifare Transfer	0xFF	0xF2	Address MSB	Address LSB	0x00

The behavior of the PR533 will depend on the type of card which is activated.

- **MIFARE Family:**

Command	CLA	INS	P1	P2	Lc
Classic 1k Transfer	0xFF	0xF2	0x00	0x00 to 0x3F	0x00
Classic 4k Transfer	0xFF	0xF2	0x00	0x00 to 0xFF	0x00

Output:

Response	
SW1	SW2

- **SW1 SW** : See below

SW1	SW2	Meaning
0x63	0x00	**No information**

Example: see 13.2.13

12.2.13 Mifare_Restore

The **`MIFARE Restore`** command will perform the RESTORE Mifare command.

Input:

Command	Class	INS	P1	P2	Lc
Mifare Restore	0xFF	0xF3	Address MSB	Address LSB	0x00

The behavior of the PR533 will depend on the type of card which is activated.

- **MIFARE Family:**

Command	CLA	INS	P1	P2	Lc
Classic 1k Restore	0xFF	0xF3	0x00	0x00 to 0x3F	0x00
Classic 4k Restore	0xFF	0xF3	0x00	0x00 to 0xFF	0x00

Output:

Response	
SW1	SW2

- **SW1 SW** : See below

SW1	SW2	Meaning
0x63	0x00	**No information**

Example:

- Mifare Load key N°1 (FF FF FF FF FF FF) into PR533 RAM

  ⇨ **`FF 82 00 01 06 FF FF FF FF FF FF`**

  ⇦ **`90 00`**

- Mifare generate authenticate on block 5 using key N°1

  ⇨ **`FF 86 00 00 05 01 00 05 60 01`**

  ⇦ **`90 00`**

- Mifare Write in block 5

  ⇨ **`FF D6 00 05 10 01 00 00 00 FE FF FF FF 01 00 00 00 05 FA 05 FA`**

⇦ **90 00**

- Mifare Read from block 5 (should find what is written above)
  - ⇨ **FF B0 00 05 10**
  - ⇦ **01 00 00 00 FE FF FF FF 01 00 00 00 05 FA 05 FA 90 00**

- Mifare Increment block 5 by 2
  - ⇨ **FF F1 00 05 04 02 00 00 00**
  - ⇦ **63 00**

- Mifare Transfer volatile memory into block 5
  - ⇨ **FF F2 00 05 00**
  - ⇦ **63 00**

- Mifare Read from block 5 (should read 03 00 00 00 FC FF FF FF 03 00 00 00 05 FA 05 FA)
  - ⇨ **FF B0 00 05 10**
  - ⇦ **03 00 00 00 FC FF FF FF 03 00 00 00 05 FA 05 FA 90 00**

- Mifare Restore block 5 into volatile memory
  - ⇨ **FF F3 00 05 00**
  - ⇦ **63 00**

- Mifare Transfer volatile memory into block 5
  - ⇨ **FF F2 00 05 00**
  - ⇦ **63 00**

- Mifare Read on Block 05 (should read 03 00 00 00 FC FF FF FF 03 00 00 00 05 FA 05 FA)
  - ⇨ **FF B0 00 05 10**
  - ⇦ **03 00 00 00 FC FF FF FF 03 00 00 00 05 FA 05 FA 90 00**

12.2.14 TDA_Communication

The **`TDA Communication`** command will perform the sending of proprietary APDU to the TDA.

Input:

Command	Class	INS	Data In
TDA Communication	0xFF	0xF4	Data

Output:

Response		
Data Out	SW1	SW2

- **Data Out** : Response of the TDA
- **SW1 SW2** : See §15

The standard APDU format is customized by removing P1 P2 Lc fields in order to fit to the maximum CCID message buffer size requirement. The length parameter Lc is deducted from the total CCID message length minus the CCID message header size.

To communicate with TDA8029, this customized APDU format encapsulates the ALPAR protocol frame that is briefly shown below. For more details on ALPAR protocol, please refer to [AN10207-4] in the Table 17

ACK/NACK byte	Length 2 bytes	Command byte	C/R-APDU	LRC

The ALPAR header has 4 bytes:

o ACK=60 or NACK=E0

o Length of C-APDU (PR533 to TDA8029) or R-APDU (TDA8029 to PR533)

o Command byte indicating the type of the command

An ALPAR frame is ended by a LRC byte (Longitudinal Redundancy Check).

It is decided that maximum 254 (0xFE) bytes of response data plus 2 status bytes SW1 SW2 can be read from TDA8029, which is 256 bytes in total. The frame size in bytes is accounted as the following.

⇨ PC/SC APDU command from Host to TDA8029:

```
2 (CLA INS) + 259 (4 for ALPAR Header + 254 (5 for APDU
Header + 249 data bytes) + 1 for LRC) = 261
```

⇨ PC/SC APDU response from TDA8029 to Host:

```
4 (ALPAR Header) + 257 (254 for response data + 2 for
SW1 SW2 + 1 for LRC) + 2 (SW1 SW2) = 263
```

Take into account the CCID header of 10 bytes, the maximum CCID message length is 273 bytes.

Example:

- Turn-on the TDA8029 with TDApowered (see SetParameter 13.2.8.6)
  - ⇨ **FF E1 03 00 01 38**
  - ⇦ **90 00**
- Activate the contact-card, an ATR is returned from TDA8029
  - ⇨ **FF F4 60 00 01 6E 00 0F**
  - ⇦ **60 00 0A 6E 3B 75 11 00 00 24 C2 01 90 00 2C 90 00**
- Select directory 4F 00
  - ⇨ **FF F4 60 00 07 00 00 A4 00 00 02 4F 00 8E**
  - ⇦ **60 00 02 00 90 00 F2 90 00**
- Write 16 bytes in contact-card
  - ⇦ **FF F4 60 00 15 00 00 D6 00 00 10 01 02 03 04 05 06 07 88 09 0A 0B 0C 0D 0E 0F 10 23**
  - ⇦ **60 00 02 00 90 00 F2 90 00**
- Read those 16 bytes written above
  - ⇨ **FF F4 60 00 05 00 00 B0 00 00 10 C5**
  - ⇦ **60 00 12 00 01 02 03 04 05 06 07 88 09 0A 0B 0C 0D 0E 0F 10 90 00 72 90 00**
- Deactivate the contact-card
  - ⇨ **FF F4 60 00 00 4D 2D**
  - ⇦ **60 00 00 4D 2D 90 00**
- Turn-off the TDA8029
  - ⇨ **FF E1 03 00 01 30**
  - ⇦ **90 00**

12.2.15 HSU_Config

The **HSU_Config** command will put, only in HSU mode, the PR533 in Power down state.

Input:

Command	CLA	INS	P1	P2	Lc	Data In
Power Down	0xFF	0xF5	0x00	0x00	0x01	WakeUpEnable
Set baudrate	0xFF	0xF5	0x00	0x01	0x01	BR

- **Power Down command:**

This command is applicable only in HSU mode. The aim is to decrease as much as possible the current consumption.

The IC will enter automatically in power down mode. It will wake-up on a command reception.

Different sources of wake up may be selected with this command with the **WakeUpEnable** parameter.

RFU	GPIO	RFU	HSU	RF Level Detector	RFU	INT1	INT0
7	6	5	4	3	2	1	0

Of course, it is possible to select more than one individual wake up source, and then the user may combine for example RFLevelDetector and HSU.

The “RFU” bit (bit 2, 5 and 7) **must** set be 0.

Remarks:

The PR533 needs approximately **1 ms** to go into Power Down mode, after the command response. Sending host commands during this time is not recommended

Wake up condition:

When the host controller sends a command to the PR533 on the HSU link in order to exit from Power Down mode, the PR533 needs some delay to be fully operational (the real waking up condition is the 5th rising edge on the serial line.

As a consequence, if the host controller wants to be sure that the command will not be lost or partially received, some precautions must be taken:

- Either send a command with large preamble containing dummy data, or
- Send first one 0x55 dummy byte and wait for the waking up delay ($T_{wake\ up\ time}$) before sending the command frame.

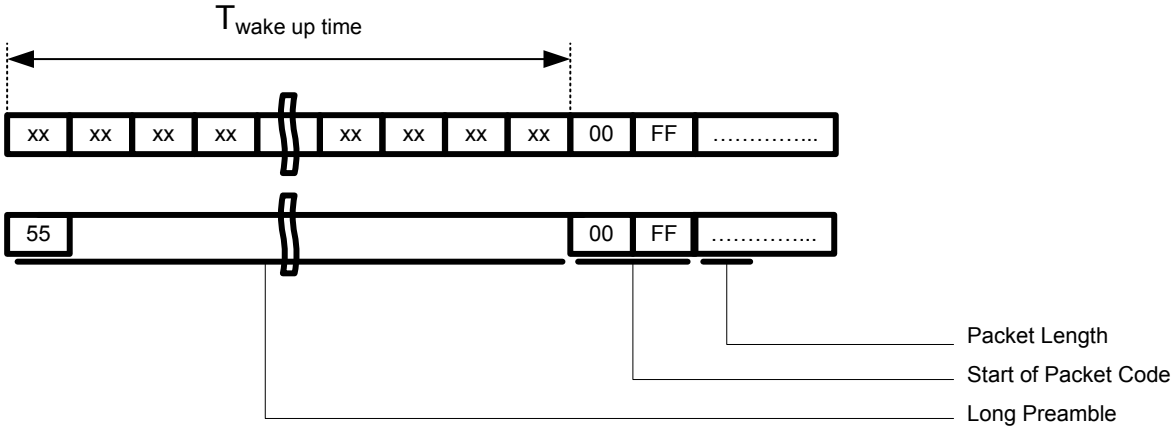


Fig 24. HSU Wake up

- **Set baudrate command:**

This command is used to change the baud rate on the HSU line. By default the IC will use a baud rate of 115200 Bps. But the Host can change it

BR is a byte indicating the baud rate requested by the host controller:

- 0x00 9.6 kbaud,
- 0x01 19.2 kbaud,
- 0x02 38.4 kbaud,
- 0x03 57.6 kbaud,
- 0x04 115.2 kbaud,
- 0x05 230.4 kbaud,
- 0x06 460.8 kbaud,
- 0x07 921.6 kbaud,
- 0x08 1.288 Mbaud.

When the USB link is used, this command is not allowed. In that case, an error code will be raised.

The PR533 changes the baud rate from the old one to the new one **only** after having sent the Response of the command **and** having received one ACK frame sent by the host controller.

This ACK frame is usually optional, but in the case of this specific set serial baud rate command, it is mandatory.

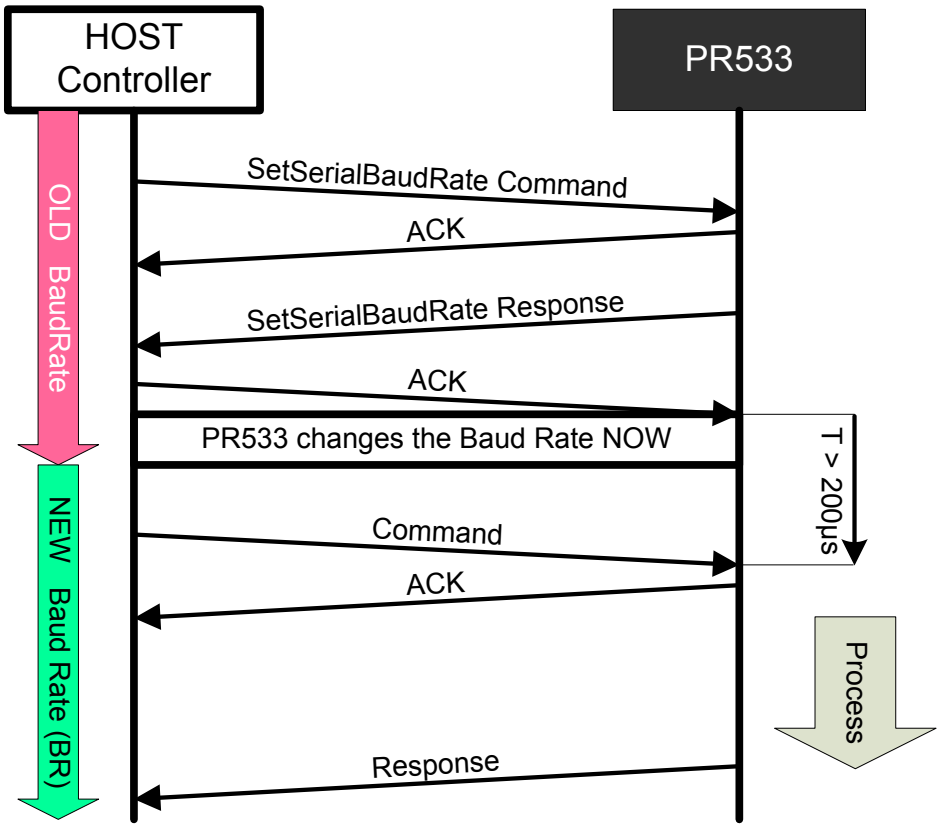


Fig 25. Set Serial Baud Rate

o The host controller shall send the next command at least 200µs after the ACK has been received.

Output:

Response	
SW1	SW2

- **SW1 SW2** : See §15

13. Example of use

13.1 PR533 in reader mode with the ATD

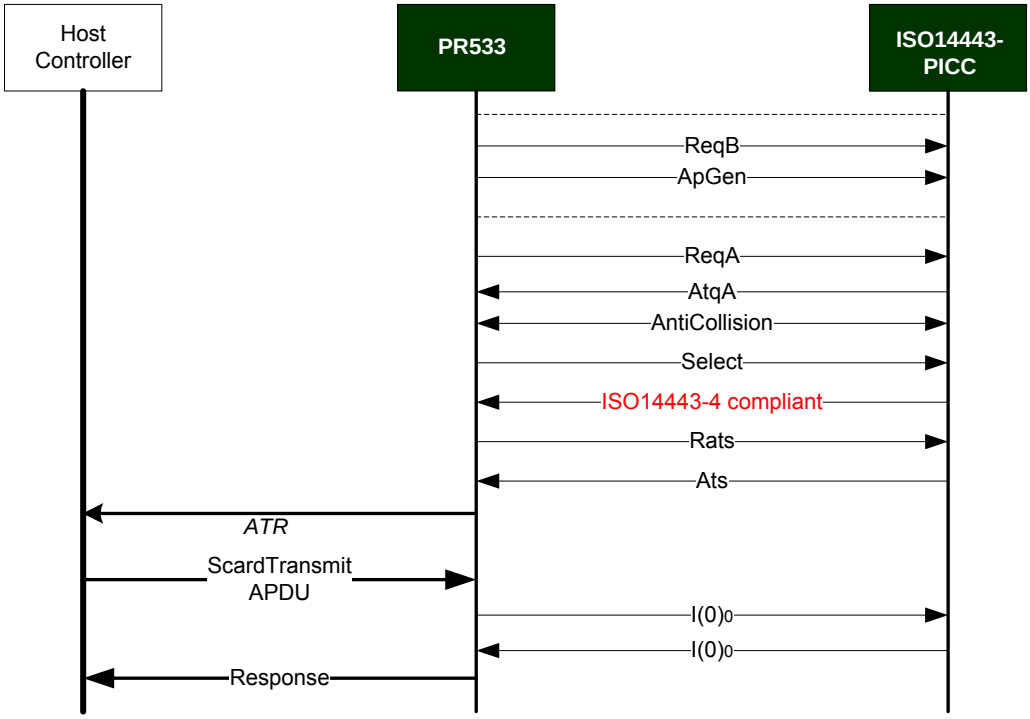


Fig 26. PR533 in reader mode

In this example, a card of type ISO14443-4A is detected and activated. The Host is informed of this by the ATR message. The Host then decides to exchange some information with the activated card.

14. Specific implementation

14.1 Parity handling in transparent mode

In transparent mode, PCSC requires that the parity is handled independently in RX and TX mode.

Due to the HW Architecture of the PR533, this management is not possible: We cannot dissociate parity in transmission and reception mode.

Then in the ‘Transmission and Reception Data Object’ (see [PCSC3-sup2]), bit 3 is not used. Only bit 2 is relevant for Parity, and is used for transmit and received data.

14.2 Chaining in HSU mode

In HSU mode, the chaining mechanism between the host and the PR533 follows the CCID specification: the parameter bChainParameter is used from PR533 to host in RDR_to_PC_DataBlock.

From host to PR533, the parameter wLevelParameters in PC_To_RDR_XfrBlock is used to request the next frame. This parameter is 2 bytes wide and is formatted in little endian (LSByte first).

Therefore the command to request next frame during chaining is:

6F 00 00 00 00 xx xx xx 10 00

(!! The byte order was different in version C350: the command was:

6F 00 00 00 00 xx xx xx 00 10)

14.3 Error byte at the end of Felica frame in transparent mode

In transparent mode, with Felica communication, the PR533 using the RxMultiple mode.

In this mode, when a frame is received, the frame is copied in the FIFO, and the error register value is copied at the end of the frame.

As we are in transparent mode, the full data is sent back to the host.

For this reason, in this mode (Felica, transparent), there is always an extra byte at the end of any received frame, corresponding to the Error Reg.

14.4 Multiple tags

In transparent mode, when several tags are used in the same command, the tags are processed in sequence, until the end, or until an error.

This means that when a tag is erroneous, the following tags will not be executed at all and will be ignored. The PR533 will answer with the error corresponding to the failed tag.

In case a tag must be executed, but is located after a failed tag, it is mandatory to send again the command, with the tags that have not been executed.

15. Known limitation and workaround

Some limitations have been discovered in latest release PR533/C360. The limitations and their Workaround are described here below:

15.1 MIFARE 4k – authentication after block 32

With MIFARE Classic 4k, the sectors are 4 blocks wide up to sector 32 (2k memory), then sectors are 16 blocks wide (last 2k memory).

The PR533 handles the whole memory as 4 blocks sectors. Then the application can only authenticate with 4 blocks at a time, even after sector 32.

The workaround is to authenticate with the 4 blocks memory part when accessing any part of the memory.

15.2 Dual Card JCOP + MIFARE

Dual card embedding JCOP and MIFARE functionalities cannot be accessed in MIFARE mode due to its SAK.
Indeed, in such a card, the SAK is merged with MIFARE information and ISO bit, and the PR533 uses this SAK value to allow read and write to a MIFARE Classic tag. If the SAK is not a MIFARE SAK, then the operation is aborted.
To bypass it, it is possible to force the expected SAK.
The SAK value is located at address 0x03B7.
Then right after the card activation, the SAK can be changed.

The following script shows how to perform this operation:
(read SAK first, then clear corresponding to ISO bit and write back, then read again to check the write went correctly):

```
#Clear fAutoRats
ScardControl:
Appli. => FF E1 03 00 01 20;
Reader <= 90 00;

# Switch off the RF field & the ATD mode;
ScardControl:
Appli. => FF E1 04 01 01 00;
Reader <= 90 00;

# InActivateDeactivateCard ISO14443A card;
ScardControl:
Appli. => FF E2 00 00 00;
Reader <= 90 00;

#Read SAK
ScardControl:
Appli. => FF E1 02 00 02 03 B7;
Reader <= 28 90 00;
```

```
#Write SAK
ScardControl:
Appli. => FF E1 02 01 03 03 B7 08;
Reader <= 90 00;

#Read SAK
ScardControl:
Appli. => FF E1 02 00 02 03 B7;
Reader <= 08 90 00;
```

15.3 MIFARE Crypto mode after deactivation

After a card session where MIFARE cryptography has been used, the deactivation does not automatically disable the Crypto mode of the PR533.
This mode is only deactivated by a new automatic connection.

As a consequence, a transparent session following an automatic MIFARE card session, will have MIFARE Crypto mode still on, and the first communication will then fail (PR533 will send MIFARE encrypted data instead of plain data).

The workaround to avoid this issue is to disable manually the MIFARE crypto mode after a MIFARE session, and before a transparent session. This mode can be disabled by clearing MFCrypto1On in CIU_Status2 register (@6338h). For instance, the command to write 00 in this register is:

FF E1 02 01 03 63 38 00;

15.4 Truncated received data in HSU transparent mode

In HSU mode, during a transparent session, the frame received from the card may be truncated if the number of bytes is too big.

The maximum data that can be received is 241 bytes from the card. If the size is larger, the last bytes are lost.

To avoid this, it is required to limit the number of bytes that the card can send. For instance:

- With ISO14443 cards, the size can be limited by sending an FSDI of 7 during RATS. Then the card will send frame with a max size of 128 bytes.
- In Felica mode, the max frame size must be limited to 241 bytes.

15.5 Abort command in HSU transparent mode

During a transparent session, in HSU mode, an abort command always resets the RF Field.

In this mode, when a card is activated, it is advised to always wait for the command to end by itself instead of aborting it; otherwise the card will be deactivated. All the commands have the capability to end by itself, either with a timeout, or with a common answer.

16. Error codes

Error codes are coded within the status word 1 and 2. This implementation respects the ISO7816-4 specification (see [ISO7816-4]).

The PCSC specification specifies the common error codes.

The table below lists these error codes:

Table 13. PCSC common error codes definition

SW1	SW2	Meaning
0x67	0x00	Wrong Length
0x68	0x00	Class byte is not correct
0x6A	0x81	Function not supported
0x6B	0x00	Wrong parameter P1-P2
0x6F	0xXX	Meaning according to SW2 (see below)

The error codes defined in the table above are valid for all commands defined within this section.

If the SW1 is set to 0x6F, SW2 codes error specified hereafter.

Table 14. Reader specific error codes

SW2	Meaning
0x01	Time Out, the target has not answered
0x02	A CRC error has been detected by the CIU
0x03	A Parity error has been detected by the CIU
0x04	During an anti-collision/select operation, an erroneous Bit Count has been detected
0x05	Framing error during MIFARE operation
0x06	An abnormal bit-collision has been detected during bit wise anti-collision at 106 kbps
0x07	Communication buffer size insufficient
0x08	Access has not been or may not be granted
0x09	RF Buffer overflow has been detected by the CIU
0x0A	RF error
0x0B	RF Protocol error
0x0D	Temperature error: the internal temperature sensor has detected overheating, and therefore has automatically switched off the antenna drivers
0x0E	Internal buffer overflow

0x0F	Baud rate not supported
0x10	Invalid parameter
0X12	Unsupported command
0x13	MIFARE or ISO/IEC14443-4: The data format does not match to the specification. Depending on the RF protocol used, it can be: • Bad length of RF received frame, • Incorrect value of PCB or PFB, • Invalid or unexpected RF received frame, • NAD or DID incoherence.
0x14	MIFARE: Authentication error
0x17	Frame OK, but block number mismatch
0x19	I2C bus line is Busy. A TDA transaction is on going
0x20	Problem on the I2C line
0x23	ISO/IEC14443-3: UID Check byte is wrong
0x25	The system/sub-system is in a state that does not allow the operation
0x26	Operation not allowed in this configuration (host controller interface)
0x27	This command is not acceptable due to the current context of the PR533
0x28	Current Command was aborted
0x29	The target has been released
0x2A	PR533 and ISO/IEC14443-3B only: the ID of the card does not match, meaning that the expected card has been exchanged with another one.
0x2B	PR533 and ISO/IEC14443-3B only: the card previously activated has disappeared.
0x2D	An over-current event has been detected
0x30	TDA is not FREE, it is busy by an old command and no response of this command is yet received
0x31	TDA is not waken up; TDA does not response; TDA is bad connected or is absent
0x32	TDA size of response is too long

All other error codes are given in the table below.

Table 15. Other error codes implemented in the PR533 FW

SW1	SW2	Meaning
0x69	0x81	Command incompatible
0x69	0x82	Security status not satisfied
0x69	0x83	Reader key not supported
0x69	0x84	Plain transmission not supported
0x69	0x85	Secured transmission not supported
0x69	0x86	Volatile memory not available, or Key type not known, or Command not allowed
0x69	0x87	Non-volatile memory not available
0x69	0x88	Key number not valid
0x69	0x89	Key length not correct
0x66	0x00	Invalid device state
0x6A	0x82	File not found
0x6C	0xXX	Wrong length, should be 0xXX
0x65	0x81	Memory failure, addressed by P1-P2 does not exist
0x63	0x00	No information given
0x62	0x82	End of file reached before Lc

Moreover command specific errors may be introduced as required in individual subsections.

17. Document management

17.1 Abbreviations and terminology

Table 16. Abbreviations and terminology

Abbreviation	Description
APDU	Application Protocol Data Unit
ATD	Automatic Tag Discovery
ATQA	Answer To Request, type A
ATQB	Answer To Request, type B
C-APDU	Command APDU
CID	Card Identifier
CIU	Contactless Interface Unit
CL	ContactLess
CPU	Central Processing Unit
CT	Cascade Tag
DEP	ISO/IEC18092 Data Exchange Protocol
DRI	Divisor Receive Integer (PCD to PICC)
DSI	Divisor Send Integer (PICC to PCD)
E.S.M	Energy Saving Mode
FSL	Maximum value for the Frame Length
HSU	High Speed UART (Universal Asynchronous Real Time)
I2C	Inter Integrated Circuit
IC	Integrated Circuit
ID	Card Identifier
N/A	Not Applicable
NAD	Node Address
N/I	Not Implemented
NU	Not Used

Abbreviation	Description
PCD	Proximity Coupling Device (Contactless PCD)
PPS	Protocol and Parameter Selection
R-APDU	Response APDU
RATS	Request for Answer To Select
RFU	Reserved for Future Use
SDD	Single Device Detection
TBD	To Be Defined
TLV	Encoding method (Type, Length, Value)
TSN	Time Slot Number
T=CL	ISO/IEC14443-4 protocol
UID	Unique Identifier, Type A
USB	Universal Serial Bus

17.2 Referenced documents

Table 17. Referenced documents

Doc ID	Doc Title	Version	Issue Date
[Datasheet]	PR533/C3 Product Datasheet	-	-
[ISO7816-4]			
[CCID]	Device Class: Smart Card – CCID	Rev. 1.1	Apr. 2005
[ISO14443-2]	ISO/IEC 14443-2		Oct. 2006
[ISO14443-3]	ISO/IEC 14443-3		Mar. 2006
[ISO14443-4]	ISO/IEC 14443-4	Second edition	Jul. 2008
[Bprime]	Innovatron Tag protocol	Rev. 3.2	Dec. 1999
[PICOPASS]	PicoPass datasheet	V1.6	Jul. 2005
[EMVco]	EMVco	V2.0.1	Jul. 2009
[PCSC1]	PCSC part 1 – Introduction	Rev 2.01.01	Sept. 2005
[PCSC2]	PCSC part 2 – Interface requirements	Rev 2.01.01	Sept. 2005
[PCSC3]	PCSC part 3 – Req. For PC-connected	Rev 2.01.09	Jun. 2007
[PCSC3-sup1]	PCSC part 3 – Supplemental	Rev 2.01.06	Jun. 2009
[PCSC3-sup2]	PCSC part 3 – Supplemental 2	Rev 2.02.00	Apr. 2010
[PCSC5]	PCSC part 5 – ICC resource manager	Rev 2.01.01	Sept. 2005
[PCSC6]	PCSC part 6 – ICC service provider	Rev 2.01.01	Sept. 2005
[PCSC8]	PCSC part 8 – Recommendations	Rev 2.01.01	Sept. 2005
[PCSC9]	PCSC part 9 – IFD with extended capabilities	Rev 2.01.01	Sept. 2005
[PCSC10]	PCSC part 10 – IFD with secure pin entry	Rev 2.02.08	Apr. 2010
[I²C]	I²C bus specification	Rev. 3.0	
[USB]	USB bus specification	Rev 2.0	
[AN10207-4]	Smart Card reader application with TDA8029 Mask06 & Mask07	Rev. 1.3	Feb. 2008

18. Legal information

18.1 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.

18.2 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.

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

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

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

18.3 Licenses

Purchase of NXP ICs with NFC technology

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

Purchase of NXP ICs with ISO/IEC 14443 type B functionality

This NXP Semiconductors IC is ISO/IEC 14443 Type B software enabled and is licensed under Innovatron's Contactless Card patents license for ISO/IEC 14443 B.

The license includes the right to use the IC in systems and/or end-user equipment.

RATP/Innovatron Technology

18.4 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.

I^2C-bus — logo is a trademark of NXP B.V.

19. List of figures

20. List of tables

21. Contents

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

Date of release: 11 January 2018
232013
Document identifier: UM10463