UM10915

PN7462AU PC CCID Reader User Manual

Rev. 1.0 — 9 March 2016
337010

User manual
COMPANY PUBLIC

Document information

Info	Content
Keywords	PN7462AU, NFC Reader, PC USB, PCSC
Abstract	This document briefs the setup environment required for PC CCID Reader use case demo on PN7462 Board

Revision history

Rev	Date	Description
1.0	20160309	First release

Contact information

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

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

1. Introduction

This document describe how to use the PN7462AU Customer Demo board as a CCID reader together with the CCID reader example.

The CCID reader example describe how to connect PN7462AU by USB interface to a PC and provide the CCID protocol implementation on the top of the physical link.

The CCID reader example is hosted on the PN7462AU and can be tested with any PC/SC application running on the PC with Windows OS.

This document also provides a description of the USB interface implementation, implementation of the CCID protocol and implementation of the PC/SC interface.

Two versions of the CCID reader example are available and explained here:

- phExCcid – this version is supporting contactless and contact interfaces,
- phExNFCCcid – this version is supporting contactless interface and P2P functionality.

1.1 Block diagram

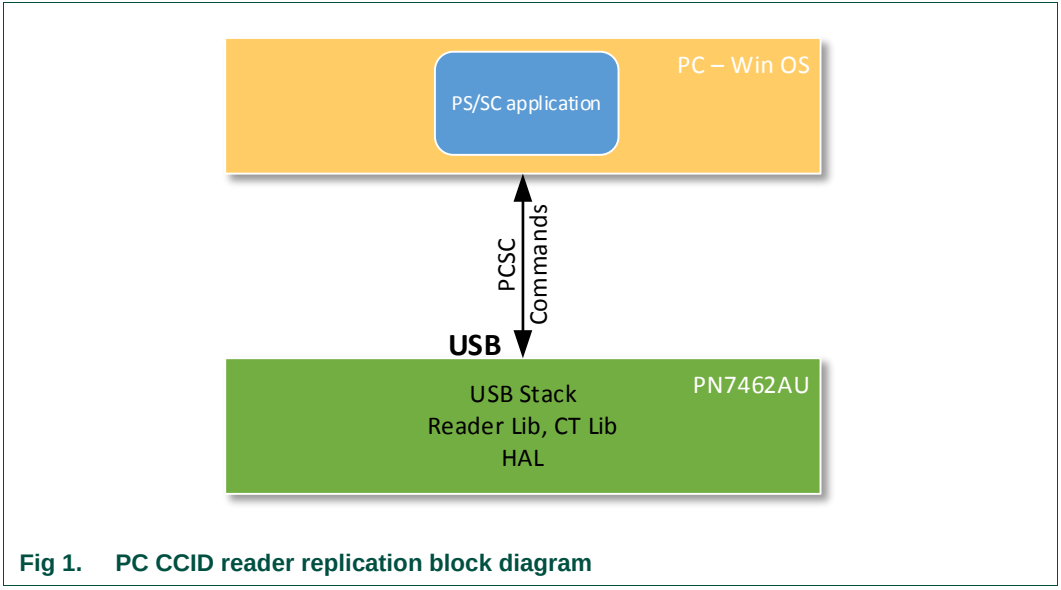

Fig 1. PC CCID reader replication block diagram

The USB Stack and CCID class is implemented in the PN7462AU. For the operation the default Windows CCID driver is used.

1.2 "phExCcid" example

This version of the CCID example demonstrates how to implement CCID functionality on the PN7462AU using contactless and contact interface. For a detailed description of the CCID protocol refer to the chapter 5.

1.2.1 Use case architecture

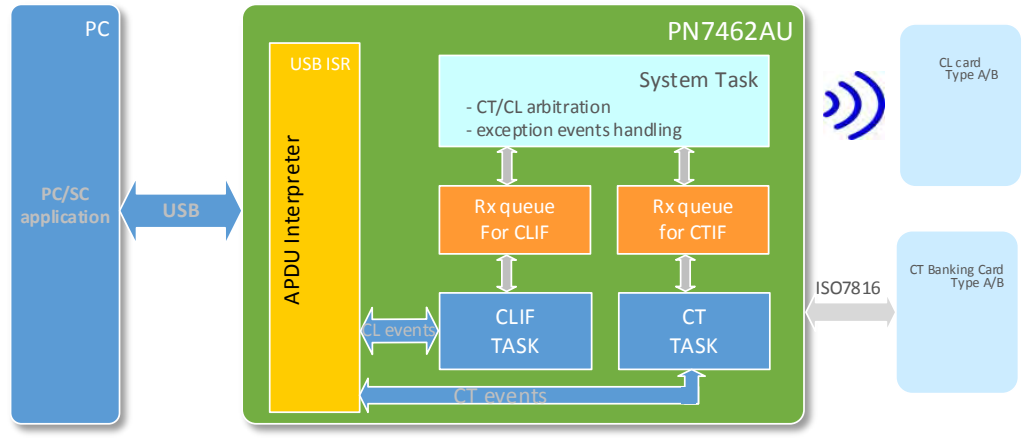

Fig 2. Application architecture of the "phExCcid" example

1.2.2 Modules overview

PN7462AU CCID reader application has following modules:

- USB ISR – send and receive the CCID class commands through the bulk out, bulk in and interrupt endpoints.
- System task
  - Responsible for any exceptions
  - Notification from CT/CL/Timer/PMU ISRs.
  - Responsible for initiating CT/CL task messages.
- CL task
  - Wait for messages from system task to start CL task for polling.
  - After polling wait for events from USB ISR for CCID commands.
- CT task
  - Wait for messages from system task to start CT task for card activation.
  - Wait for events from USB ISR for CCID commands after activation.

1.3 "phExNFCCcid" example

This version of the CCID example demonstrates how to implement CCID functionality on PN7462AU using contactless interface and P2P functionality. For a detailed description of the CCID protocol refer to the chapter 5.

This example also supports P2P functionality, sending a generated NDEF message via ISO18092 protocol to another NFC device. Type of the NDEF message is URL and contain "nxp.com" web address.

1.3.1 Use case architecture

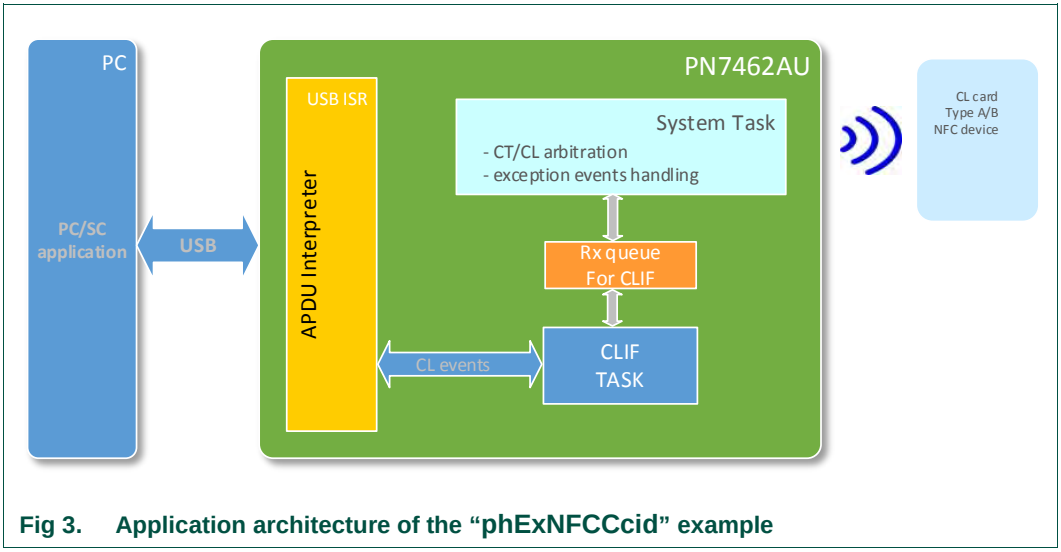

Fig 3. Application architecture of the "phExNFCCcid" example

1.3.2 Modules overview

PN7462AU CCID reader application has following modules:

- USB ISR – send and receive the CCID class commands through the bulk out, bulk in and interrupt endpoints.
- System task
  - Responsible for any exceptions
  - Notification from CL/Timer/PMU ISRs.
  - Responsible for initiating CL task messages.
- CL task
  - Wait for messages from system task to start CL task for polling.
  - After polling wait for events from USB ISR for CCID commands.
  - Polling for contactless cards and NFC devices
  - Support P2P functionality

2. Demo setup

This section describes the setup and execution environment required for CCID reader application.

To prepare the HW environment components listed in the table below are required.

Table 1. HW Components required in this example

Item	Version	Purpose
PN7462B	V2.1	Engineering development board.
LPC-Link2	1.0	Stand-alone debug adapter
USB Cable		USB To Micro
Power Adapter		NTS1000

2.1 Hardware setup

This chapter describes the hardware setup and the connection details to run this demo.

Fig 4 depicts the hardware setup to be used for the demo.

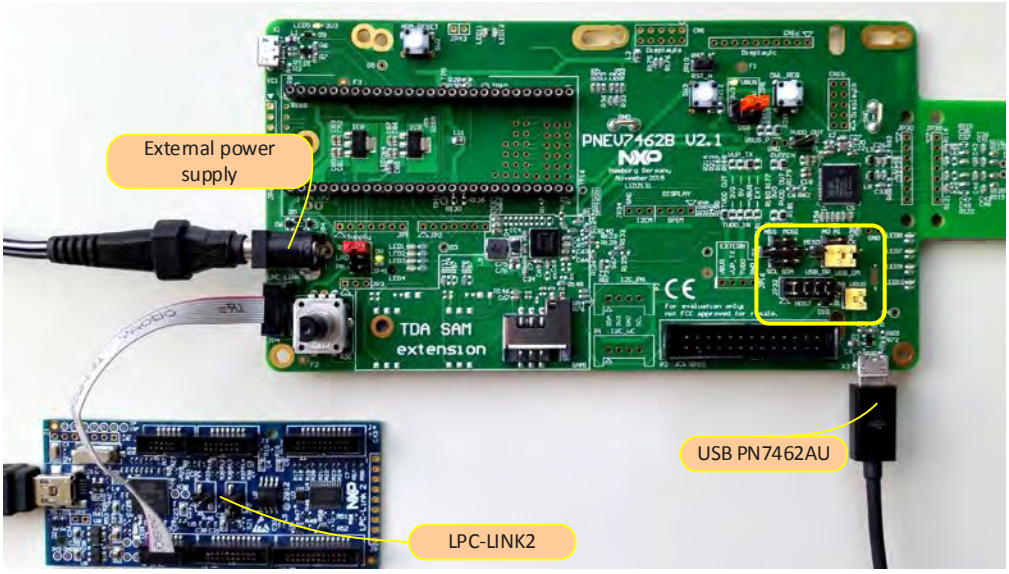

Fig 4. PC CCID reader hardware setup - PN7462B v2.1 board

Note:
External Power Adapter is not mandatory, the board can supplied by USB only - see figure below. Change the jumper JP41 to the PN to supply by USB.

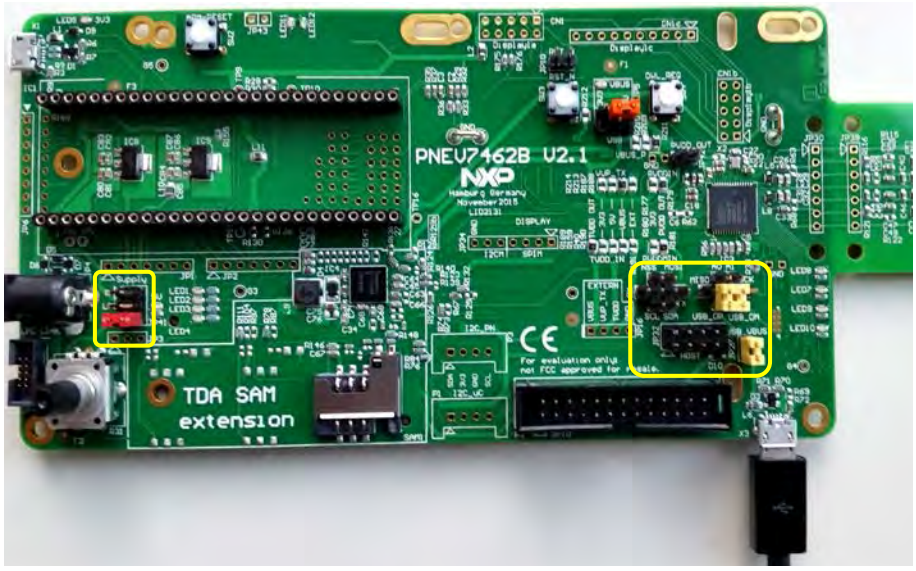

Fig 5. PC CCID reader hardware setup - PN7462B v2.1 board powered by USB

2.1.1 Jumpers settings

Before starting the application the following jumpers must be set.

Pre-Cautions:

Ensure that you are using the PN7462B v2.1 board and ensure that the basic jumpers for a proper functionality of the PN7462AU board are connected according to the settings below.

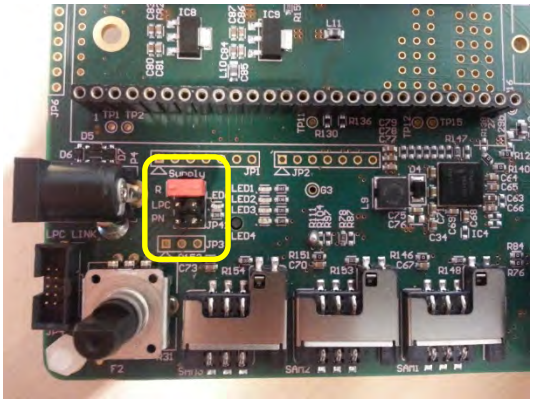

Fig 6. External Power Supply Settings

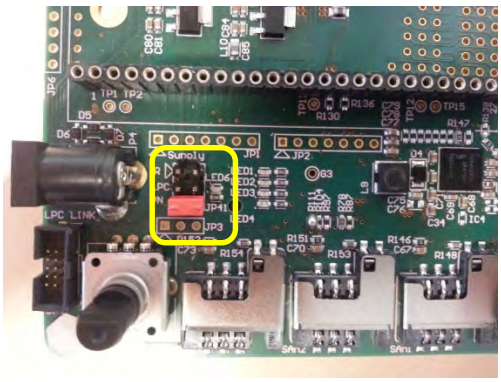

Fig 7. USB Power Supply Settings

Host Interface Selection – USB

Fig 8. Host Interface Selection - USB

Note:
No Interface selection is needed to execute the application.
USB host interface selection is required to establish the connection with the PC.

3. PC CCID reader application source

This section describes how to:

1. Download and import the PC CCID Reader application source code.
2. Build the environment
3. Compilation and loading.

3.1 Download source code

The source code of the PC CCID Reader application is part of a delivered “PN7462AU customer support package”. After installation of the support package, project file with source files is located in *“.\PN7462AU Software”* folder.

3.2 Development environment

To prepare the project and build the source code, components listed in the Table 2 are required.

Table 2. Development environment

Item	Version	Purpose
PN7462B	2.1	Engineering development board.
LPC-Link2	1.0	Standalone debug adapter
LPCXpresso IDE	8.0.0 and above	Development IDE
LPCXpresso PN7462AU Plugin	8.0.0	Add PN7462AU reader to the LPCXpresso

Note:
Installation procedure of the *“LPCXpresso IDE” and “LPCXpresso PN7462AU Plugin”* is described in the *“UM10883 - PN7462AU Quick Start Guide - Customer Board”* document. Latest LPCXPresso plugin version is required.

3.3 Importing project

After installation of the “PN7462AU Customer Support Package” please follow steps described below.

The sequence of preparing the project is:

- Open the LPCXpresso IDE and select an empty workspace
- Select the option “Import project(s)” in the Quickstart Panel

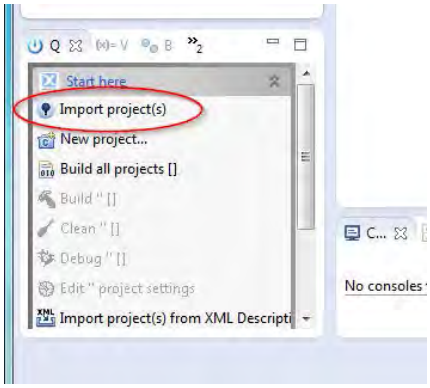

Fig 9. Importing project to LPCXpresso IDE

The dialog for project import opens.

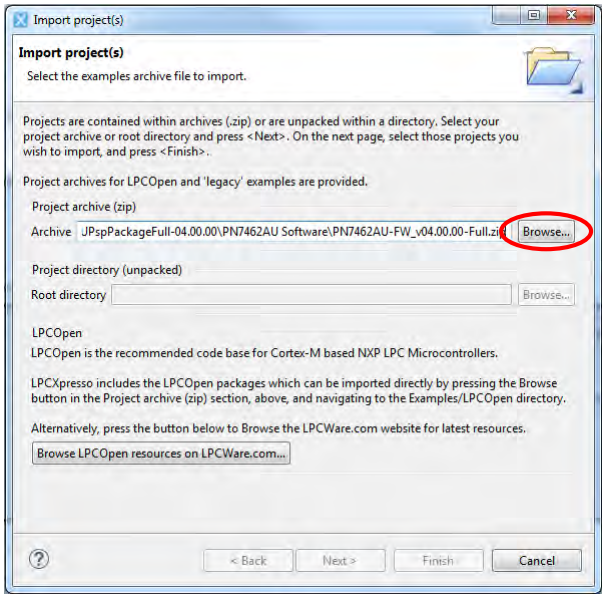

Fig 10. Importing project to LPCXpresso IDE

Browse to the project zip file *".\PN7462AU Software\PN7462AU-FW_v04.00.00-Full.zip"* and click *"Next"*.

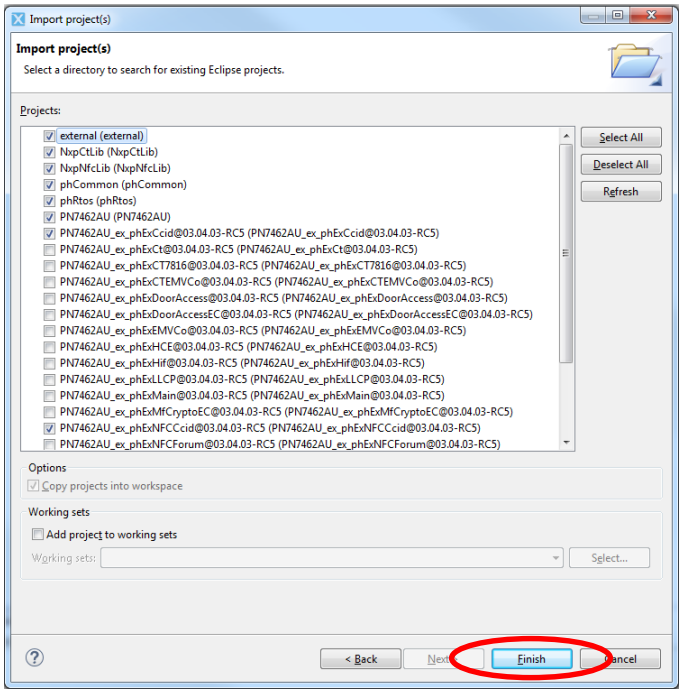

Fig 11. Select project

In the projects window all available projects in the package will be listed. To import only CCID examples, it is mandatory to select next projects in the list:

- external
- NxpCtLib
- NxpNfcLib
- phCommon
- phRtos
- PN7462AU_ex_phExCcid
- PN7462AU_ex_phExNFCCcid

All projects in the list can be also selected and imported to the workspace.

Select appropriate projects and click Finish. Selected applications are going to be imported to the workspace.

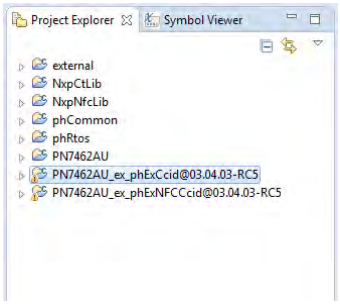

Fig 12. Project Workspace with both CCID projects

3.4 Building project

Building projects in a workspace is a simple case of using the “Quickstart Panel” - ‘Build all projects’. Alternatively a single project can be selected in the “Project Explorer View” and built separately. Note that building a single project may also trigger a build of any associated library projects.

To build the project select appropriate project and press “Build” as shown in the figure below.

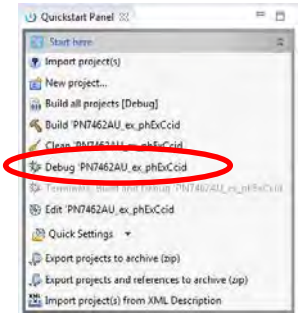

Fig 13. Build project

After successful project build, there should be no errors as shown on the picture below.

Fig 14. Successful build

As a part of the build output, the binary file for Flash is created. This binary file can be used to update PN7462AU Flash via USB mass storage interface or by using Flash tool or debug in LPCXpresso IDE. In case that “Binaries” folder is not visible in the project structure, refresh the project (right click on project and select “Refresh”).

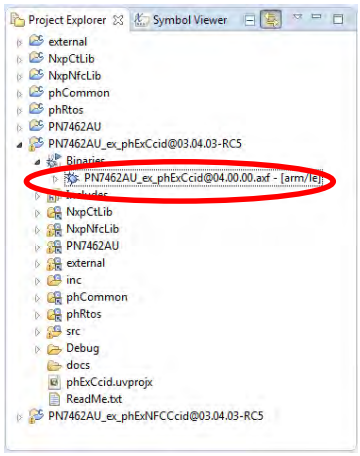

Fig 15. Build output - flash binary file

3.5 Flashing

This section briefs the steps how to flash generated binary (PN7462AU_ex_phExCcid.axf).

1) Select “*PN7462AU_ex_phExCcid.afx*” in “Project Explorer View”
2) Click on the “Program Flash Option”.
3) Ensure that all the options are set properly and Click OK.

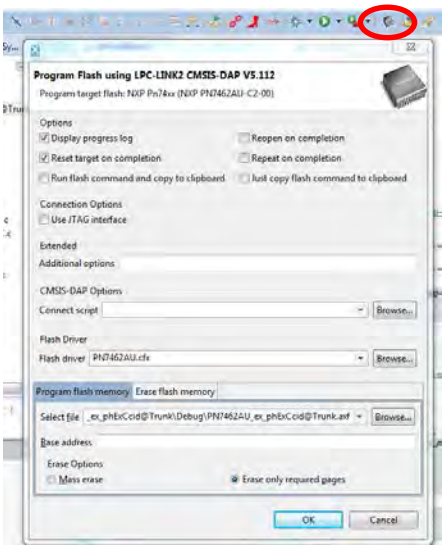

Fig 16. Flash binary file

3.6 Release mode setup

In the Release Mode, after flashing the “PN7462AU_ex_*phExCcid.axf*” file, the board can be connected to the PC and communicating with the PC applications without the LPC Link2 debugger.

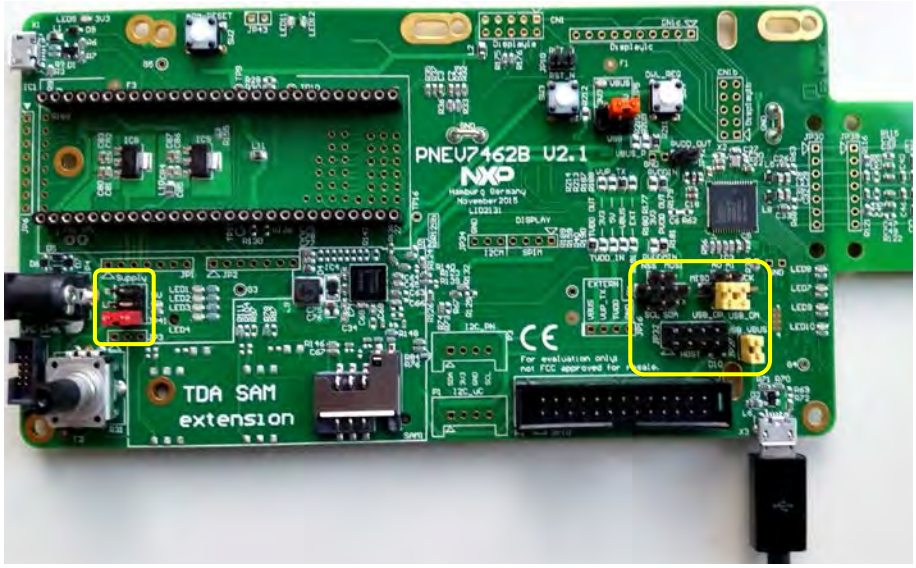

Fig 17. PN7462AU Customer Demo board – release mode

3.7 Debug mode setup

Debug mode is used in case of debugging the code or tracing the process execution.

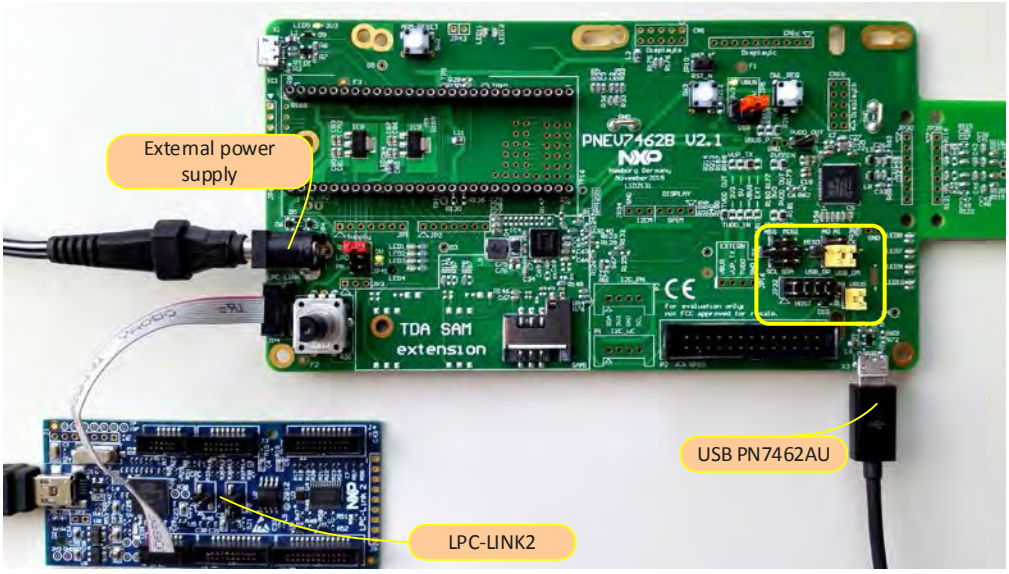

Fig 18. PN7462AU Customer Demo board – debug mode

Application can be flashed and debugged using LPC-LINK2.

To debug the application on the PN7462AU, simply highlight the project in the Project Explorer and click Debug in the Quick start Panel, as shown in Fig 19. The LPCXpresso IDE will first build application and start with debugging.

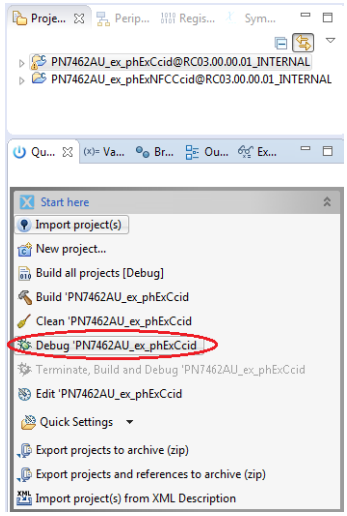

Fig 19. Debug the project

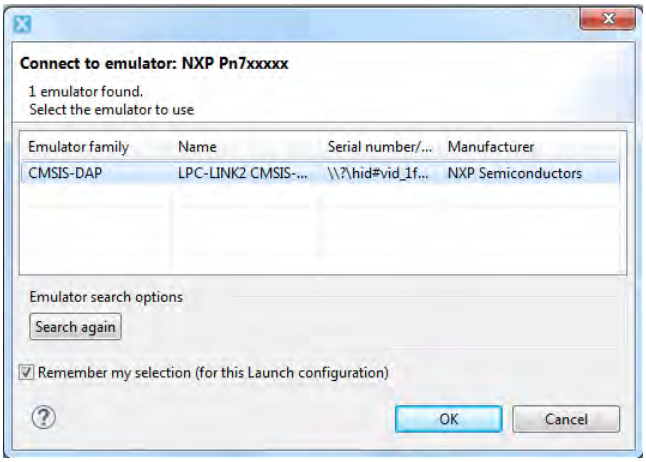

Fig 20. Select Debug Emulator – LPC-LINK2

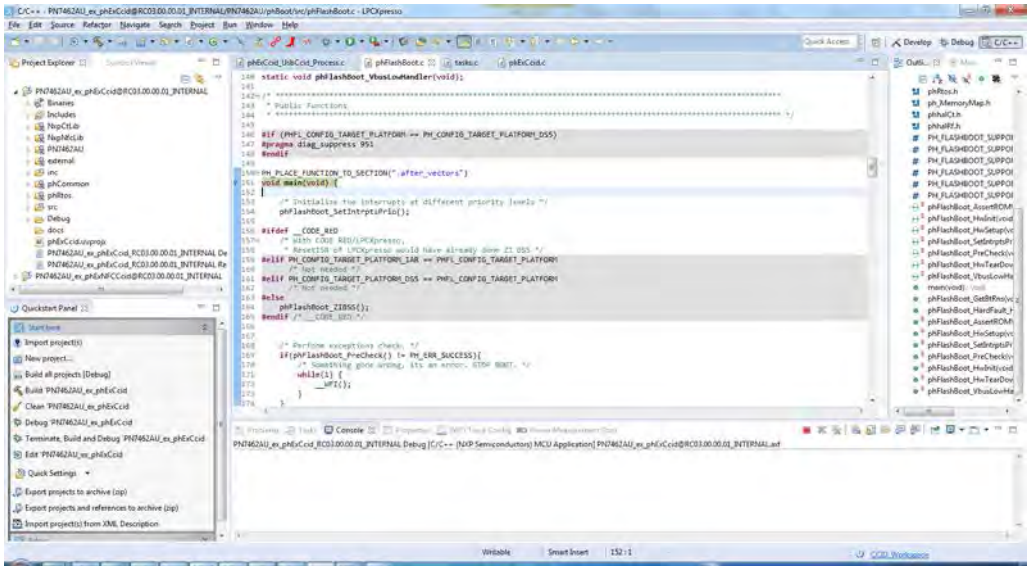

Fig 21. Debug project

The application execution is stopped at the beginning, to continue execution must be resumed, click resume button or press F8 key.

3.8 Features Support

3.8.1 Suspend Resume and Remote Wakeup Feature

In order to enable the Suspend Resume and Remote Wakeup features, following macros in “APP_NxpBuild.h” file must be set. By default both features are disabled.

```
PH_EXCCID_USB_IF_USB_SUSPEND_RESUME_FTR=1
PH_EXCCID_USB_IF_USB_REMOTE_WAKEUP_FTR=1
```

[Note: PH_EXCCID_USB_IF_USB_REMOTE_WAKEUP_FTR should be enabled only if Suspend Resume feature is enabled]

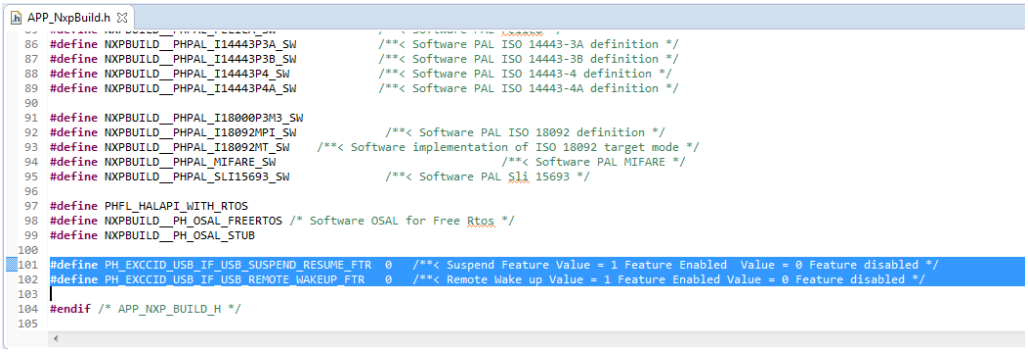

Fig 22. Project Properties – USB Suspend Mode

3.8.2 Enabling the Suspend Feature on PC

When the Remote Wakeup feature is enabled the PC has the capability to put the device in Suspend mode when there is no transaction on the USB bus for some 3ms duration. To use this feature enable the following option in the Device Manager Section of the PC under Microsoft Usbccid Smartcard Reader.

The options to enable the suspend resume feature and remote wakeup feature are available in the Power Management Tab Section.

Enabling the Option 1 as shown in the figure below will enable the Suspend Feature.

By Default Option 2 as shown in the figure below will be enabled for the remote wake up feature to function.

Once the Device enters into suspend using the CT Insertion/Removal the Resume operation can be performed.

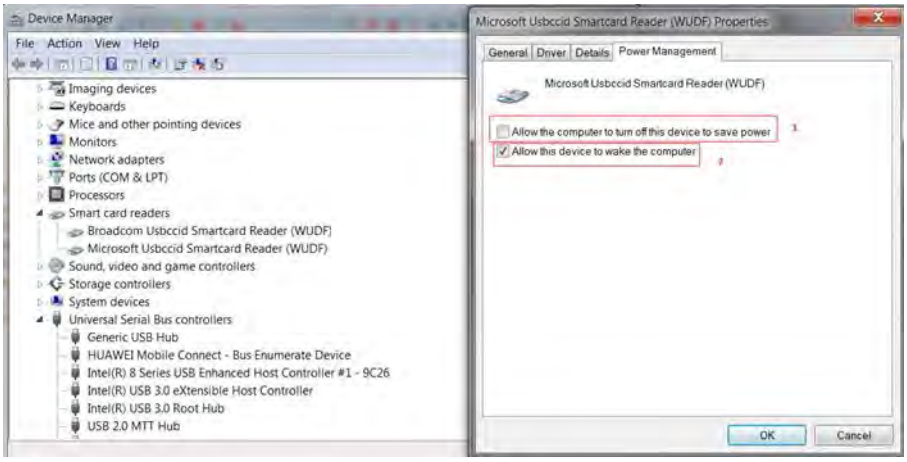

Fig 23. Usbccid Smartcard Reader Properties

3.8.3 Enabling the Escape Command Feature in Registry:

In order perform the Escape Command of the CCID reader directly with the device following has to be done in the Registry.

Under the WUDFUsbccidDriver folder enter the following parameter "EscapeCommandEnable" and set it with the REG_DWORD as 0x1.

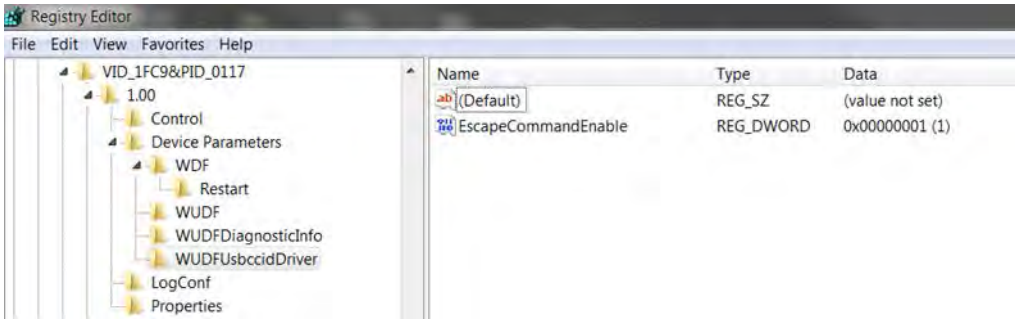

Fig 24. Registry Editor – EscapeCommandEnable

4. Host Interface - USB Link

4.1 Introduction

The System host controller communicates with the PN7462AU by using the USB link. The protocol between the host controller and the PN7462AU on top of this physical link is the CCID protocol.

4.2 USB Interface Model

In this example PN7462AU uses 3 endpoints in addition to two mandatory default endpoints to control IN/OUT.

Control commands are sent on control endpoints. These include class-specific requests and USB Standard requests. Commands that are sent on the default endpoint report information back to the host on the default control endpoint.

CCID Events like card detection or removal are sent on the Interrupt Endpoint.

CCID Commands are sent on BULK-OUT Endpoint. Each command sent to PN7462AU has an associated ending response.

CCID Responses are sent on BULK-IN Endpoint. All commands sent to PN7462AU have to be sent synchronously.

4.2.1 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 Descriptors for the PN7462AU PC CCID Reference reader are listed below:

Table 3. 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	40h	--
8	idVendor	2	1FC9h	--
10	idProduct	2	0117h	--
12	bcdDevice	2	0101h	--
14	iManufacturer	1	01h	--
15	iProduct	1	02h	--
16	iSerialNumber	1	03h	--
17	bNumConfigurations	1	01h	--

Table 4. 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		--
4	bNumInterfaces	1	01h	--
5	bConfigurationValue	1	01h	--
6	iConfiguration	1	00h	--
7	bmAttributes	1	80h	--
8	bMaxPower	1	7Dh	--

Table 5. 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	000037F0h	--
18	bNumClockSupported	1	00h	--
19	dwDataRate	4	000026B5h	--
23	dwMaxDataRate	4	CF080h	--
27	bNumDataRatesSupported	1	00h	--
28	dwMaxIFSD	4	000000FEh	254 bytes
32	dwSynchProtocols	4	00000000h	--

Table 6. Interface Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	36h	Size of this descriptor
1	bDescriptorType	1	21h	**Functional descriptor**

Offset	Field	Size	Value	Description
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	000037F0h	--
18	bNumClockSupported	1	00h	--
19	dwDataRate	4	000026B5h	--
23	dwMaxDataRate	4	CF080h	--

Table 7. Endpoint Descriptor BULK IN

Offset	Field	Size	Value	Description
0	bLength	1	07h	7 bytes of descriptor length
1	bDescriptorType	1	05h	**Endpoint descriptor**
2	bEndpointAddress	1	82h	Physical Ept #2, type IN
3	bmAttributes	1	02h	BULK endpoint
4	wMaxPacketSize	2	0040h	64 bytes of max. packet size
6	bInterval	1	00h	

Table 8. Endpoint Descriptor BULK OUT

Offset	Field	Size	Value	Description
0	bLength	1	07h	7 bytes of descriptor length
1	bDescriptorType	1	05h	**Endpoint descriptor**
2	bEndpointAddress	1	02h	Physical Ept #2, type OUT
3	bmAttributes	1	02h	BULK endpoint
4	wMaxPacketSize	2	0040h	64 bytes of max. packet size
6	bInterval	1	00h	-

Table 9. Endpoint 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	0040h	64 bytes of max. packet size
6	bInterval	1	04h	

4.2.2 Frames Structure

Communication between the host controller and the PN7462AU is performed through frames which respects the CCID Specification.

CCID Frame	Payload

4.2.3 Dialog Structure

The following sections explain the dialog structure about the physical link used.

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

- It sends a command to the PN7462AU
- The PN7462AU executes the command.
- PN7462AU sends back the corresponding answer to the host controller.

4.2.3.1 Data Link Level

Successful exchange at data link level

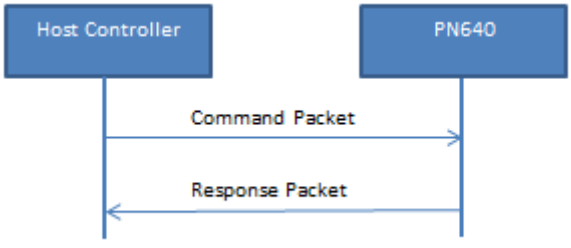

Fig 25. Data Link Level

4.2.3.2 Application Level

a) Successive Exchanges

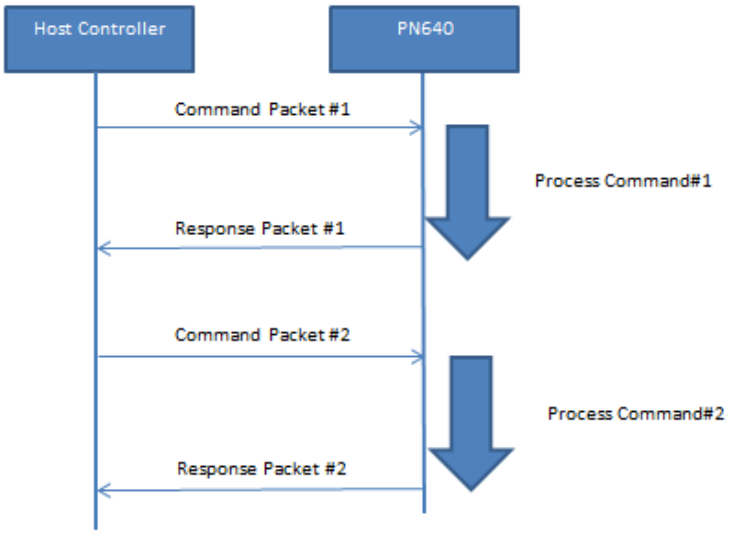

Fig 26. Application Level

b) Abort

The host controller can force the PN7462AU to abort an ongoing process. If the PN7462AU receives a command before having answered to the previous one, it will only send response to the previous command.

c) Error at application level

When the PN7462AU 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.

5. CCID Protocol

PN7462AU doesn't support all features of CCID Specification. The below table depicts the list of CCID commands. The column "Supports" informs if the command is supported or not on the PN7462AU. If the command is not supported the status in the response is set to command not supported.

Table 10. CCID Command set

Command	bMessage Type	Supports	Reference
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	Yes	[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	No	[CCID] 6.1.13
PC_to_RDR_SetDataRateAndClockFrequency	0x73	No	[CCID] 6.1.14

6. PC/SC API

PC/SC interface is a specification for contact and contactless smartcard into the computer eco-system. When the PN7462AU is plugged with the USB link, the device is controlled using this interface on top of the USB driver. PN7462AU is compliant with the CCID transport protocol. It does not need a native driver. The device uses the driver from the operating system.

This Section will describe some of the PCSC API for application programming usage. For detail, please refer to Microsoft MSDN Library or PCSC Workgroup.

6.1 SCardEstablishContext

The **SCardEstablishContext** function establishes the resource manager context within which database operations are performed.

Refer: http://nxp.com/SCARDESTABLISHCONTEXT

6.2 SCardListReaders

The **SCardListReaders** function provides the list of readers within a set of named reader groups, eliminating duplicates.

The caller supplies a list of reader groups, and receives the list of readers within the named groups. Unrecognized group names are ignored. This function only returns readers within the named groups that are currently attached to the system and available for use.

Refer: http://nxp.com/SCARDLISTREADERS

6.3 SCardConnect

The **SCardConnect** function establishes a connection (using a specific resource manager context) between the calling application and a smart card contained by a specific reader. If no card exists in the specified reader, an error is returned.

Refer: http://nxp.com/SCARDCONNECT

6.4 SCardControl

The **SCardControl** function gives you direct control of the reader. You can call it any time after a successful call to SCardConnect and before a successful call to SCardDisconnect. The effect on the state of the reader depends on the control code.

Refer: http://nxp.com/SCARDCONTROL

6.5 SCardTransmit

The **SCardTransmit** function sends a service request to the smart card and expects to receive data back from the card.

Refer: http://nxp.com/SCARDTRANSMIT

6.6 SCardDisconnect

The **SCardDisconnect** function terminates a connection previously opened between the calling application and a smart card in the target reader.

Refer: http://nxp.com/SCARDDISCONNECT

6.7 APDU Flow

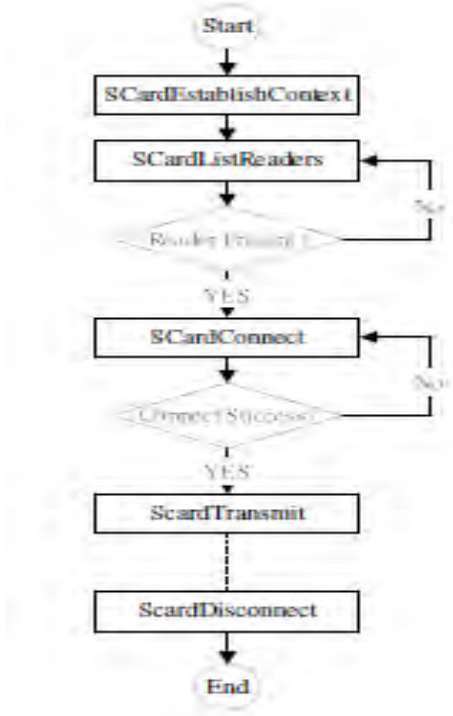

Fig 27. APDU Flow

6.8 Escape Command Flow

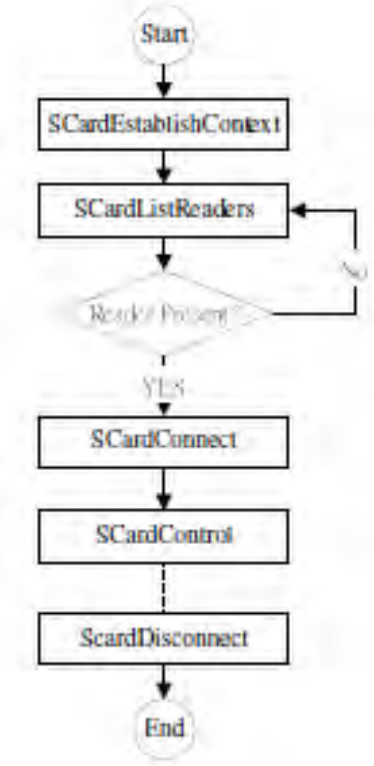

Fig 28. Escape Command Flow

6.9 Contactless Smart Card Protocol

6.9.1 ATR Generation

If the Reader detects a PICC, an ATR will be sent to the PCSC driver for identifying the PICC.

6.9.1.1 ATR format – ISO 14443 Part 3 PICCs

Table 11. Table title here

Table description (optional)

Byte	Value (Hex)	Designation	Description
0	3B	Initial header	
1	3N	T0	Higher nibble 8 means: no TA1, TB1 and TC1, only TD1 is following. Low nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)
2	80	TD1	Higher nibble 8 means: no TA2, TB2 and TC2, only TD2 is following. Low nibble 0 means T = 0
3	01	TD2	Higher nibble 8 means: no TA2, TB2, TC2 and TD3. Low nibble 0 means T = 0
4	80	T1	Category indicator byte, 80 means A status Indicator may be present in an optional COMPACT-TLV data object
To	4F	Tk	Application identifier – Presence indicator
	0C		Length
	RID		Registered Application Provider identifier (RID) #A0 00 00 03 06
3 + N	SS		Byte for standard
	C0..C1		Byte for card name
	00 00 00 00	RFU	RFU #00 00 00 00
4 + N	UU	TCK	Exclusive – oring of all the bytes T0 to TK

e.g.
ATR for MIFARE 1K = { 3B 8F 80 01 80 4F 0C A0 00 00 03 06 03 00 01 00 00 00 00 6A}
Length (YY) = 0x0C
RID = {A0 00 00 03 06} (PC/SC Workgroup)
Standard (SS) = 0x03 (ISO 14443A, Part 3)
Card Name (C0..C1) = {00 01) (MIFARE 1K)

Table 12. Standard (SS)

Code	
03	ISO 14443A, Part 3
11	Felica

Card Name (C0..C1)

Table 13. Card Name

Code	
00 01	MIFARE 1K

Code	
00 02	MIFARE 4K
00 03	MIFARE Ultralight
00 26	MIFARE Mini
00 30	Topaz and Jewel
00 3B	Felica
FF 28	JCOP 30
FF [SAK]	undefined tags

6.9.1.2 ATR format – ISO 14443 Part 4 PICCs

<table>
<tr><th>Byte</th><th>Value (Hex)</th><th>Designation</th><th>Description</th></tr>
<tr><td>0</td><td>3B</td><td>Initial Header</td><td></td></tr>
<tr><td>1</td><td>8N</td><td>T0</td><td>Higher nibble 8 means: no TA1, TB1, TC1 only TD1 is following.
Lower nibble N is the number of historical bytes (HistByte 0 to HistByte N-1)</td></tr>
<tr><td>2</td><td>80</td><td>TD1</td><td>Higher nibble 8 means: no TA2, TB2, TC2 only TD2 is following.
Lower nibble 0 means T = 0</td></tr>
<tr><td>3</td><td>01</td><td>TD2</td><td>Higher nibble 0 means no TA3, TB3, TC3, TD3 following.
Lower nibble 1 means T = 1</td></tr>
<tr><td rowspan="2">4
to
3 + N</td><td>XX</td><td>T1</td><td rowspan="2">Historical Bytes:

ISO14443A:
The historical bytes from ATS response. Refer to the ISO14443-4 specification.

ISO14443B:</td></tr>
<tr><td>XX
XX
XX</td><td>Tk</td></tr>
<tr><td></td><td></td><td></td><td><table><tr><th>Byte1-4</th><th>Byte5-7</th><th>Byte8</th></tr><tr><td>Application Data from ATQB</td><td>Protocol Info Byte from ATQB</td><td>Higher nibble=MBLI from ATTRIB command
Lower nibble (RFU)=0</td></tr></table></td></tr>
<tr><td>4+N</td><td>UU</td><td>TCK</td><td>Exclusive-oring of all the bytes T0 to Tk</td></tr>
</table>

Fig 29. ATR format – ISO 14443 Part 4

E.g. 1.

ATR for DESFire = {3B 81 80 01 80 80 } // 6 bytes of ATR

Hint: Use the APDU “FF CA 01 00 00” to distinguish the ISO14443A-4 and ISO14443B-4 PICCs, and retrieve the full ATS if available. ISO14443A-3 or ISO14443B-3/4 PICCs do have ATS returned.

APDU Command = FF CA 01 00 00

APDU Response = 06 75 77 81 02 80 90 00

ATS = {06 75 77 81 02 80}

E.g 2.

ATR for ez-link = {3B 88 80 01 1C 2D 94 11 F7 71 85 00 BE}

Application Data of ATQB = 1C 2D 94 11

Protocol Information of ATQB = F7 71 85

MBLI of ATTRIB = 00

7. APDU

PN7462AU receives an ISO7816-4 compliant APDU within a CCID frame. With respect to PC/SC specification the reader interprets this APDU.

Command APDU

CLA	INS	P1	P2	Lc	Data	Le

Response APDU

Data	SW1	SW2

7.1 APDU Supported

Following sections will give the description of each command.

This description contains:

- ⇨ The Frame structure, including the type and the amount of data:
  - o That the host application has to deliver to the PN7462AU. (**INPUT**).
  - o That the PN7462AU returns to the host application. (**OUTPUT**).
  - o When existing the possible error causes. (**Error Status Word**)

Table 14. Command Set

PCSC Standard commands	INS
Load Key	0x82
General Authenticate	0x86

PCSC Standard commands	INS
Get Data	0xCA
Read Binary	0xB0
Update Binary	0xD6
Sample NXP proprietary command	**INS**
Get FW Version	0xE1
User Defined APDU Command	CLA
LED Control	0xA0

7.1.1 Load Key

The Load Key Command will load MIFARE keys in the PN7462AU. These keys are used by the General Authenticate Command.

Note: This command will load data in a volatile memory. So if the PN7462AU is turned OFF, the key will be lost. Moreover if the MIFARE card is deactivated (or lost) the 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:**

0x00 – Key is loaded into the reader volatile memory. Others – Reserved.

⇨ **Key Number:**

0x00 – Only 1 key can be stored by the PN7462AU. Other values are RFU.

⇨ **Key**

The key value of 6 bytes loaded into the reader.

Output:

Response	
SW1	SW2

⇨ **Results:**
90 00 **– Success**

Example:

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

⇨ **R-APDU:** `90 00`

7.1.2 General Authenticate

The General Authenticate command will perform the Authenticate Sequence on a MIFARE card.

This command is applicable on the following cards: MIFARE 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**
Type of Key to be used:
- o 0x60 – MIFARE KEY_A
- o 0x61 – MIFARE KEY_B

⇨ **Key Number**
The Key number to be used for the authentication. It corresponds to the key number set with the Load Key command.

Output:

Response	
SW1	SW2

⇨ **Results:**
90 00 – **Success**

Example:

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

This command performs an authentication on block 4 using the MIFARE Key_A and the key number 0.

⇨ **R-APDU:** `90 00`

Get Data

Get Data command will retrieve the card information.

Input:

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

The behavior of the PN7462AU 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; •
0x01	Only for ISO14443-4 cards: • Type A: Historical bytes from ATS; • Type B INF field of ATTRIB response;

Output:

Response		
[Data Out]	SW1	SW2

⇨ **Data Out:** Response of the Card
⇨ **SW1 SW2:** Status

Example

⇨ **Get the UID of MIFARE DESFire**

```
C-APDU: FF CA 00 00 00
R-APDU: 04 4F 22 21 70 1C 80 90 00
```

7.1.3 Read Binary

The Read Binary command will read data from a contactless card. The card has to be activated before. This command is applicable only for contactless storage cards like MIFARE.

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 card activated.

MIFARE Family

Command	CLA	INS	P1	P2	Le
UL READ 16	0xFF	0xB0	0x00	0x00 to 0x30	0x10
Classic 1k READ 16	0xFF	0xB0	0x00	0x00 to 0x40	0x10
Classic 4k READ 16	0xFF	0xB0	0x00	0x00 to 0x40	0x10

Output:

Response		
[Data Out]	SW1	SW2

- ⇨ **Data Out:** Response of the Card
- ⇨ **SW1 SW2:** Status

Example

- ⇨ **MIFARE Read 16 bytes from block 04**

```
C-APDU: FF B0 00 04 10
R-APDU: 94 D5 B0 46 6B 2A 4F B9 94 D5 B0 46 93 6C 93 6C 90
00
```

7.1.4 Update Binary

The update Binary command will try to write data in the activated card.

The command is applicable only for contactless storage cards.

Input:

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

The behavior of PN7462AU 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 0x30	0x04	Data
Classic 1k WRITE 16	0xFF	0xD6	0x00	0x00 to 0x40	0x10	Data
Classic 4k WRITE 16	0xFF	0xD6	0x00	0x00 to 0x40	0x10	Data

Output

Response	
SW1	SW2

⇨ **SW1 SW2:** Status

Example

⇨ **MIFARE UL Write 4 bytes in block 06**

```
C-APDU: FF D6 00 06 04 01 02 03 04
R-APDU: 90 00
```

7.1.5 Get FW Version

This command is custom/user-defined APDU command to get the FW Version.

Input

Command	Class	INS	P1	P2	Le
Get FW Version	0xFF	0xE1	0x00	0x00	0x1C

Output

Response		
[Data Out]	SW1	SW2

- ⇨ **Data Out:** The FW version of the CCID device is displayed.
- ⇨ **SW1 SW2:** Status

Example

```
C-APDU: FF E1 00 00 1C
R-APDU: 36 34 30 20 46 57 20 30 30 20 30 34 20 30 30 20 55
43 20
        30 30 20 30 31 20 30 30 20 90 00
```

8. Using example

The following section provides some example commands for testing the CCID reader example using the Terminal Simulator and also PCSC diagnostic tool.

8.1 Pay pass terminal simulator

- Open the Terminal Simulator application.
- Press F2 and select the NXP PN7462AU CCID 0.
- Press OK

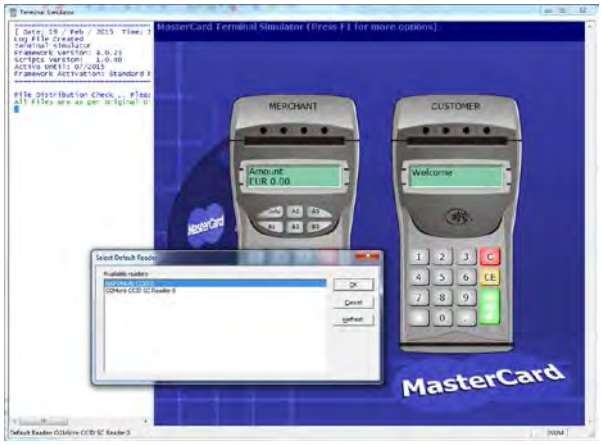

Fig 30. Select PN7462AU as CCID Reader

8.1.1 Banking card check

In the merchant terminal perform the following steps:

1) Press any amount from the number keypad.
2) Press the GREEN button.

In the screen following message appears:

```
================================================================================
[ Date: 12 / Dec / 2014  Time: 12:16:42 ]
Start transaction
User entered amount = 3.00 EUR
Transaction Currency Code = 978


================================================================================

================================================================================
Connected Reader: NXP PN640 CCID 0
Waiting for card...
================================================================================
```

Place the banking card on the antenna for CL check or insert the card in to the CT slot for CT check.

The status of transactions will be appearing in the screen of the application.

Note:

The JCOP Card used should contain the proper application for successful transaction.

8.2 SCRTester tool

To check the CCID reader example SCRTester tool can be used. This tool is part of the CCID delivery package and can be found in "*Software/SCRTester*" folder. Application is also available on the nxp.com website.

http://nxp.com/documents/software/141410.zip

File: SCRTester1641_setup.rar

Application: SCRTester.exe

Version: 1.6.4.1

Application must be installed to before proceeding.

8.2.1 Using application

Before start, customer demo board needs to be connected as described in chapter 2 and CCID example must be running. How to flash and run the example is described in chapter 3.

To start using SCRTester tool please follow next steps:

- Connect the PN7462AU Customer demo board to the PC with the USB cable.
- Open the SCRTester tool

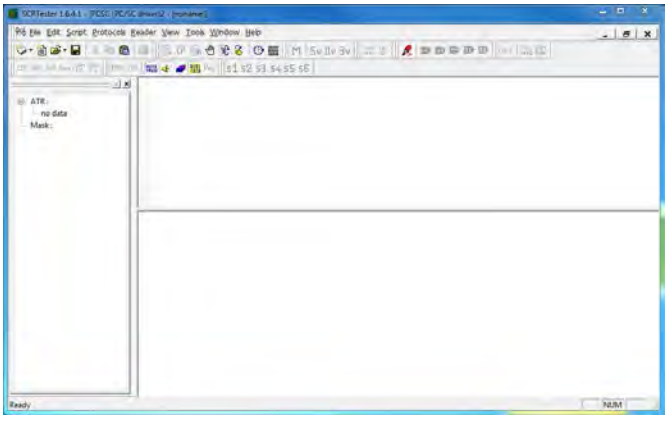

Fig 31. SCRTester tool

Select PCSC driver

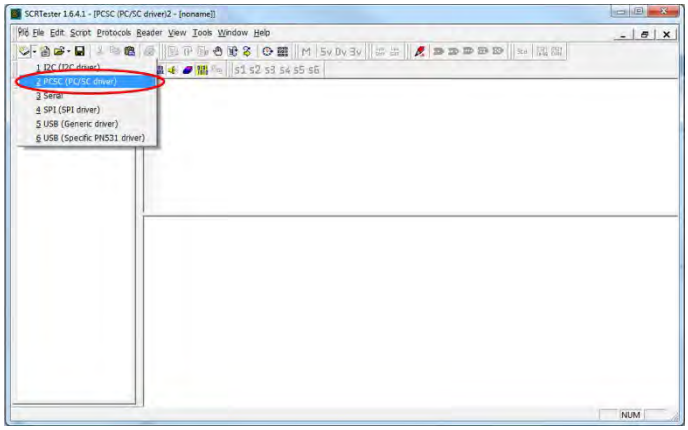

Fig 32. PCSC driver

Select the PN7462AU as a PCSC reader.

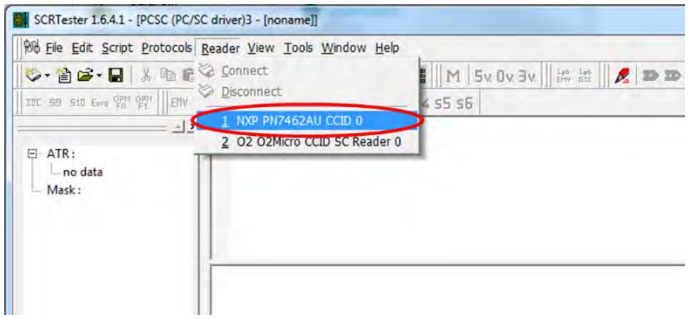

Fig 33. Select reader

Next step is to select PCSC protocol

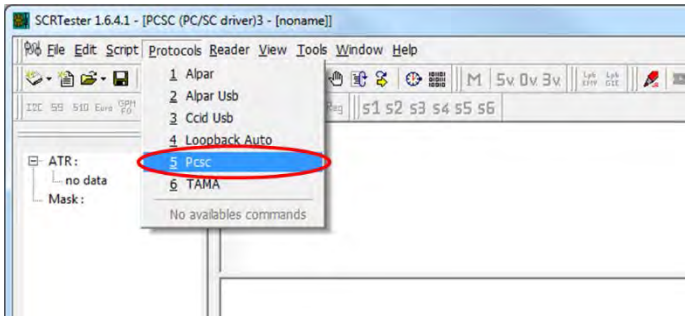

Fig 34. Select PCSC Protocol

Place the card on the antenna and connect application with the reader.

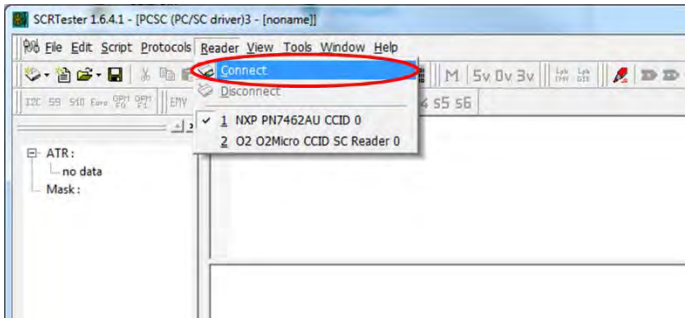

Fig 35. Connect

The ATR will appear in the card ATR section as shown below.

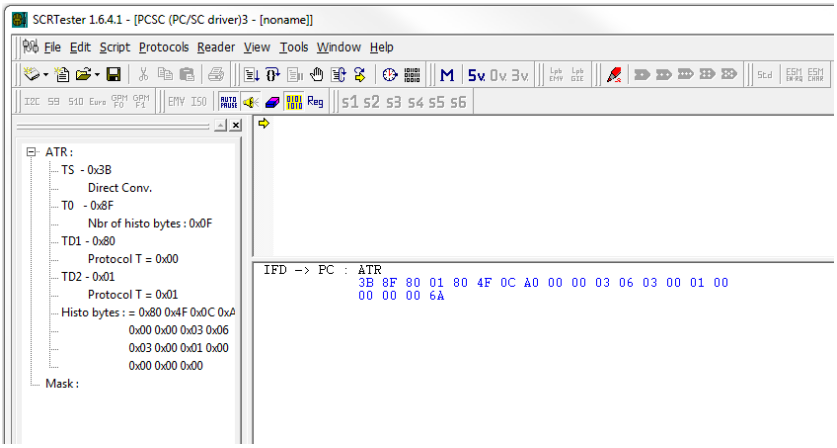

Fig 36. ATR appear

8.2.2 MIFARE Classic example commands

- Place the MIFARE Classic card on the antenna and connect application with the reader, the card gets detected and the ATR appears.
- After the above step enter the following commands one by one in the command APDU area and check the response in the below area.
- To start execution press F5 or click “Start” button.
- Commands:
  - **FF 82 00 00 06 FF FF FF FF FF FF FF** - cards get authenticated with success 90 00 and Green LED glows.
  - **FF 86 00 00 05 01 00 04 60 00** - success message 90 00 is displayed.
  - Read operation - **FF B0 00 04 10** success Message 90 00 is displayed and the read data is displayed.
  - Write operation - **FF D6 00 04 10 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE 0F** - success message is displayed.
  - Read back - **FF B0 00 04 10** - success message 90 00 is displayed and the written data is displayed.

8.2.3 Get UID command

- Place any card on the antenna, the card gets detected and the ATR appears.
- Enter the following command to get the UID of the card placed.
  - **FF CA 00 00 00** and press “F5” or click on Start button.

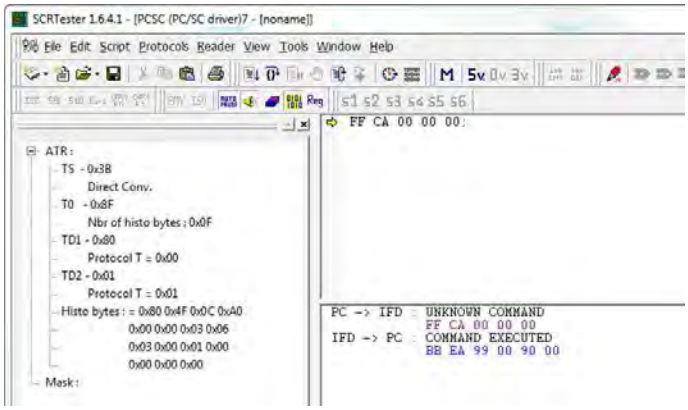

Fig 37. Get UID of the card

8.2.4 User APDU command

- Place any card over the antenna, the card gets detected and the ATR appears.
- Enter the following command to get the version information of the use case.
  - **FF E1 00 00 1C** and press transmit.
  - The response is displayed and in the ACII translation the firmware and use case version can be visualized.

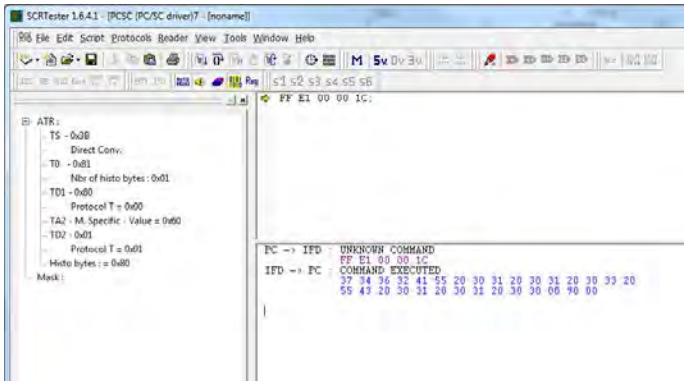

Fig 38. PC/SC diagnostic tool – Send APDU command

8.2.5 User APDU commands – LED control

- Place any card over the antenna, the card gets detected and the ATR appears.
- Enter the following commands to control the LED's present in the board.
- BLUE LED
  - ON – **A0 11**
  - OFF – **A0 10**
- GREEN LED
  - ON – **A0 21**
  - OFF – **A0 20**
- YELLOW LED
  - ON – **A0 31**
  - OFF – **A0 30**
- RED LED
  - ON – **A0 41**
  - OFF – **A0 40**
- ALL LED ON – **A0 A1**
- ALL LED OFF – **A0 A0**

8.2.6 Get challenge command – DESFire cards

- Place the MIFARE DESFire EV1 card over the antenna, the card gets detected and the ATR appears.
- Enter the following command to check the get challenge command.
  - **00 84 00 00 10** and press transmit.
  - The Response is displayed.

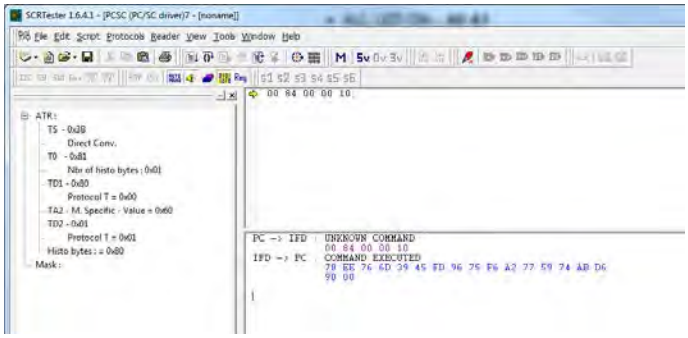

Fig 39. Send APDU command

9. References

[1] PN7462AU Product Support Package -

[2] LPCXpresso IDE - www.nxp.com/redirect/lpcware.com/lpcxpresso/download

[3] UM10883 - http://nxp.com/documents/user_manual/UM10883.pdf

[4] Pay Pass Simulator - http://nxp.com/TERMINALSIMULATOR

[5] SRCTester Tool - http://nxp.com/documents/software/141410.zip

10. Legal information

10.1 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.

10.2 Disclaimers

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

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

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

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

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

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

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

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

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

10.3 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 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

10.4 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.

I²C bus logo — is a trademark of NXP B.V.

ICODE and I•CODE — are trademarks of NXP B.V.

11. 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: 9 March 2016
337010
Document identifier: UM10915