UG10068 PN7220 – Quick start guide Rev. 2.0 – 4 April 2024

User guide

Document information

Information	Content
Keywords	PN7220, Android, NFC Forum, EMVCo
Abstract	This document describes the PNEV7220BP1 and PNEV7220BP2 and how to use it.



1 Introduction

This document describes how to work with PNEV722xBPx. It contains relevant information about hardware characteristics, antennas, software integration, and the PN722x evaluation kit for reference implementation.

Note: Throughout this document, PN7220 and PN7221 may be referred to as "PN722x". Both NFC controllers posses a similar set of features, while PN7221 also includes the Enhanced Contactless Polling (ECP) protocol by Apple. For more information, refer to the product page on nxp.com ([1]).

PN722x has two boards:

- PNEV722xBP1 = single host
- PNEV722xBP2 = dual host

Note: If not stated otherwise, the information presented is valid for both variants.

UG10068 User guide

2 General description of PN722x

PN722x is a complete NFC controller solution with an integrated FW and NCI interface, designed for contactless communication at 13.56 MHz and contact interface via TDA.

It is a solution for quickly integrating NFC technology into other applications, especially using Android OS. Further information on the technical details can be found in the data sheet [2].

3 PNEV722xBPx integration steps

This chapter explains how to combine i.MX M8 (Nano, Mini, ULP) boards and PNEV722xBPx. The first section <u>Section 3.1 "Hardware description"</u> describes the hardware on all three variants of i.MX 8M, PNEV722XBP1, and PNEV722XBP2. <u>Section 3.2 "Software description"</u> describes the software for i.MX 8M and PNEV722xBPx.

3.1 Hardware description

PN722x is a close controller and needs DH to control it. NXP supports an i.MX 8M Nano/Mini/ULP as DH, but integration is possible with other DH.

PNEV722xBPx can be connected to an i.MX 8M board via J27 and J43 connectors.

Note: For proper operation, PNEV722x needs to be supplied with a voltage of 5V and a current of 1.5A via the USB-C connector. The connected cable is required to provide a ferrite choke.

UG10068 User guide

3.1.1 i.MX 8M Nano and Mini

An i.MX 8M Nano and Mini can be connected to PNEV722xBPx via J1003 (highlighted in red in Figure 1 and Figure 2). Table 1 describes the connections between PNEV722xBPx and both variants of i.MX 8M.

Table 1. i.MX 8M Nano and Mini connections to PNEV722xBPx

Pin Name	PNEV722xBPx	i.MX 8M Nano or Mini
VEN	J27 - 7 or J43 - 24	J1003 - 40
IRQ	J27 - 6 or J43 - 23	J1003 - 37
SDA	J27 - 3 or J43 - 21	J1003 - 3
SCL	J27 - 2 or J43 - 19	J1003 - 5
MODE_SWITCH	J43 - 32	J1003 - 38
GND	J27 - 1 or J43 - 1	J1003 - 39

To connect PNEV722xBP2, existing connections (Table 1) must be extended with additional connections. Table 2 shows the extended connections between both variants of i.MX 8M and PNEV722xBP2.

Table 2. i.MX 8M Nano and Mini extended connections for PNEV722xBP2

Pin Name	PNEV722xBP2	i.MX 8M Nano or Mini
I2C_SW	J43 - 14	J1003 - 36
MODE_SW_SP	J43 - 31	J1003 - 21
MODE_SW_SP_DONE	J43 - 26	J1003 - 24

For Android flashing, check Section 5. One HW change is needed to distinguish between flashing Android and running Android. SW1101 and SW1102 (yellow square in the Figure 1 and Figure 2) must be changed as follows:

i.MX 8M Nano:

Note: When changing switches, the board must be powered off.

- Flashing Android (1 10):
 - SW1101: 1 0 0 0 x x x x x x
 - SW1102: x x x x x x x x x x x
- Running Android (1 10): - SW1101: 0 1 0 0 x x x x x x
 - SW1102: x x x x x x x x x x x

i.MX 8M Mini:

Note: When changing switches, the board must be powered off.

- Flashing Android (1 10):
 - SW1101: 1 0 1 0 x x x x x x
 - SW1102: x x x x x x x x x 0
- Running Android (1 10):
 - SW1101: 0 1 1 0 1 1 0 0 0 1
 - SW1102: 0 0 0 1 0 1 0 1 0 0

To power up the i.MX 8M Nano and Mini, use USB-C (highlighted in green in <u>Figure 1</u> and <u>Figure 2</u>). For connecting the board to the PC, use USB-C (highlighted in blue in <u>Figure 1</u> and <u>Figure 2</u>) and micro-USB (highlighted in purple in <u>Figure 1</u> and <u>Figure 2</u>). USB-C will as an Android device on the PC. To set up the environment, check <u>Section 4</u>.



3.1.2 i.MX 8M ULP

The board can be connected to PNEV722xBPx via an Arduino Interface J20 (highlighted in red in <u>Figure 4</u>). To power the ULP board, use P1 (highlighted in blue in <u>Figure 3</u>). Connection to a PC can be achieved via a J15 (highlighted in yellow in <u>Figure 3</u>). <u>Table 3</u> shows the connections between boards.

Table 3. i.MX 8M ULP connections to PNEV722xBPx

Pin Name	PNEV722xBPx	i.MX 8M ULP
VEN	J27 - 7 or J43 - 24	J20 - D9
IRQ	J27 - 6 or J43 - 23	J20 - D13
SDA	J27 - 3 or J43 - 21	J20 - D11
SCL	J27 - 2 or J43 - 19	J20 - D12
MODE_SWITCH	J43 - 32	J20 - D10
GND	J27 - 1 or J43 - 1	J20 - GND

Same as with Nano and Mini, PNEV722xBP2 needs additional connections. <u>Table 4</u> shows what must be added.

 Table 4. i.MX 8M ULP extended connections for PNEV722xBP2

Pin Name	PNEV722xBP2	i.MX 8M ULP (coming soon)
I2C_SW	J43 - 14	to be added
MODE_SW_SP	J43 - 31	to be added
MODE_SW_SP_DONE	J43 - 26	to be added

To switch between flashing Android and running Android, hardware changes are necessary. SW5 (highlighted in red in Figure 3) is used for switching.

Note: When changing switches, the board must be powered off.

- Flashing Android (from 1 to 8): 0 0 0 0 0 0 1 0
- Running Android (from 1 to 8): 0 0 0 0 0 0 0 1



Figure 3. i.MX 8M ULP front



3.1.3 PNEV722xBP1

Figure 5 shows PNEV722xBP1 board. To power up a board either via USB-C (J7) or directly from DH. The NXP recommendation is to connect the power supply via USB-C.

To switch between power supply options, J5 must be changed as follows:

- 1. Power supply via USB-C: Pins 1 and 2 must be connected => Recommended
- 2. Powered via DH: Pins 3 and 4 must be connected.

In option 2, J43 must be used to power up the board.

Table 5. PNEV722xBP1 default jumper settings

Name	Default setting
VBAT (J4)	CONNECTED
VUP (J1)	NOT CONNECTED
VBATPWR (J2)	CONNECTED
AUX2AUX1 (J65)	NOT CONNECTED
SW_ON (J47)	NOT CONNECTED
VBAT_SEL (J3)	NOT CONNECTED
EXT_PWR_SEL (J5)	1-2 CONNECTED

PN722x IC can be found on the module board (highlighted in red in Figure 5).



Figure 5. PNEV722xBP1 front



Figure 6 shows J43 (highlighted in red), which can be used to connect board with i.MX 8M boards.

To switch between 1.8V and 3.3V for VDDIO_SEL, change the solder jumper to 1-3 (3.3V) or 1-2 (1.8V). For more details, check the data sheet [2].

3.1.4 PNEV722xBP2

<u>Figure 7</u> shows the PNEV722xBP2 board. To power up, use either via USB-C (J7) or directly from DH. The NXP recommendation is to power via USB-C.

To switch between power supply options, J5 must be changed:

- 1. Power supply via USB-C: Pins 1 and 2 must be connected => Recommended
- 2. Powered via DH: Pins 3 and 4 must be connected
- 3. USB DEVICE POWER INPUT: Pins 5 and 6 must be connected

In option 2, J43 must be used to power up the board.

Table 6.	PNEV722xBP2	default	jumper	settings
----------	-------------	---------	--------	----------

Name	Default setting
VBAT (J4)	CONNECTED
VUP (J1)	NOT CONNECTED
VBATPWR (J2)	CONNECTED
AUX2AUX1 (J10)	NOT CONNECTED
OpenSDA - MCU - RST (J56)	NOT CONNECTED
VBAT_SEL (J3)	NOT CONNECTED
EXT_PWR_SEL (J5)	1-2 CONNECTED
HOST_SEL0 (J63)	1-2 CONNECTED
J59	CONNECTED
J60	CONNECTED

PN7220 IC can be found on the module board (yellow square in <u>Figure 7</u>). Red square on <u>Figure 7</u> shows K82, which exists only on the PNEV722xBP2 board. To flash it, use a debugger like J-Link and connect it to J35 (orange square on <u>Figure 7</u>). To run examples on K82, some pins must be shorted on J27. <u>Table 7</u> shows what must be shorted.

Table 7. J27 shorted pins

Enabled example	Pins to short
SCL	1 - 2
SDA	3 - 4
Mode Switch SP	5 - 6
Mode Switch SP done	7 - 8
SPI_CITO	11 - 12
SPI_CLK	13 - 14
SPI_CS0	15 - 16
SPI_COTI	17 - 18
IRQ	19 - 20

PN7220 – Quick start guide



Red square on Figure 8 shows J43, which can be used to connect board with i.MX 8M boards.



To switch between 1.8V and 3.3V for VDDIO_SEL, change the solder jumper to 1-3 (3.3V) or 1-2 (1.8V). For more details, check the data sheet ([2])

3.2 Software description

PN722x supports the NCI 2.2 interface, which is suitable for use with the Android OS.

NXP provides changes in AOSP source code and patches. In general NXP provides changes in AOSP code, so that the code is compliant with PN722x features. If there is an interest to build Android for i.MX 8M or other DH and with support for PN722x, check [11].

There is also the possibility to get the prebuild Android images and applications for an i.MX 8M Nano, Mini, and ULP. Images can be downloaded from [16]. For flashing the images and using devices, the environment must be prepared. The following chapter provides instructions for environment setup.

4 Environment setup

The following subsections provide instructions on how to prepare an environment on Windows and Linux.

4.1 Windows environment setup

To enable ADB ([3]) on a Windows computer, additional tools must be installed.

For ADB and Fastboot, SDK Platform Tools must be installed. To do so, follow the instructions below:

1. Download platform-tools from [4] (see Figure 9).

developers 📥	Platform	Android Studio	Jetpack	More 👻	Q Search	English	+ Andr	oid Studio Sign
Filter What's new in SDK Tools SDK build tools release SDK command line tool notes SDK notes SDK notes	notes Is release	If you're an And SDK Manager or right place with But if you want j • Downloa	roid develope r from the so the rest of yo ust these co ad SDK Plat	er, you should get Skmanager comm our Android SDK to mmand-line tools, form-Tools for V	the latest SDK Platform-T nand-line tool. This ensure cools and easily updated. , use the following links: Vindows	ools from Android Studi s the tools are saved to	o's the	On this page Downloads Revisions
SDK platform tools rele	ease notes	Downton	Id SDK Plat	form-loots for P	140			
Command line tools	~	Downlos	ad SDK Plat	form-Tools for L	inux			
		Although these	links do not o	change, they alwa	ys point to the most recer	it version of the tools.		

2. Accept the terms and conditions and click the download button (see Figure 10).



Figure 12).

UG10068

PN7220 – Quick start guide

 Android_drivers_tools
 Open

 Name
 Open with Code

 Image: start all
 Image: start all

 Image: start all
 Image: start

3. After downloading, extract the folder and add "adb.exe" and "fastboot.exe" to the path (see Figure 11 and

🔳 adb.exe 12/31/2007 11:00 PM 5,854 KB Application AdbWinApi.dll 12/31/2007 11:00 PM Application exten... 96 KB AdbWinUsbApi.dll 12/31/2007 11:00 PM Application exten... 62 KB dmtracedump.exe 12/31/2007 11:00 PM Application 237 KB etc1tool.exe 12/31/2007 11:00 PM Application 430 KB 📑 fastboot.exe 12/31/2007 11:00 PM Application 1,606 KB hprof-conv.exe 12/31/2007 11:00 PM Application 43 KB libwinpthread-1.dll 12/31/2007 11:00 PM Application exten... 227 KB make_f2fs.exe 489 KB 12/31/2007 11:00 PM Application make_f2fs_casefold.exe 12/31/2007 11:00 PM Application 489 KB 🎽 mke2fs.conf 12/31/2007 11:00 PM **CONF File** 2 KB 📑 mke2fs.exe 12/31/2007 11:00 PM Application 746 KB VOTICE.txt 12/31/2007 11:00 PM 2,768 KB Normal text file source.properties 12/31/2007 11:00 PM Properties Source ... 1 KB 📧 sqlite3.exe 12/31/2007 11:00 PM Application 1,190 KB Figure 12. Add adb.exe and fastboot.exe

Instructions for adding tools to the path can be found under this link: [5].

After install the win-usb driver, follow the instructions below:

1. Choose the driver from the webpage [6] (see Figure 13).

developers 📥	Platform	Android Studio	Jetpack	More *	Q Search	⊕ English ▼	Android Studio	Sign in
Download A	ndroid Studio editor	Android Grad	lle Plugin	SDK tools	Preview			
▼ Filter		* Note: If hardwa	you're developi re device	ng on macOS or Li	nux, then you do not need to inst	tall a USB driver. Instead see <u>P</u>	n aces on a	
 Run your app on a h device 	hardware	There are two v	vays to downl	oad the Google	USB Driver for Windows:			
Get started	dian 1	• Downlo	ad the Goog	le USB Driver	ZIP file (ZIP),			

2. Accept the terms and conditions and click the download button (see Figure 14).



3. Extract all and right-click to "android_winusb.inf" and press "Install" (see Figure 15).



UG10068 User guide After the steps are completed, an i.MX 8M Nano/Mini/ULP can be connected to the computer. In the Device Manager, check if the device appears in the list. The goal is to have the device listed as an Android device (Figure 16).

> 🔙 Android Device	
Figure 16. Android device in Device Manager	

It can occur that the computer does not recognize the i.MX 8M as an Android device when it first starts, but lists it under "Other devices" as "Unknown device" or "Android". Check Figure 17.



Installing the correct driver for the device:

- 1. Right-click to "Unknown device" or "Android" and click "Update driver"
- 2. "Browse my computer for drivers" (see Figure 18).

How do you want to search for	
→ Search automatically for drivers Windows will search your computer for the best available driver and install it on your device.	
→ Browse my computer for drivers Locate and install a driver manually.	
Cancel Cancel	

PN7220 – Quick start guide

3. Click: "Let me pick from a list of available drivers on my computer" (see Figure 19).



4. Select "Android device" (see Figure 20).

	Select your device's type from the list below.
	Common hardware types:
	Show All Devices Android Device Audio inputs and outputs Audio Processing Objects (APOs) Audio/video control devices Batteries Biometric devices Bluetooth Cameras Compute accelerators Computer
	Next Cancel
Figure 20. And	droid Device

UG10068

PN7220 – Quick start guide

5. Select "Android Composite ADB Interface" (see Figure 21).

5	Select the device driver you want to install for th	is hardware.						
	Select the manufacturer and model of your hardware device and then click Next. If you have a disk that contains the driver you want to install, click Have Disk.							
	Model Android ADB Interface Android Bootloader Interface Android Composite ADB Interface							
[This driver has an Authenticode(tm) signature. <u>Tell me why driver signing is important</u>	Have Disk						
		Next Cancel						
Figure 21. Andro	d Composite ADB Interface							

6. Click "Yes" (Figure 22).

	Update Driver Warning	\times
	Installing this device driver is not recomme Windows cannot verify that it is compatibl hardware. If the driver is not compatible, y not work correctly and your computer migl or stop working completely. Do you want installing this driver?	nded because e with your your hardware will ht become unstable to continue
	Yes	s No
Figure 22. Click Yes		

The next step is flashing the i.MX 8M with Android images.

4.2 Linux environment setup

Reference Linux version: Ubuntu 22.04.2 LTS.

On Linux, preparing the environment can be achieved with the following the steps to add adb and fastboot support:

- 1. Download zip from [4]
- 2. Unzip
- 3. Open a terminal and type "sudo nano .bashrc"
- 4. Add "export PATH=\${PATH}:/path/to/adb_fastboot" for example

export PATH=\${PATH}:/home/nxp/Downloads/platform-tools

The next step is flashing the i.MX 8M with Android images.

5 Installing prebuild images

Prebuild images can be downloaded from [16]. On the webpage, click "Design Resources" (Figure 23)

EMV L1 Compliant NFC Controller with NCI Interface Supporting EMV and NFC Forum Applications

Overview	Product Details	Documentation	Design Resources 🛈	Support	BUY/PARAMETRICS PACKAGE/QUALITY
Figure 23.	Design resourc	ces			

After clicking this button, the page automatically scroll to the "Design Resources" section. In this section, look for the "Software" section. All prebuild images are located there (see Figure 24). It is important to know which evaluation board is used (PN722xBP1 (Single Host) or PN722xBP2 (Dual host)). Which version of the board should be used with a specific prebuild image is highlighted in blue in Figure 24. It is also important to know, which i.MX8 board variant is used by the customer (highlighted in orange in Figure 24).

Design Resources	3	
Software		
• NXP (4)	√ Filter by keyword	
FILTER BY Design Software Configuration Tools	4 software files CONFIGURATION TOOLS NFC Antenna Design Hub	Sort by Relevance
Embedded Software	BSPS AND DEVICE DRIVERS PN7220_IMX8M_Nano_13_21_00_Single_host NEW ZIP Rev 13.21.00 Feb 26, 2024 1658343 KB PN7220_IMX8M_Nano_Single_host	DOWNLOAD
Figure 24. Download prebui	ld images	

When the correct prebuild image is found, click "Download". Sign into your NXP account. You will be asked to accept the **NXP software license agreement** in order to download the files.

When the download is completed, unzip the files and follow the next steps:

- 1. Go into the unzipped directory
- 2. Open "Command Prompt"
- 3. Before running any command, configuration on the i.MX8 board must be in "Flash Android" mode. Check Section 3.1.1 or Section 3.1.2
- 4. Run the following command:

./uuu imx android flash.bat -f imx8mX -a -e -d pn7220

Note: Depend on the I.MX8 board that is used, imx8mX is different (imx8mn = i.MX 8M Nano, imx8mm = i.MX 8M Mini, imx8ulp = I.MX8 ULP)

5. <u>Boards need to be connected</u> shows the expected output.



- 6. Put the switch back to "Running Android" mode. Check Section 3.1.1 or Section 3.1.2.
- 7. Run the device.

Booting of the device takes some time as the Android boot process is now performed. After booting, the user can begin using the device like a normal Android device.

After booting, open the Extension folder in the downloaded packet, and run "flash.bat" <u>Figure 26</u>. This .bat file installs additional software on the device.



6 ADB commands explanation

After the images are flashed to the i.MX 8M, and the PNEV722xBPx is connected, adb logs can be used to monitor the communication. To check the communication, a terminal (command prompt on Windows) must be open.

There are various commands that can be used to check if PNEV722xBPx and i.MX 8M are communicating with each other. The basic adb commands are:

- adb devices: list all available devices currently connected to the computer
- fastboot devices: list all available devices in bootloader mode that are connected to the computer
- adb logcat: print all logs from Android
- adb shell: enter Android device terminal
- adb pull: pull files from DH to the PC
- **adb push**: push files to Android devices. To use this command, the device must be in root mode and remounted. This can be done with:
 - adb root: to root the device as the root user
 - adb remount: to remount the device
- adb install: install any *.apk (for example: TagInfo)

With the commands listed above, numerous tasks can be completed when debugging the setup. One of the most important commands is adb logcat, which prints all logs from Android as output. By using additional inputs and a combination of commands, the output can be restricted to specific data.

adb shell logcat | grep "NxpNci" #this command prints only lines where NxpNci is mentioned. Check Figure 27.

Instead of NxpNci, the following values can also be entered:

- NxpNciX: to get only NFC Forum NCI commands send from DH
- NxpNciR: to get only NFC Forum NCI commands received to DH
- EMVCoNciX: to get only EMVCo profile commands send from DH
- EMVCoNciR: to get only EMVCo profile commands received to DH
- "nfc" or "emvco": with one of these two words, specific data can be generated via logcat.

Commands can be combined as follows:

logcat | grep -i -e emvco -e nxpnci.

>adb	shel	11
db845c:/ # logcat g	rep '	"NxpNci"
05-20 11:12:46.433	453	2156 D NxpNciX : len = 4 > 21060100
05-20 11:12:46.433	453	2155 D NxpNciR : len = 4 > 41060100
05-20 11:12:46.434	453	2156 D NxpNciX : len = 6 > 20030301A00F
05-20 11:12:46.435	453	2155 D NxpNciR : len = 40 > 4003250001A00F20000000000000000000000000000000
00000000000000		
05-20 11:12:46.435	453	2158 D NxpHal : phNxpNciHal_print_res_status: response status =STATUS_OK
05-20 11:12:46.435	453	2156 D NxpNciX : len = 4 > 20000100
05-20 11:12:46.440	453	2155 D NxpNciR : len = 4 > 40000100
05-20 11:12:46.440	453	2155 D NxpNciR : len = 13 > 60000A02002004050053030100
05-20 11:12:46.440	453	2158 D NxpHal : NxpNci> FW Version: 3.1.0
05-20 11:12:46.451	453	2158 D NxpHal : Terminating phNxpNciHal client thread
05-20 11:12:46.451	453	2158 D NxpHal : NxpNciHal thread stopped
05-20 11:12:46.466	453	453 D NxpHal : phNxpNciHal_close - phOsalNfc_DeInit completed
05-20 11:12:46.466	453	453 D NxpHal : phNxpNciHal_release_info mGetCfg_info

Figure 27. Logcat example

For more information on adb, refer to [3].

UG10068

© 2024 NXP B.V. All rights reserved.

7 Test applications

At this point, communication between PNEV722xBPx and i.MX 8M should be established. Communication with the card should be working, this can be verified with a adb logcat command. The last step is to use the test applications provided by NXP. The test applications can be found in the package located in [17].

Note: All native applications must be built for the selected host to perform as intended. NXP provides native applications only for i.MX 8M boards. If the selected host is different, use the source code from [12] and build it.

The script "flash.bat" will install test applications with the following commands:

• adb root

- adb remount
- adb push test app name /location/location
- adb shell chmod 0777 /location/location => gives executable rights to the application

Example:

```
$adb root
$adb remount
$adb push image/EMVCoAidlHalComplianceTest/EMVCoAidlHalComplianceTest system/etc
$adb shell chmod 0777 /system/etc/EMVCoAidlHalComplianceTest
```

To try the application, go to the location where the application is installed. For example, to run EMVCoAidlHalComplianceTest, the following commands must be executed in the terminal:

```
$adb shell
$cd system/etc
$./EMVCoAidlHalComplianceTest Type AB
```

The following subsections describe all test applications provided by NXP. DTA and EMVCo compliance execution applications are provided by NXP to enable customers to run NFC Forum compliance or EMVCo compliance, and are not required for basic functionality testing.

7.1 Test applications only for PNEV722xBP1

This section describes the test applications for the PNEV722xBP1 board, and gives instructions on installation and how to use them.

7.1.1 EMVCo Compliance Execution

This application is running an EMVCo loopback application and is part of the release packet:

To install it, run the following commands:

```
$adb root
$adb remount
$adb push EMVCoAidlHalComplianceTest /system/etc/
$adb shell
$cd system/etc/
$chmod 0777 EMVCoAidlHalComplianceTest
```

Open terminal (command prompt on Windows) and run:

```
$adb shell
$cd system/etc
$./EMVCoAidlHalComplianceTest type AB
```

Figure 28 shows the output of a successful run of test application.



If invalid parameters are passed to the EMVCoAidIHalCompliance application, it will fail to run. See Figure 29 for the expected output in this case.



Figure 29. EMVCo compliance execution: Fail

7.1.2 EMVCo Mode Switch

This application shows the usage of the Mode Switch API. It does not run any EMVCo loopback application in the background. To test communication when PN722x works in EMVCo mode, use other applications.

To install this application, download it from [16], and run the following commands:

```
$adb root
$adb remount
$adb install EMVCoModeSwitchApp.apk
```

To run Android emulation use Vysor [7] or any similar tool. Navigate to the EMVCo mode switch application. Figure 30 shows default view of application. By default, NFC mode is active on boot, so the NFC indicator is shown in green and the EMVCo indicator in red.

EMVCoMode	Switch		Û
Use EMVCo	🗋 Туре В	🗌 Type F	
	NFC ON	EMVCO OFF	-
re 30. EMVCo mode switch applicati	ion: default view	• • •	1

UG10068

PN7220 – Quick start guide

Select different technologies and enable the "Use EMVCo" switch. If valid technologies are selected and EMVCo mode is selected, the EMVCo indicator glows with green color, and the NFC indicator glows with red color. See <u>Figure 31</u> for reference.

12:52	0	
EMVCoModeSwitch		
🗹 Type A 🛛 🗹 Type B	Type F	
Use EMVCo	•	
NFC OFF	EMVCO ON	
EMVCo Poll Activated		

UG10068

PN7220 – Quick start guide

If an error occurs, invalid technologies are selected and EMVCo mode is not activated, the EMVCo indicator is shown in red and the NFC indicator in green. Try again with a different technology combination. See Figure 32 for reference.

		12:52			Û
		EMVCoModeS	witch		
		Туре А	🗌 Туре В	🗌 Type F	
		Use EMVCo			0.0
		NE	C 0N	EMVCO OFF	
				EMVCOOFF	
		EMVCo start failed.	Select valid technolgy/tech	nolgy combination to pol	
			× 📉 🕏		
32. EN	MVCo mode sw	/itch: failure			

Steps to disable the EMVCo poll:

- Disable the "Use EMVCo" switch. This enables NFC.
- The EMVCo indicator is shown in red and the NFC indicator in green.

7.1.3 EMVCo CT test application

The application is part of the following release packet [16]:

To install it, run:

```
$adb root
$adb remount
$adb push EMVCoAidlHalTDATest /system/etc/
$adb shell
$cd system/etc/
$chmod 0777 EMVCoAidlHalTDATest
```

Open the terminal (command prompt on Windows) and run:

```
$adb shell
$cd system/etc
$./EMVCo EMVCoAidlHalTDATest type AB CT
```

A contact card can be inserted into the TDA.



To get logs, the adb logcat with EMVCo greb must be used. Check Figure 34.

12-01 21:5			8 > 2103050200010101
12-01 21:5	4:02.240 457		4 > 41030100
12-01 21:5	4:02.242 457	457 D EMVCoNciX: len =	4 > 21060100
12-01 21:5		6075 D EMVCoNciR: len =	4 > 41060100
12-01 21:5	4:02.299 457	457 D EMVCoNciX: len =	5 > 2201022001
12-01 21:5	4:02.301 457	6075 D EMVCoNciR: len =	4 > 42010100
12-01 21:5	4:02.345 457	6075 D EMVCoNciR: len =	4 > 62010100
12-01 21:5	4:02.348 457	6075 D EMVCoNciR: len =	38 > 62002320000100010011C3BDF18FF81F1FE43003F07834D49464152452053414D204156333000
12-01 21:5			9 > 200406030101022000
12-01 21:5			7 > 40040400FF010A
12-01 21:5			23 > 0A001400A404000E315041592E5359532E444446303100
12-01 21:5			6 > 600603010A01
12-01 21:5			5 > 0A00026E00
12-01 21:5			23 > 0A001400A404000E315041592E5359532E444446303100
12-01 21:5			6 > 500503010A01
12-01 21:5			5 > 0A00026E00
			-1
igure 34. Commu	nication v	with contact car	a

UG10068

7.1.4 EMVCo Transac test

The application is part of the following release package [16]:

To install, run:

```
$adb root
$adb remount
$adb push EMVCoAidlHalTransacTest /system/etc/
$adb shell
$cd system/etc/
$chmod 0777 EMVCoAidlHalTransacTest
```

Open the terminal (command prompt on Windows) and run:

```
$adb shell
$cd system/etc
$./EMVCo EMVCoAidlHalTransacTest Type A
```

2|db845c:/system/etc # ./EMVCoAidlHalTransacTest Type A

Valid Technology selected for polling

Figure 35. EMVCoAidIHalTransacTest application run

7.2 Test applications only for PNEV722xBP2

This section describes the test applications for the PNEV722xBP2 board. It provides instructions for installation and usage. As the BP2 board uses a secure MCU, all EMVCo related applications are executed on this secure MCU.

7.2.1 Secure MCU mode switch application

This application is part of the release package that can be found [16]. The application is used to switch the host from Android to K82.

To install, run:

```
$adb root
$adb remount
$adb push SmcuSwitchV2_0 /system/lib64/
$adb shell
$cd system/etc/
$chmod 0777 SmcuSwitchV2 0
```

Open terminal (command prompt on Windows) and run:

```
$adb shell
$cd system/lib64
$SmcuSwitchV2_0
```

For this application, the K82 on the PNEV722xBP2 board is also used. For more information on dual-host setup and how to flash examples on K82, refer to [13].



Figure 36. SmcuSwitchV2_0 application

7.3 Combined test applications

Applications in this section can be used with both variants of the boards, PN7220 and PN7221.

7.3.1 DTA application

This application is responsible for runningning the NFC Forum compliance test. It can be found under [16].

NP	_	Device Test A	Application		
		Cert.Rel CR13	V-F Con.Dev		
	Execution Mode		00 👻	Pattern Number	Custom
		RF Techn	ology		
		Poll (Passive)	Listen (Passive)		
		ABFV	ABFV		
		P2P 🖉 🗌 🖉 🗌	✓ P2P		
		RD/WR Ø Ø Ø			
			ESE		
		P2P-ACM (Initiator)	P2P-ACM (Target)		
		A B F V	A B F V		
		P2P-ACM	P2P-ACM		
			Params		
		Utilitie	05		
Log Messages	Device				
File Enter File Name	Info FirmWare Version: XX.X				
Console	DTA Lib Version: XX.X				
Auto Mode Messages	Current Status: Stopped				

Figure 37. DTA

		P2P-ACM (Initiation) P2P-ACM (Target) A B V A B V P2P-ACM P2P-ACM P2P-ACM P2P-ACM P2P-ACM		
		LLCP DONNECT POU Params		
		Utilities		
Log Messages File Enter File Name Console	Device UI Version: XX.X FirmWare Version: XX.X MiddleWare Version: XX.X DTA Lis Version: XX.X			
Auto Mode Messages	Current Status: Stopped			
Custom Message Show Message	ge			
Run Stop Exit				
Figure 38. DTA run button				

• The "Cert.Rel" field must reflect the Certification Release version targeted.

- The "TSN-F" field defines the NFC-F technology Time Slot Number and must be set according to the test execution requirement.
- The "Con.Dev" field defines the Connection Device Limit and must be set according to the test execution requirement.
- Only the "Manual" option of "Execution Mode" is available for now, "Auto" mode being reserved for future use.
- "Pattern Number" must be set according to the test execution requirement.

PN7220 – Quick start guide

- The RF technology tabs allow selecting individually each technology for each possible mode.
- The "LLCP" field allows enabling specific "Pattern Number" for dedicated test execution.
- The "SNEP" field allows running dedicated tests, requiring also the "Android Beam" feature been enabled in the Android device settings.
- The "Log messages" field allows to output the trace to a file (under "/sdcard/nxpdtalog/" folder) and/or a console.

7.3.2 NfcTdaTest

This application is used for testing PN722x with SAM cards. The Application can be found under [16].

To perform this test, TDA8035 must be installed in the PNEV722xBPx board. Refer to [14] for instructions.

Open terminal (command prompt on Windows) and run:

```
$adb root
$adb remount
$adb install NfcTdaTest.apk
```

Figure 39 shows how to open the application.

	6	Q, Search apps					
		Calendar	Camera	Contacts		Files	Gallery
			NP		Q	1Ô1	
		JRCP_Proxy	NfcTdaTest	NXP Device Test App	Search	Settings	WebView Browser Te
							•
igure 39.	Open Nfc	TdaTest app	lication				

PN7220 – Quick start guide

11.14		
NicTdaDemoA		
C BAART	123 534042	The local sector in the sector of
Discover TDA info h		
Discover TDA Info N	iot Available	
APDU CMD		
APE	Use the taskbar to switch apps	
		- HELST
ollow the instructions		

In the pop-up, click "next" until the application is not open. Check Figure 40.

UG10068

PN7220 – Quick start guide

	🔤 db845c	– 🗆 ×	
	NfcTdaDemoApp		
	TDA_DISCOVER_DONE		
	SAM1 SAM2	OPEN TDA	
	TDA ID - 0x21 SAM1 Cord		
	Status : 0x1 Connected and Disabled Num. of Protocols : 0x1		
	Protocol Value : 0x0 Num Of Cardinfo : 0x0 Cardinfo Not Available !!!!		
	TDA ID : 0x22 SAM2 Card		
	Num, of Protocols : 0x1 Protocol Value : 0x0		
	CardInfo Not Available !!!!		
	APDU CMD :	SEND APDU	
	APDU RSP :		
	CLOSE TDA		
	🖂 🔀 😨 🖽 🖶	< • =	
Figure 41 Discover con	nocted smart cards over TDA		
Figure 41. Discover com	Hecteu Smalt Calus Over TDA		

Click "DISCOVER TDA" to find the smart card connected over TDA. The button changes the color to green and the text to "TDA_DISCOVER_DONE". Check <u>Figure 41</u>.

Г

UG10068

PN7220 – Quick start guide

	db845c	- • ×
	NfcTdaDemoApp	
	DISCOVER TDA	
	SAM1 SAM2	CID: 0X0B
	TDA ID : 0x21 SAM1 Card Status : 0x0 Connected and Enabled Num. of Protocols : 0x1 Protocol Value : 0x0 Num Of Cardinfo : 0x1 Type : 0x1 Length : 0x1c Value : 3BDF18FF81F1FE43003F07834D49464152452053414D	2041 56333000
	TDA ID: 0x22 SAM2 Card Status: 0x1 Connected and Disabled Num. of Protocols: 0x1 Protocol Value: 0x0 Num Of CardInfo: 0x0 CardInfo Not Available !!!!	
	APDU CMD : 08001400A404000E315041592E5359532E44444630310 APDU RSP :	0 SEND APDU
	CLOSE TDA	
		< • •
oure 42. Open a spec	ific SAM slot	

Select the SAM slot and click "OPEN TDA". See Figure 42.

PN7220 – Quick start guide



Send the APDU to Contact card using send "SEND APDU". See Figure 43.

Г

PN7220 – Quick start guide

the Idaberriore	DISCOVER TDA		
C SAM1	SAM2	OPEN TOA	
Discover TDA Info No	Available		
Discover TDA info No	K Available		
APDU CMD :		SEND APOL	
APDU RSP:	CLOSE TDA		
	E 🔚 🖸 🖬 🚭		

The TDA can be closed with the "CLOSE TDA" button. See Figure 44.

7.3.3 Other useful applications

Other NXP applications can also be used, for example:

- TagInfo Application [8]
- TagWriter Application [8]
- CTS Verifier Application [9]

The TagInfo and TagWriter application can be installed through the Android Play Store.

To install the CTS Verifier Application, follow the instructions under [9].

UG10068 User guide

7.3.4 NFC Cockpit

PN722x is the first NXP NCI-based controller that supports the NFC Cockpit tool. It can be downloaded from [15].

Install the NFC Cockpit and follow the instructions below to run the NFC Cockpit with PN722x:

Note: Support started from NFC Cockpit version 8.0.0

1. Go into the installation folder of NFC Cockpit: For example: "C:\nxp\NxpNfcCockpit_v8.0.0.\firmware\ Secondary_Pn722x\Android"

Name	Date modified	Туре	Size
🔤 PN722x_UcBalAndroid_LogDisabled.bat	2/9/2024 9:56 AM	Windows Batch File	2 KB
PN722x_UcBalAndroid_LogEnabled.bat	2/9/2024 9:56 AM	Windows Batch File	2 KB
pnscr_nfc	2/9/2024 9:56 AM	File	91 KB
🧟 readme.txt	2/9/2024 9:56 AM	Normal text file	3 KB
restore_NFC_HAL.bat	2/9/2024 9:56 AM	Windows Batch File	1 KB

2. Run "PN722x_UcBalAndroid_LogDisabled.bat" or "PN722x_UcBalAndroid_LogEnabled.bat" and follow the instructions in Command Prompt



PN7220 – Quick start guide

3. Enter:

cd system/bin

4. Enter the command below (see Figure 47 for expected output):

pnscr true

```
evk 8mn:/ # cd system/bin
evk_8mn:/system/bin # pnscr true
NNC uC ADBSocket 04.00.00
 ***********
13:49:08.663
            SEND :
                     4 > 20000100
            RCEIVE :
                    4 < 40000100
13:49:08.670
            RCEIVE : 13 < 60000A02002004052053030201
13:49:08.672
13:49:08.672
            SEND :
                    5 > 2001020000
            RCEIVE : 23 < 4001140000060000014000FF0000FF0003010002008000
13:49:08.677
13:49:08.677
            SEND
                     4 > 2F000100
                     4 < 4F000100
13:49:08.679
            RCEIVE :
Socket successfully created..
Listening at port number 8059
Socket successfully binded
Server listening..
```

Figure 47. Run pnscr

- 5. Open the NFC Cockpit application on the PC and start using it.
- 6. When the NFC Cockpit is not needed anymore, close the GUI and run:

restore_NFC_HAL.bal

It is important to run restore_NFC_HAL.bal when the user is finished with using the NFC Cockpit, otherwise the MW will not be able bootup.

8 Firmware update

The firmware (FW) update procedure is different between both use cases of PN722x. On a single host (PNEV722xBP1), the firmware update is done via the Android host. In dual host (PNEV722xBP2), the firmware update must be performed via a secure MCU.

8.1 Single host

To update FW of the PN722x, the .so file must be pushed to the DH. After reset the NFC stack, MW checks if FW on PN722x and FW in the .so file are different. If yes, the FW update procedure is started automatically.

FW can be found [10].

How to update FW:

- Go to the location of *libpn72xx_fw.so*
- Open a terminal and run the following commands

```
$adb root
$adb remount
$adb push "libpn7220_fw.so" vendor/lib64/libpn72xx_fw.so
$adb shell svc nfc disable
$adb shell svc nfc enable
```

Note: The name of the .so file can be different. In this case "libpn7220_fw.so", must be replaced with the actual name.

After running the above commands, FW should be updated.

Checking if FW has been successfully update:

Save the adb logs, as described in <u>Section 6</u>, then open the log file and search for "FW". <u>Figure 48</u> shows the relevant parts and the current version of FW.

2158 D Nxp <mark>Fw</mark> Dnld:	Send Success
2158 D Nxp <mark>Fw</mark> Dnld:	Response timer started
2158 D Nxp <mark>Fw</mark> Dnld:	phNxpNciHal fw dnld chk integrity cb - Request Successful
2158 D Nxp <mark>Fw</mark> Dnld:	pn72xx phNxpNciHal fw dnld chk integrity cb - Valid Resp Buff!!
2158 D Nxp <mark>Fw</mark> Dnld:	crc status code area len 0x6
2158 D Nxp <mark>Fw</mark> Dnld:	crc status code data len 0x11
2158 D Nxp <mark>Fw</mark> Dnld:	crc status code area 0xffff803f
453 D Nxp <mark>Fw</mark> Dnld:	Processing Normal Sequence
453 D Nxp <mark>Fw</mark> Dnld:	Initializing Sequence
453 D Nxp <mark>Fw</mark> Dnld:	Response Timer Created Successfully
453 D Nxp <mark>Fw</mark> Dnld:	Inserting FrameId
453 D Nxp <mark>Fw</mark> Dnld:	Frame created successfully
453 D Nxp <mark>Fw</mark> Dnld:	phDnldNfc_GetDieId Request submitted successfully
2158 D Nxp <mark>Fw</mark> Dnld:	Send Success
2158 D Nxp <mark>Fw</mark> Dnld:	Response timer started
453 D Nxp <mark>Fw</mark> Dnld:	phNxpNciHal_ <mark>fw_</mark> dnld_complete: Download Status = 0x0
453 E Nxp <mark>Fw</mark> Dnld:	Fw Download success
453 E Nxp <mark>Fw</mark> Dnld:	Returning Download Failed Status to Caller!!
453 E Nxp <mark>Fw</mark> Dnld:	free library SUCCESS !!
453 D Nxp <mark>Fw</mark> Dnld:	phNxpNciHal_ <mark>fw</mark> _dnld_complete : SUCCESS
453 D Nxp <mark>Fw</mark> Dnld:	fragment len set 22a
453 D NxpFwDnld:	Freeing Mem for Dnld Context
453 D NxpHal : p	hNxpNciHal_Update <mark>Fw</mark> Status Enter
453 D NxpHal : p	property_set_intf, key[nfc. <mark>fw</mark> .downloadmode_force], value[0]
453 D NxpHal : p	hNxpNciHal setSystemProperty : Enter Key = nfc.fw.downloadmode_force, value = 0
2158 D NxpHal : N	xpNci> FW Version: 3.1.0
2158 D NxpHal : N	xpNci> FW Version: 3.1.0
453 D Nxp <mark>Fw</mark> Dnld:	gphNxpNciHal_ <mark>fw</mark> _IoctlCtx.bClkSrcVal = 0x1
453 D Nxp <mark>Fw</mark> Dnld:	gphNxpNciHal_ <mark>fw</mark> _IoctlCtx.bClkFreqVal = 0x8
2158 D NxpHal : N	xpNci> FW Version: 3.1.0
2158 D NxpHal : N	xpNci> FW Version: 3.1.0
Figure 48. FW Upda	ite logs

FW version can be checked anytime, even if the FW update procedure was skipped (see red highlight square in Figure 48).

By default in *libnfc-nxp.conf*, the FW update is blocked with the flag. To enable the FW update, the following steps must be performed:

- 1. Open the command prompt
- 2. Run the following command:

adb pull /vendor/etc/libnfc-nxp.conf

3. Open the pulled command and change the flag to 0x01, or 0x02 or 0x03 => 0x02 is recommended. **Note:** If 0x03 is selected, be aware that this option is not for production, but only for debugging purposes. Since FW is written into EEPROM, frequent overwriting can damage the PN722x.

#######################################				
<pre># Flashing Options Configurations</pre>				
<pre># FLASH_UPPER_VERSION 0x01</pre>				
<pre># FLASH_DIFFERENT_VERSION 0x02</pre>				
# FLASH_ALWAYS 0x03				
<pre># FLASH_BLOCKED 0x04 (default value)</pre>				
NXP_FLASH_CONFIG=0x04				
Figure 49. Configuration for FW update				
After the change is done, save the file				
5. Run the following commands:				

```
adb root
adb remount
adb push libnfc-nxp.conf /vendor/etc/
adb reboot
```

After the reboot is done, the FW update starts.

UG10068

8.2 Dual host

FW update is executed via a secure MCU. NXP provides the NciLib ([17]) and "Secure MCU mode switch" application to show how to perform FW update on a dual host with PNEV722xBP2 board. For more information on how to run the applications, refer to [13].

9 Configuration files

Using configuration files, settings can be changed on PN722x. Five configuration files can be stored to a specific location on DH.

Table 8. Configuration file and location

Configuration filename	Location on DH
libemvco-nxp.conf	/vendor/etc/
libnfc-nci.conf	/system/etc/
libnfc-nxp.conf	/vendor/etc/
libnfc-nxp-eeprom.conf	/vendor/etc/
libnfc-nxprfExt.conf	/vendor/etc/

For a detailed explanation of these settings, refer to [11].

To push the configuration files to a specific location, the following commands must be executed,

\$adb root
\$adb remount
\$adb push config_file_name /config_file_location/

It is also possible to pull the configuration files from DH

\$adb pull config_file_location/config_file_name

UG10068

PN7220 – Quick start guide

10 Abbreviations and acronyms

Table 9. Abbrev	Table 9. Abbreviations				
Acronym	Description				
HW	hardware				
SW	Software				
OS	Operating System				
MHz	Mega Hertz				
NFC	Near Field Communication				
NCI	Near Field Communication Controller Interface				
FW	Firmware				
MW	MiddleWare				
DH	Device Host				
V	Volt				
AOSP	Android Open Source Project				
ADB	Android Debug Bridge				
DTA	Device test application				
SDK	Software Development Kit				

UG10068 User guide

11 References

- Web page PN7220 EMV L1 Compliant NFC Controller with NCI Interface Supporting EMV and NFC Forum Applications (<u>link</u>)
- [2] Datasheet PN7220 EMV L1 Compliant NFC Controller with NCI Interface Supporting EMV and NFC Forum Applications (<u>link</u>)
- [3] Resources ADB (link)
- [4] Resources Platform-Tools (link)
- [5] Webpage Add Tool Locations to the PATH Environment Variable (link)
- [6] Resources Win-usb driver (link)
- [7] Web page Vysor (link)
- [8] Web page TagInfo and TagWriter applications(<u>link</u>)
- [9] Webpage CTS Verifier (<u>link</u>)
- [10] Resources PN722x FW (<u>link</u>)
- [11] Application note AN13971: Android porting guide (link)
- [12] Resources PN722x MW (link)
- [13] Application note AN14224: How to use PN7220 in Dual-Host mode (link)
- [14] Application note AN14225: How to use PN7220 with contact cards (link)
- [15] Web page NFC Cockpit (link)
- [16] Web page Prebuild Android images (<u>link</u>)
- [17] Webpage NciLib_PUB (link)

12 Radio Equipment Directive (RED)

The following information is provided per Article 10.8 of the Radio Equipment Directive 2014/53/EU:

(a) Frequency bands in which the equipment operates.

(b) The maximum RF power transmitted.

Table 10. Characteristics

PN	RF Technology	(a) Freq Ranges (EU)	(b) Max Transmitted Power
PN7220BP1	NFC	13.56 MHz ±7 kHz	-11 dBm

EUROPEAN DECLARATION OF CONFORMITY (Simplified DoC per Article 10.9 of the Radio Equipment Directive 2014/53/EU). This apparatus, namely PN7220BP1, conforms to the Radio Equipment Directive 2014/53/EU.

The full EU Declaration of Conformity for this apparatus can be accessed in the future following this link: <u>https://www.nxp.com/products/rfid-nfc/nfc-hf/nfc-readers/emv-l1-compliant-nfc-controller-with-nci-interface-supporting-emv-and-nfc-forum-applications:PN7220</u>.

13 Note about the source code in the document

Example code shown in this document has the following copyright and BSD-3-Clause license:

Copyright 2023-2024 NXP Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials must be provided with the distribution.
- 3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

PN7220 – Quick start guide

14 Revision history

Table 11. Revision history

Document ID	Release date	Description
UG10068 v.2.0	04 April 2024	 <u>Section 3 "PNEV722xBPx integration steps"</u> updated. <u>Section 5 "Installing prebuild images"</u> updated. <u>Section 7 "Test applications "</u> added. <u>Section 8 "Firmware update"</u> added. <u>Section 9 "Configuration files"</u> added.
UG10068 v.1.0	11 July 2023	Initial version

UG10068

PN7220 – Quick start guide

Legal information

Definitions

Draft — A draft status on a document indicates that 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 in a draft version of a document and shall have no liability for the consequences of use of such information.

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). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

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.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

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.

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

UG10068

PN7220 – Quick start guide

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at <u>PSIRT@nxp.com</u>) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

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.

Trademarks

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

NXP — wordmark and logo are trademarks of NXP B.V.

Apple — is a registered trademark of Apple Inc.

i.MX — is a trademark of NXP B.V.

PN7220 – Quick start guide

Tables

Tab. 1.	i.MX 8M Nano and Mini connections to	
	PNEV722xBPx5	
Tab. 2.	i.MX 8M Nano and Mini extended	
	connections for PNEV722xBP25	
Tab. 3.	i.MX 8M ULP connections to	
	PNEV722xBPx7	
Tab. 4.	i.MX 8M ULP extended connections for	
	PNEV722xBP27	

Figures

Fig. 1.	i.MX 8M Mini6	
Fig. 2.	i.MX 8M Nano6	
Fig. 3.	i.MX 8M ULP front8	
Fig. 4.	i.MX 8M ULP back8	
Fig. 5.	PNEV722xBP1 front9	
Fig. 6.	PNEV722xBP1 back10	
Fig. 7.	PNEV722xBP2 Front12	
Fig. 8.	PNEV722xBP2 Back12	
Fig. 9.	Platform-Tools webpage14	
Fig. 10.	Accept&Download14	
Fig. 11.	Unzip platform-tools15	
Fig. 12.	Add adb.exe and fastboot.exe15	
Fig. 13.	win-usb driver webpage16	
Fig. 14.	Accept terms&conditions and download16	
Fig. 15.	Install driver16	
Fig. 16.	Android device in Device Manager17	
Fig. 17.	Device manager17	
Fig. 18.	Browse my computer for driver17	
Fig. 19.	Let me pick from a list of available drivers	
	on my computer18	
Fig. 20.	Android Device18	
Fig. 21.	Android Composite ADB Interface	
Fig. 22.	Click Yes 19	
Fig. 23.	Design resources21	
Fig. 24.	Download prebuild images21	
Fig. 25.	Running the command22	

Tab. 5.	PNEV722xBP1 default jumper settings .	9
Tab. 6.	PNEV722xBP2 default jumper settings .	11
Tab. 7.	J27 shorted pins	11
Tab. 8.	Configuration file and location	46
Tab. 9.	Abbreviations	47
Tab. 10.	Characteristics	49
Tab. 11.	Revision history	51

Fig. 26.	Run flash.bat	22
Fig. 27.	Logcat example	23
Fig. 28.	EMVCo compliance execution: Working	25
Fig. 29.	EMVCo compliance execution: Fail	25
Fig. 30.	EMVCo mode switch application: default	
-	view	26
Fig. 31.	EMVCo mode switch: EMVCo mode is	
J.	selected	27
Fig. 32.	EMVCo mode switch: failure	28
Fig. 33.	Run EMVCo CT test application	29
Fig. 34.	Communication with contact card	29
Fig. 35.	EMVCoAidlHalTransacTest application run	30
Fig. 36.	SmcuSwitchV2_0 application	31
Fig. 37.	DTA	32
Fig. 38.	DTA run button	32
Fig. 39.	Open NfcTdaTest application	33
Fig. 40.	Follow the instructions	34
Fig. 41.	Discover connected smart cards over TDA	35
Fig. 42.	Open a specific SAM slot	36
Fig. 43.	Send APDU to contact card	37
Fig. 44.	Close TDA	38
Fig. 45.	Cockpit location and files	40
Fig. 46.	Running bat files	40
Fig. 47.	Run pnscr	41
Fig. 48.	FW Update logs	43
Fig. 49.	Configuration for FW update	44
-	-	

UG10068

PN7220 – Quick start guide

Contents

1	Introduction	2
2	General description of PN722x	3
3 2 4	Herdware description	4
3.I	Hardware description	4
3.1.1		5 7
3.1.Z		1
3.1.3		9
3.1.4	PNEV/22XBP2	۲۲ ۱۵
3.Z	Soliware description	13
4	Environment setup	14
4.1	kinger and in the set of the set	14
4.Z	Linux environment setup	20
5	ADD commande cynteretter	21
0 7	ADB commands explanation	23 24
1	Test applications	24
7.1	Test applications only for PNEV722XBP1	20 25
7.1.1	ENVCo Compliance Execution	25
7.1.Z	ENVCO Mode Switch	02
7.1.3		29 20
7.1.4	Test englisations only for DNEV/722yPD2	30 24
1.Z 7.0.4	Test applications only for PNEV722XBP2	ו כ זינ
7.2.1	Secure MCU mode switch application	3 I
7.3	DTA emplications	3Z
7.3.1	DTA application	3Z
1.3.Z	Other useful employetions	
7.3.3	NEC Cookrit	39
7.3.4 o	NFC Cockpit	40
0 0 1	Single best	4 2
0.1 0.0	Single host	42
0.Z	Configuration files	45
9 40	Abbroviations and coronyme	40
10	Abbreviations and acronyms	41 40
11	References	40
12	Note about the source code in the	49
10	document	50
11	Bovision history	
14	Legal information	
	Leyai iniorination	52

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

© 2024 NXP B.V.

All rights reserved.

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