UG10068

PN7220 – Quick start guide

Rev. 4.0 — 30 January 2025 **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 same 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.*

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 or direct access to SE.

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) boards and PNEV722xBPx. The first section Section 3.1 "Hardware description" describes the hardware on both variants of i.MX 8M, PNEV722XBP1, and PNEV722XBP2. Section 3.2 "Software description" 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 the i.MX 8M Nano/Mini as DH, but integration is possible with other device hosts as well.

PNEV722xBPx can be connected to an i.MX 8M board with a J43 connector.

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

3.1.1 i.MX 8M Nano and Mini

The 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. The connections below are just for information, the J43 can be directly connected to J1003.

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

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

To connect PNEV722xBP2, existing connections from 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
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
  - SW1102: x x x x x x x x x
- Running Android (1 - 10):
  - SW1101: 0 1 0 0 x x x x x
  - SW1102: 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
  - SW1102: x x x x x x x x 0
- Running Android (1 - 10):
  - SW1101: 0 1 1 0 1 1 0 0 1
  - SW1102: 0 0 0 1 0 1 0 1 0

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

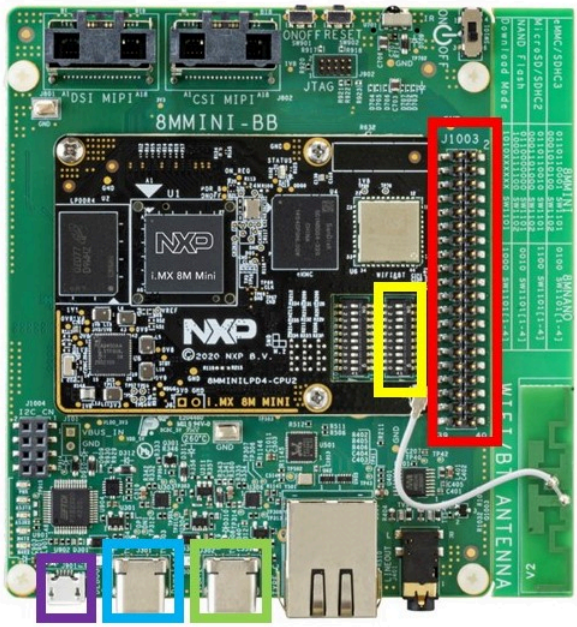

Figure 1. i.MX 8M Mini

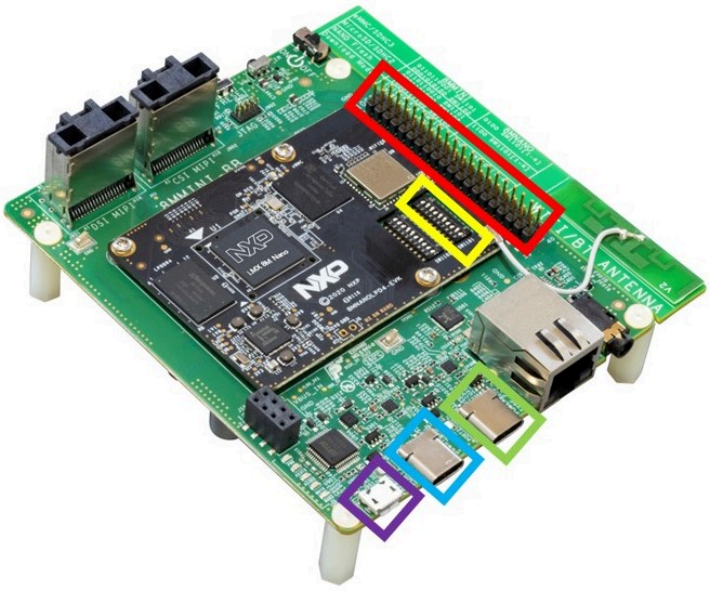

Figure 2. i.MX 8M Nano

Image Figure 3 shows PNEV7220BP1 board connected to i.MX 8M Nano board. The same connection can be used for i.MX 8M Mini.

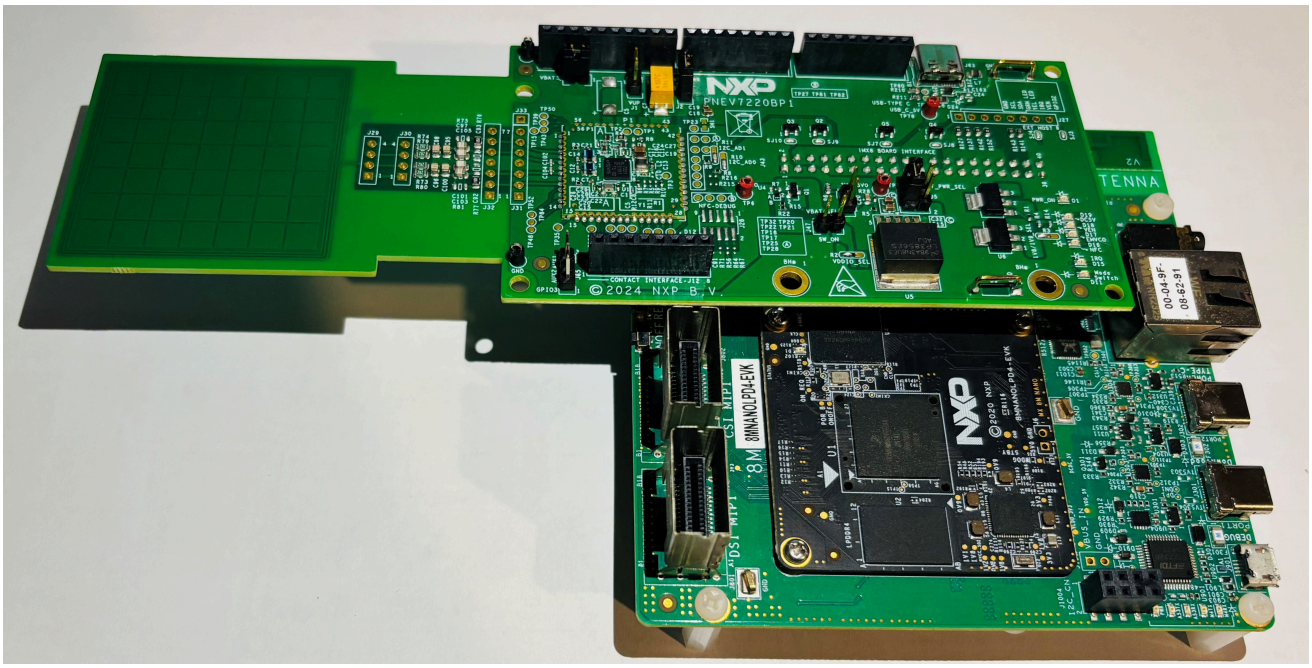

Figure 3. i.MX 8M Nano with PNEV7220BP1

Image Figure 4 shows PNEV7220BP2 board connected to i.MX 8M Nano board. The same connection can be used for i.MX 8M Mini.

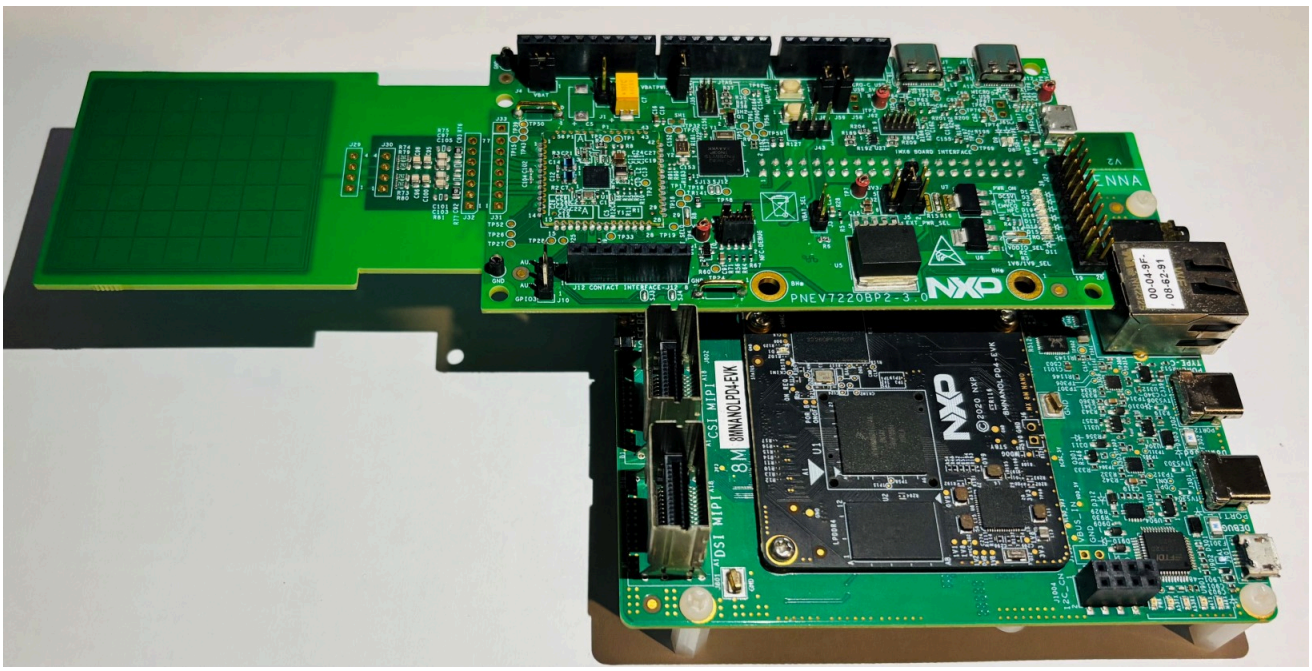

Figure 4. i.MX 8M Nano with PNEV7220BP2

3.1.2 PNEV722xBP1

Figure 5 shows PNEV722xBP1 board. To power up a board, use either USB-C (J63)(External power supply) 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 (External power supply): Pins 1 and 2 must be connected → **Recommended**
2. Powered via DH: Pins 3 and 4 must be connected.

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

Table 3. PNEV722xBP1 default jumper settings

Name	Default setting
VBAT (J4)	CONNECTED
VUP (J1)	NOT CONNECTED
VBATPWR (J2)	CONNECTED
AUX2AUX1 (J65)	NOT CONNECTED
SW_ON	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).

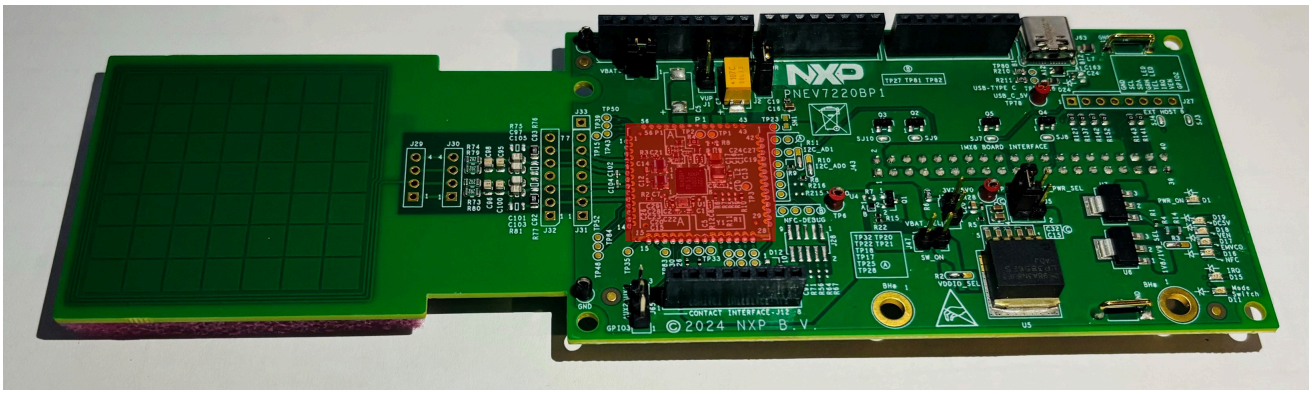

Figure 5. PNEV722xBP1 front

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

Note: *J43 Pin 1 on PNEV722XBP1 board must be connected to J1003 Pin 1.*

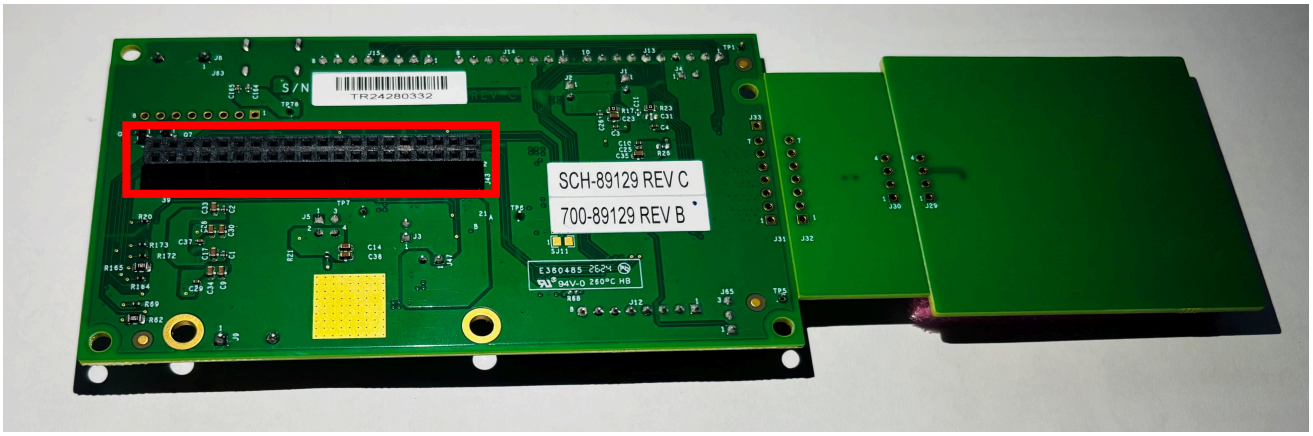

Figure 6. PNEV722xBP1 back

To switch between 1.8 V and 3.3 V for VDDIO_SEL, change the solder jumper to 1-3 (3.3 V) or 1-2 (1.8 V).

For more details, check the data sheet [2].

3.1.3 PNEV722xBP2

Figure 7 shows the PNEV722xBP2 board. To power up the board, use either USB-C (J7)(External power supply) 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 (External power supply): 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 is used to power up the board.

Table 4. PNEV722xBP2 default jumper settings

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

PN7220 IC can be found on the module board (yellow square in Figure 7). Red square on Figure 7 shows K82, which exists only on the PNEV722xBP2 board. To flash it, use a debugger like J-Link which must be connected to J35 (orange square on Figure 7). To run examples on K82, some pins must be shorted on J27. Table 5 shows what must be shorted. If pins are not shorted, the examples on K82 will not work.

Table 5. J27 shorted pins

Pin name	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

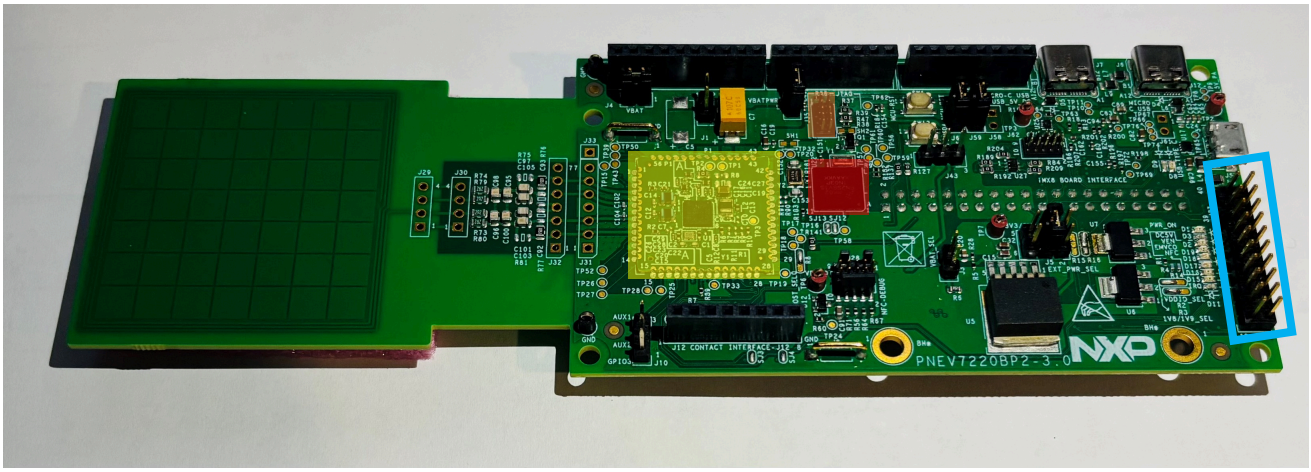

Figure 7. PNEV722xBP2 Front

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

Note: *J43 Pin 1 on PNEV722XBP2 board must be connected to J1003 Pin 1.*

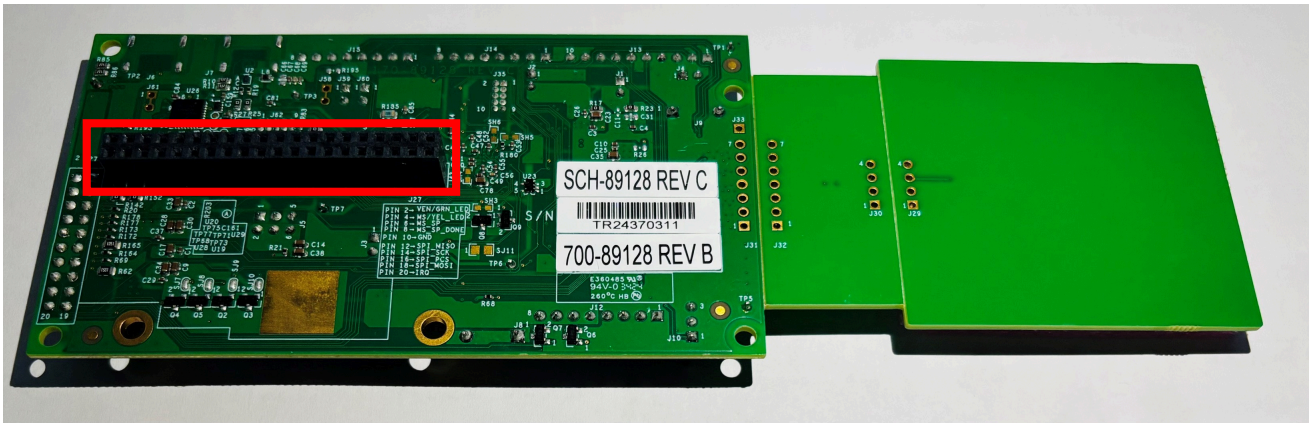

Figure 8. PNEV722xBP2 Back

To switch between 1.8 V and 3.3 V for VDDIO_SEL, change the solder jumper to 1-3 (3.3 V) or 1-2 (1.8 V).

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 the AOSP source code and patches, to ensure 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 [12].

There is also the possibility to get prebuild Android images and applications for an i.MX 8M Nano and Mini. Images can be downloaded from the Prebuild Android images (see Section 5). 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 ([4]) 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 [5] (see Figure 9).

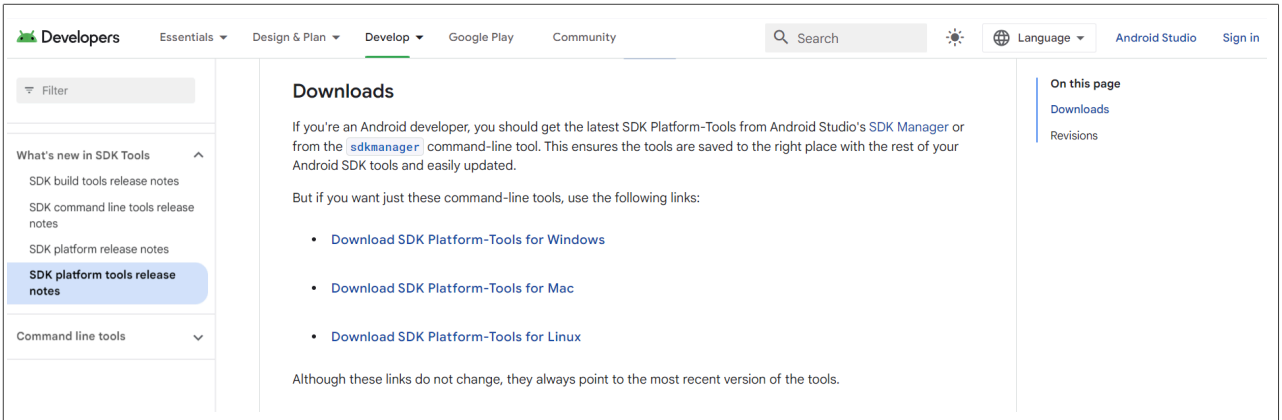

Figure 9. Platform-Tools webpage

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

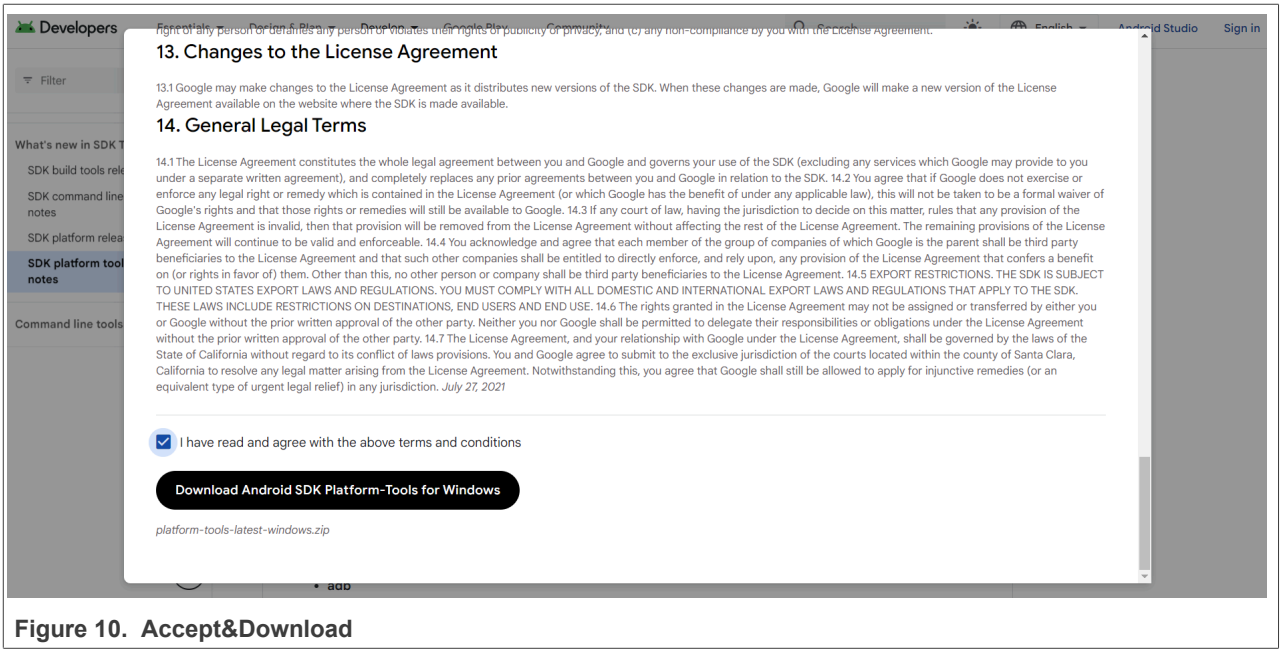

Figure 10. Accept&Download

3. After downloading, extract the folder and add "adb.exe" and "fastboot.exe" to the path (see Figure 11 and Figure 12).

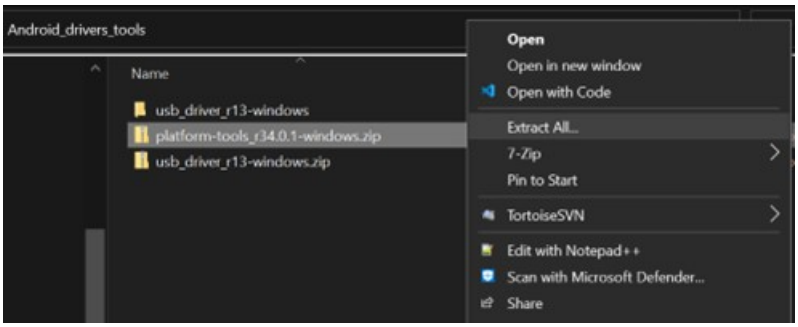

Figure 11. Unzip platform-tools

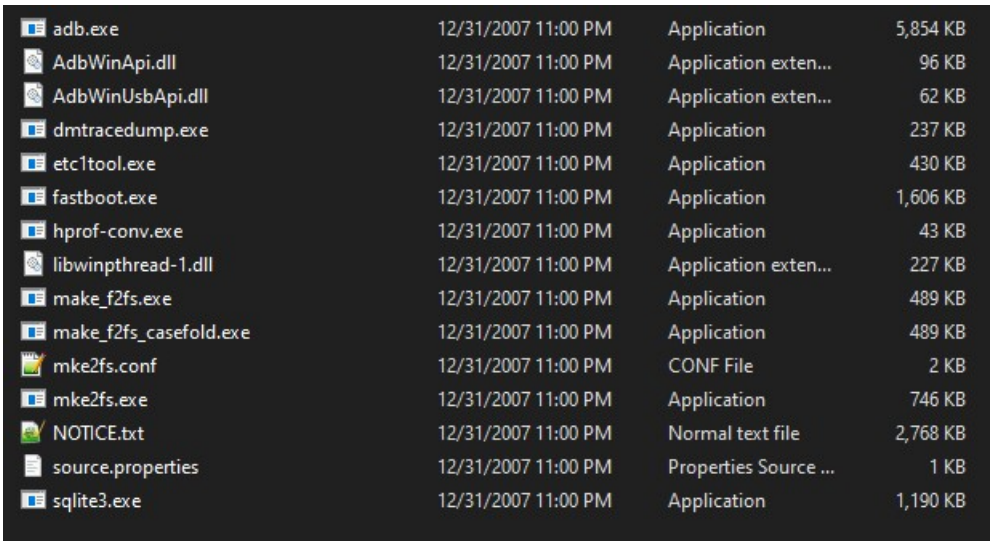

Figure 12. Add adb.exe and fastboot.exe

Instructions for adding tools to the path can be found under this link: [6].

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

1. Choose the driver from the webpage [7] (see Figure 13).

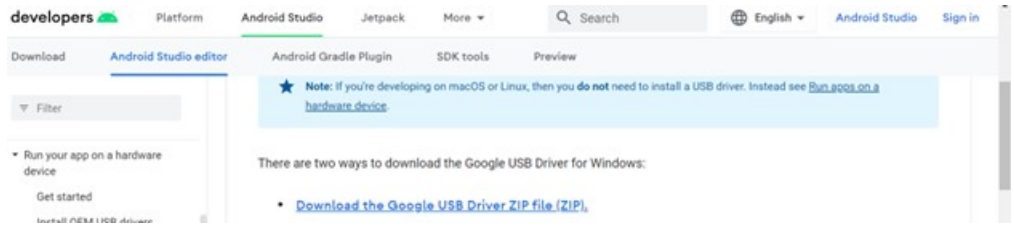

Figure 13. win-usb driver webpage

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

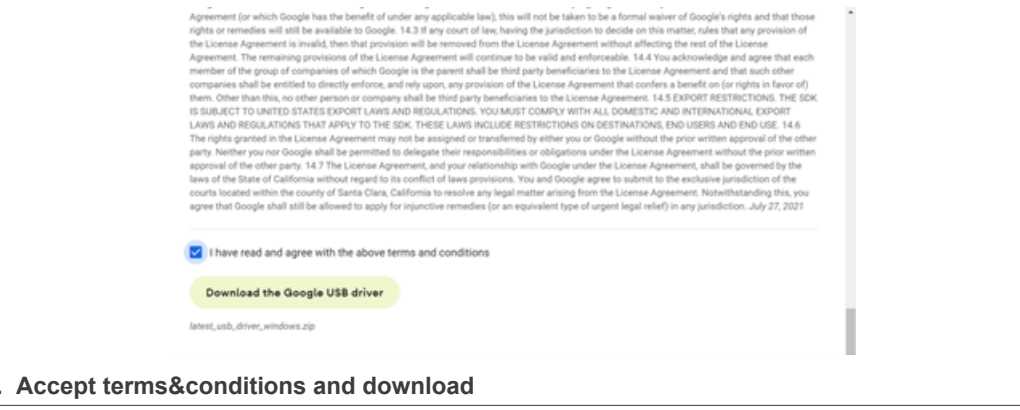

Figure 14. Accept terms&conditions and download

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

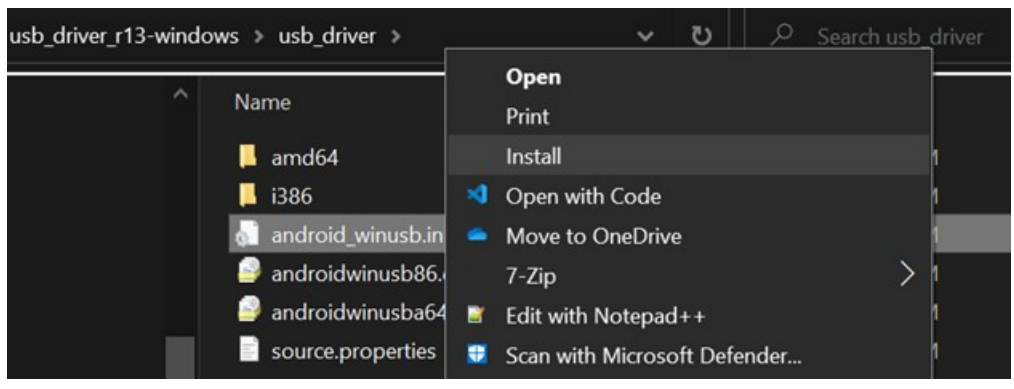

Figure 15. Install driver

After the steps are completed, an i.MX 8M Nano/Mini 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).

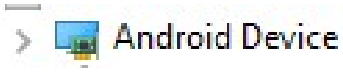

Figure 16. Android device in Device Manager

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.

Figure 17. Device manager

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).

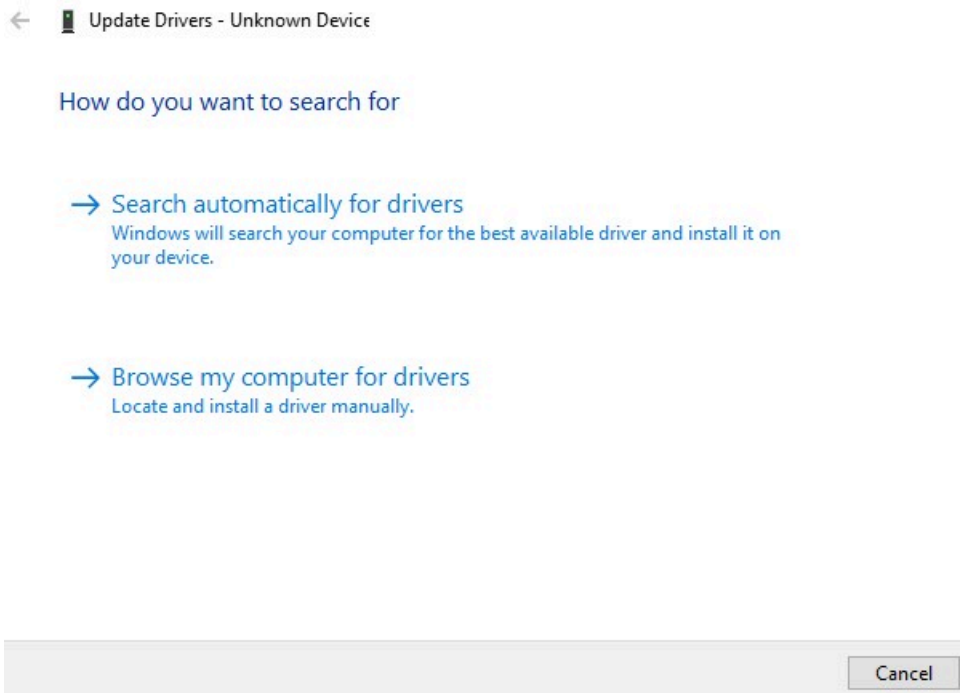

Figure 18. Browse my computer for driver

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

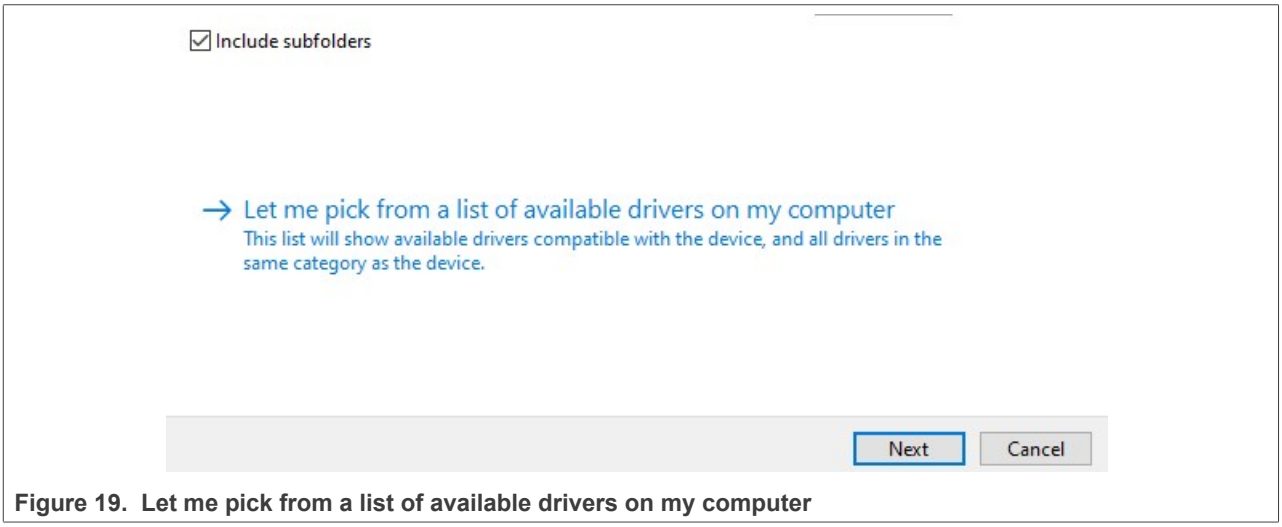

Figure 19. Let me pick from a list of available drivers on my computer

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

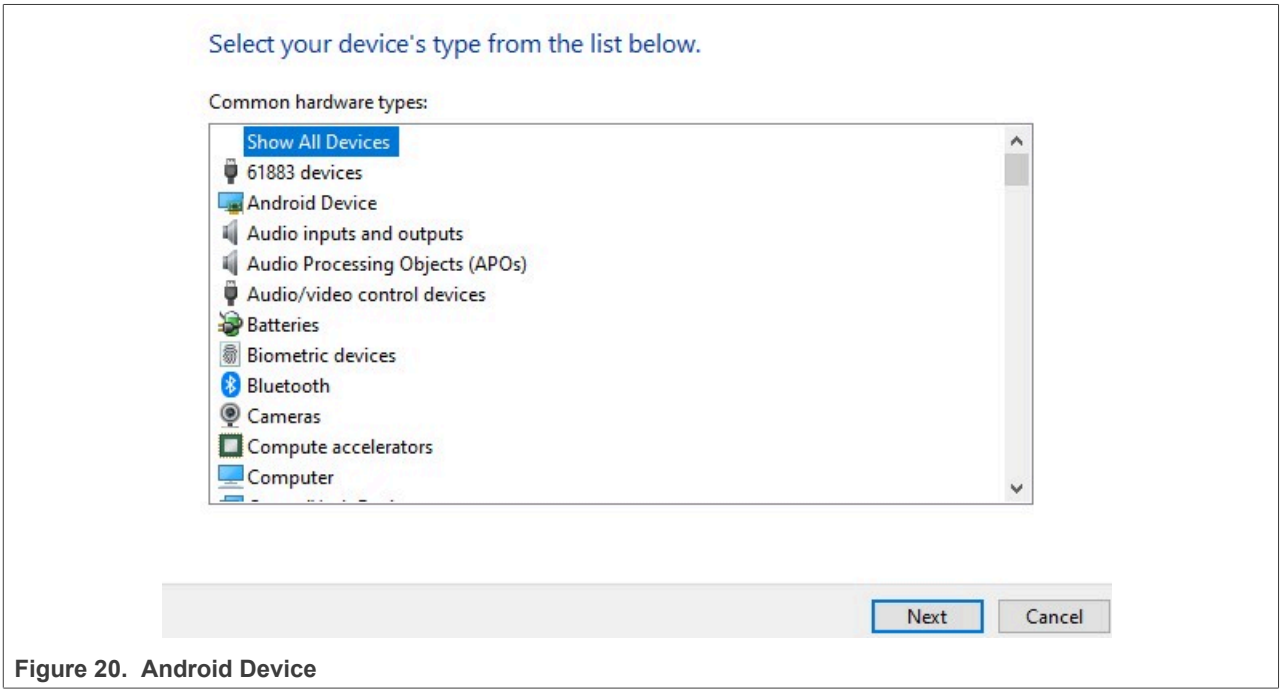

Figure 20. Android Device

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

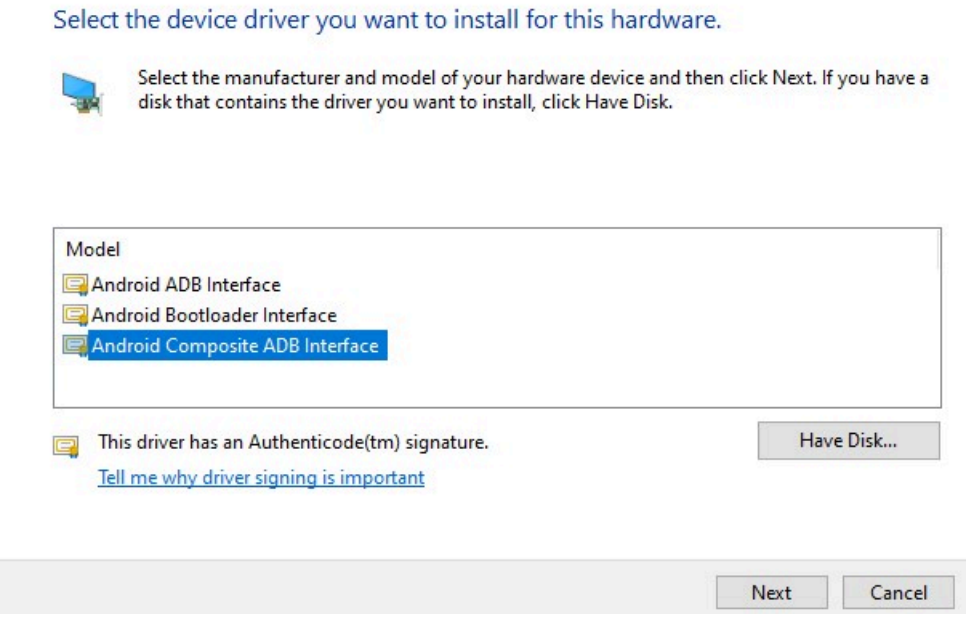

Figure 21. Android Composite ADB Interface

6. Click "Yes" (Figure 22).

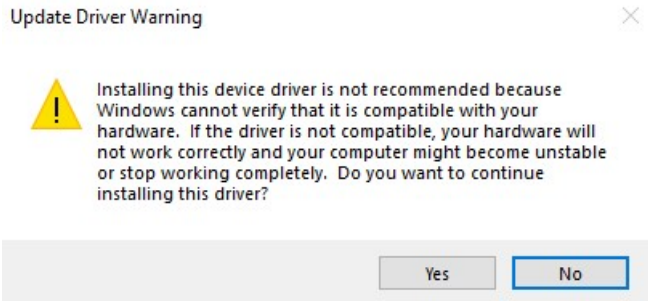

Figure 22. Click Yes

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 [5]
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 the Prebuild Android images. On the webpage, click "Design Resources" (Figure 23)

Figure 23. Design resources

After clicking this button, the page automatically scrolls to the "Design Resources" section. In this section, look for the "Software" subsection. All prebuild images are located here (see Figure 24). It is important to know which evaluation board is used (PNEV722xBP1 (Single Host) or PNEV722xBP2 (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).

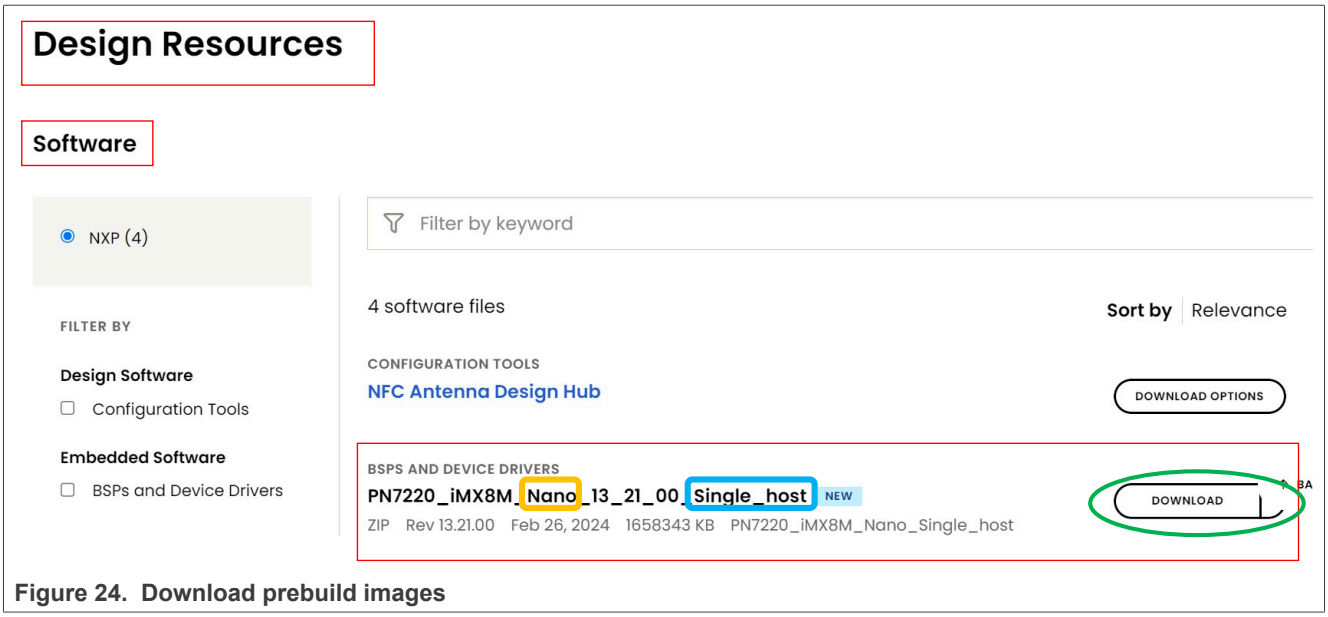

Figure 24. Download prebuild images

When the correct prebuild image is found, click "Download". Sign into your NXP account. Users are 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" or better use PowerShell as Admin since some Windows computers have security rules that block flashing.
3. Before running any command, configuration on the i.MX8 board must be in "Flash Android" mode. Check Section 3.1.1
4. Run the following command:

```
./uuu_imx_android_flash.bat -f imx8mX -a -e -d pn7220
```

Note: *Depend on the i.MX 8M board that is used, imx8m**X** is different:*

- *If using i.MX 8M Nano, the **X** needs to be set as n (imx8m**n**)*
- *If using i.MX 8M Nano, the **X** needs to be set as m (imx8m**m**)*

5. Running the command shows the expected output.

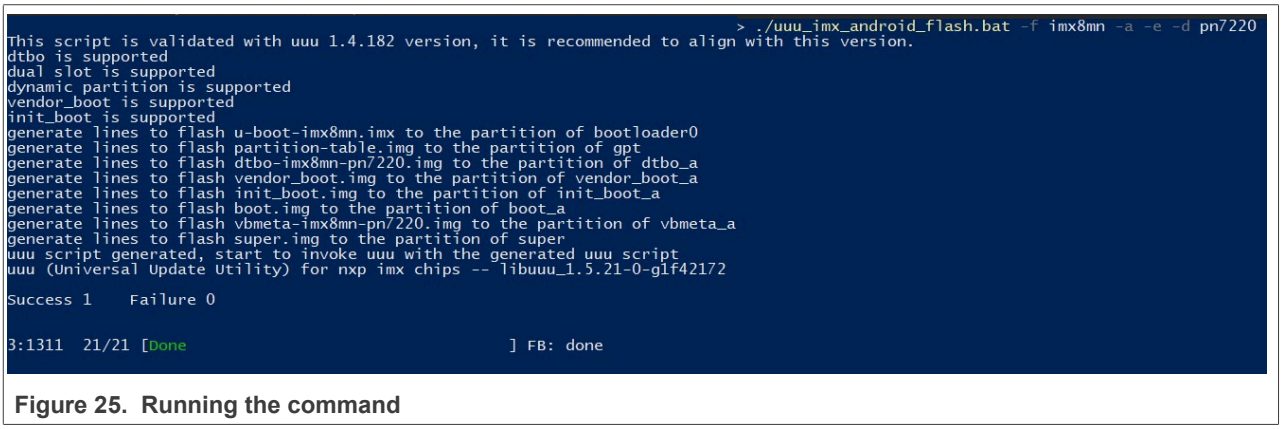

Figure 25. Running the command

6. Put the switch back to "Running Android" mode. Check Section 3.1.1.
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 Extensions folder in the downloaded packet, and run "flash.bat" Figure 26. This .bat file installs additional software on the device.

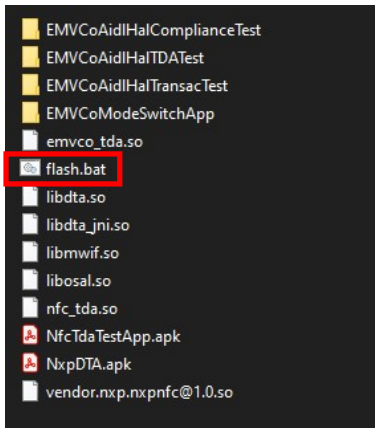

Figure 26. Run flash.bat

Note: *This image is just the example and does not reflect the actual state. The flash.bat is also inside the actual package.*

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
```

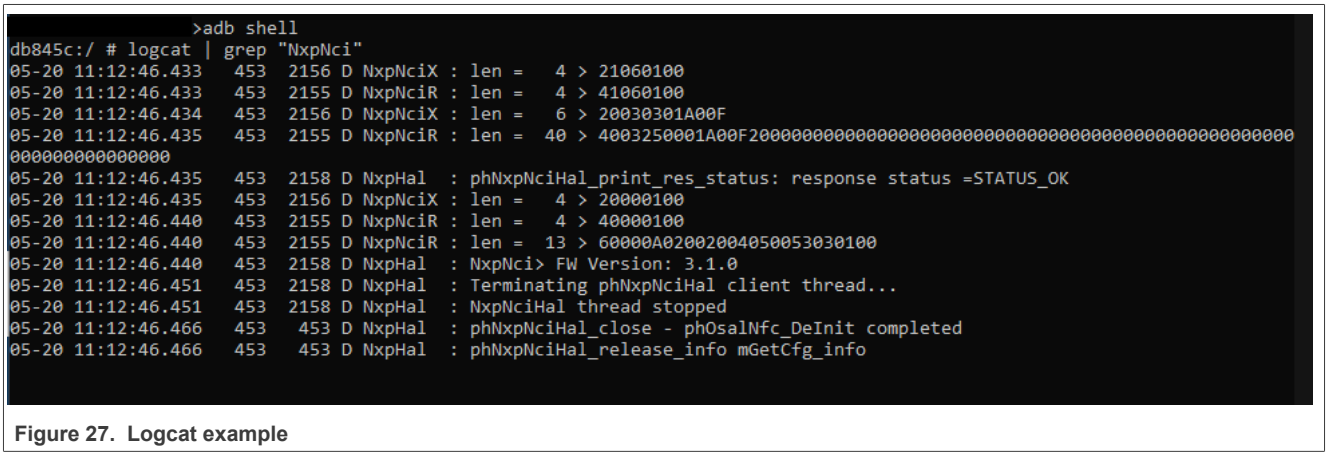

Figure 27. Logcat example

For more information on `adb`, refer to [4] .

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 the Prebuild Android images (see Section 5).

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 [13] and build it.

The script "flash.bat" located in the Extensions folder installs test applications with the following commands:

Note: *While executing flash.sh some java error or file exist can occur. Those errors can be ignored, since after the next reboot all applications will exist and work.*

- `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 EMVCoAidlHalComplianceTest/EMVCoAidlHalComplianceTest system/etc
$adb shell chmod 0777 /system/etc/EMVCoAidlHalComplianceTest
```

To try the application, go to the location where it 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 package Prebuild Android images, found in the Extensions folder.

To install it, run the following commands:

```
$adb root
$adb remount
$adb push EMVCoAidlHalComplianceTest/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
or
$./EMVCoAidlHalComplianceTest type ABF
```

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

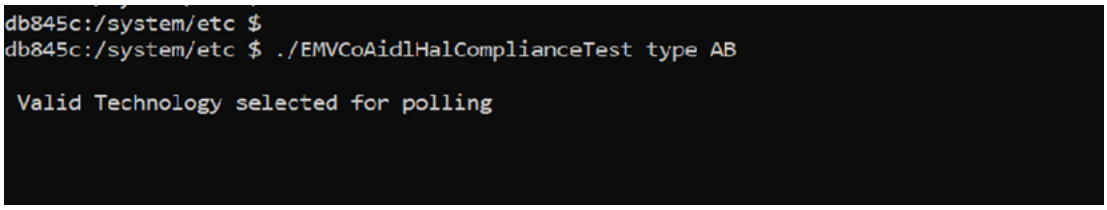

Figure 28. EMVCo compliance execution: Working

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

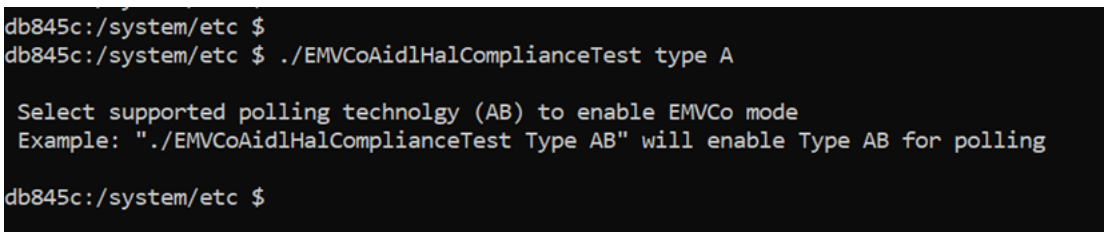

Figure 29. EMVCo compliance execution: Fail

7.1.2 EMVCo Mode Switch

The application is part of the release package Prebuild Android images, found in the Extensions folder (see Section 5).

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, run the following commands:

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

To run Android emulation use Vysor [8] 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.

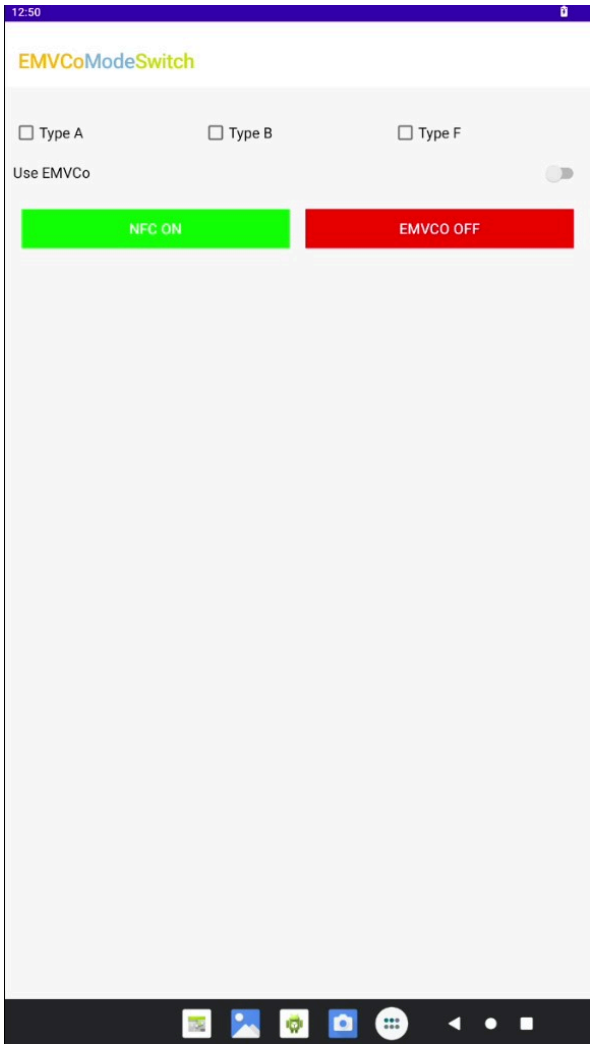

Figure 30. EMVCo mode switch application: default view

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 Figure 31 for reference.

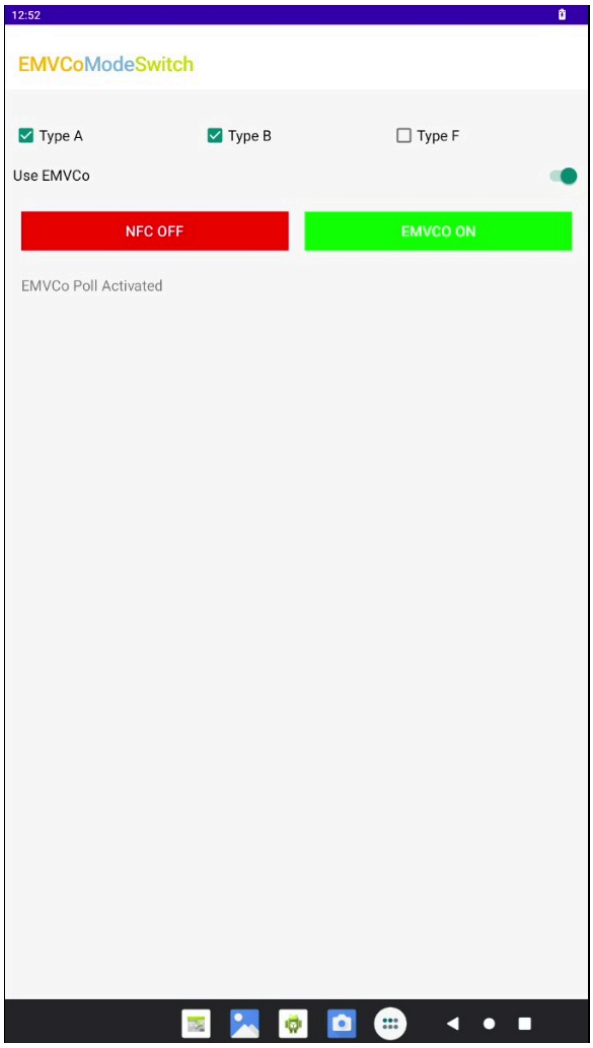

Figure 31. EMVCo mode switch: EMVCo mode is selected

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.

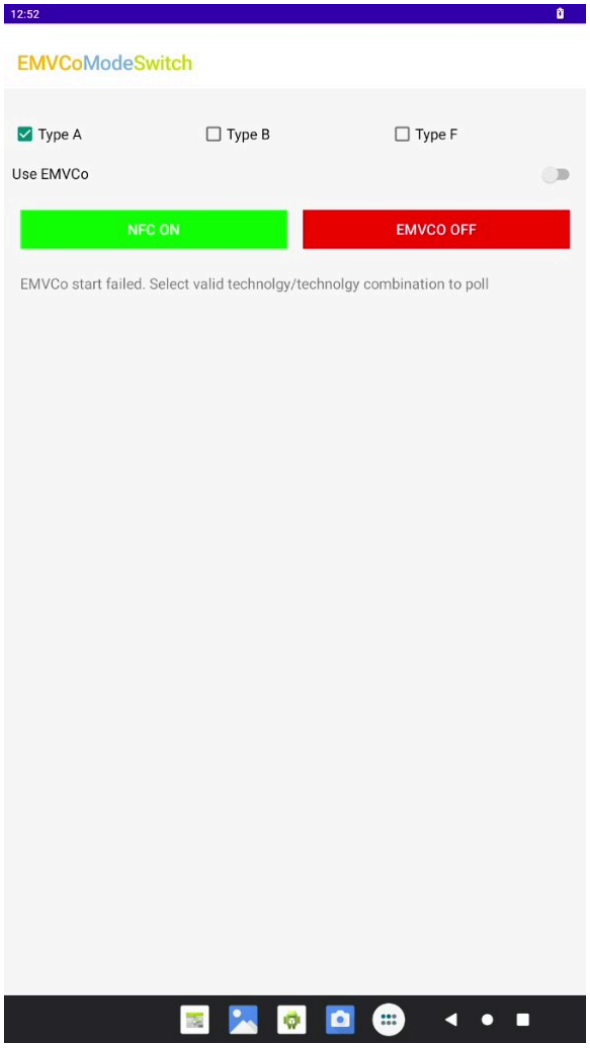

Figure 32. EMVCo mode switch: failure

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 release package Prebuild Android images, found in the Extensions folder (see Section 5).

This application will automatically open and close the TDA connection when a contact card is inserted or removed from the TDA slot.

To install it, run the following commands:

```
$adb root
$adb remount
$adb push EMVCoAidlHalTDATest/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.

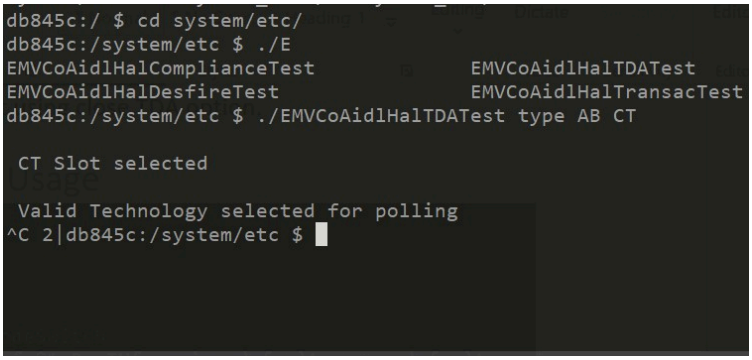

Figure 33. Run EMVCo CT test application

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

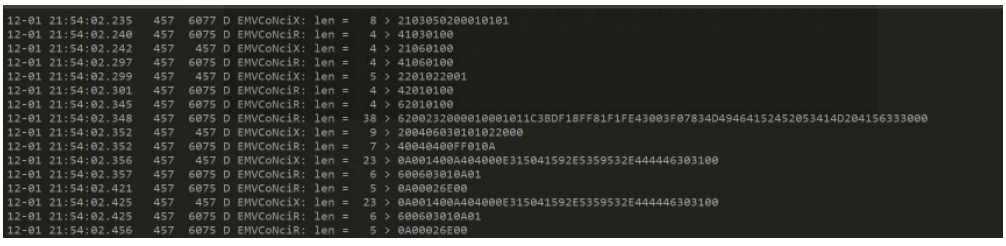

Figure 34. Communication with contact card

7.1.4 EMVCo Transac test

The application is part of the release package Prebuild Android images, found in the Extensions folder (see Section 5).

To install, run the following commands:

```
$adb root
$adb remount
$adb push EMVCoAidlHalTransacTest/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
```

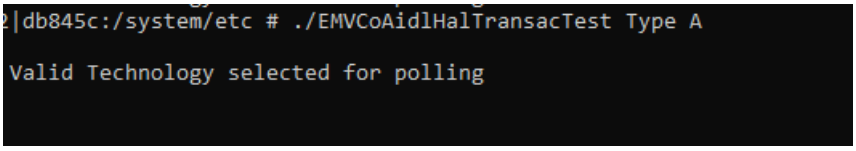

Figure 35. EMVCoAidlHalTransacTest application run

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. Also FW update on PN7220 BP2 is possible only via a secure MCU.

For more information check [14], which explains how to use K82 and where prebuild binaries for K82 are located.

7.2.1 Secure MCU mode switch application

This application is part of the release package that can be found in the Prebuild Android images → Extensions folder (see Section 5). The application is used to switch the host from Android to K82.

To install, run the following commands:

```
$adb root
$adb remount
$adb push SmcuSwitchV2_0/SmcuSwitchV2_0 /system/lib64/
$adb shell
$cd system/lib64/
$chmod +x 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 [14].

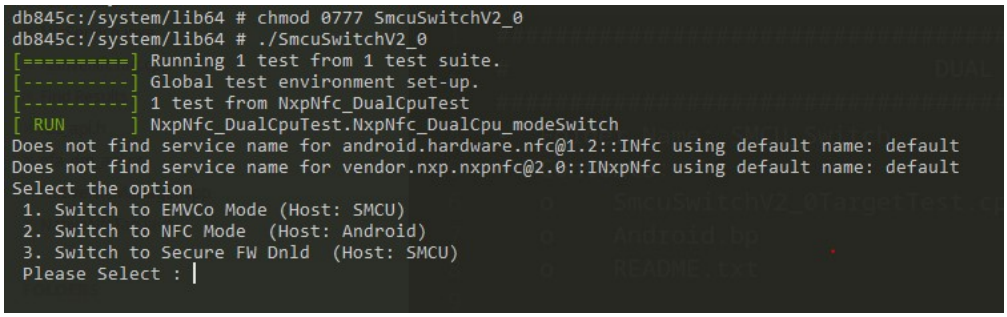

Figure 36. SmcuSwitchV2_0 application

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 in the Prebuild Android images → Extensions folder (see Section 5).

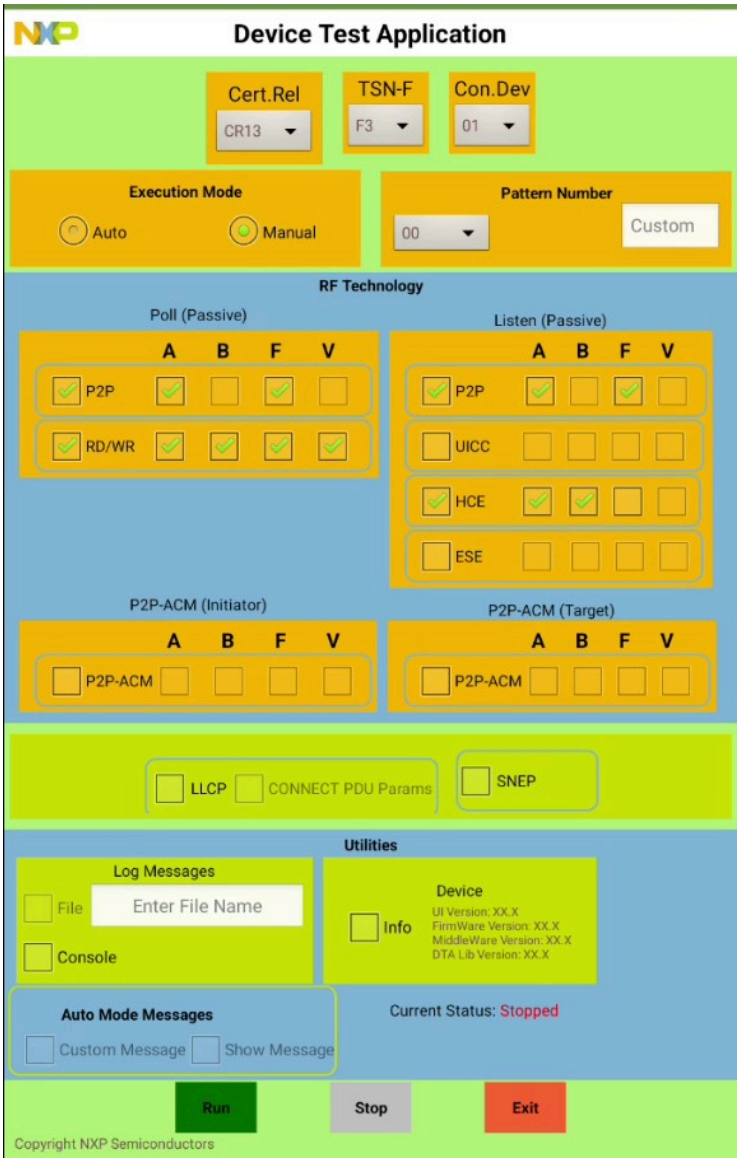

Figure 37. DTA

- 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.
- 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 inside Prebuild Android images in Extensions folder (see Section 5).

In this application open/close connection with contact card, the user needs to perform some manual actions. Follow the instructions below to open/close the connection.

To perform this test, TDA8035 must be connected to the PNEV722xBPx board. Refer to [16] for instructions.

Open terminal (command prompt on Windows) and run:

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

1. Figure 38 shows how to open the application.

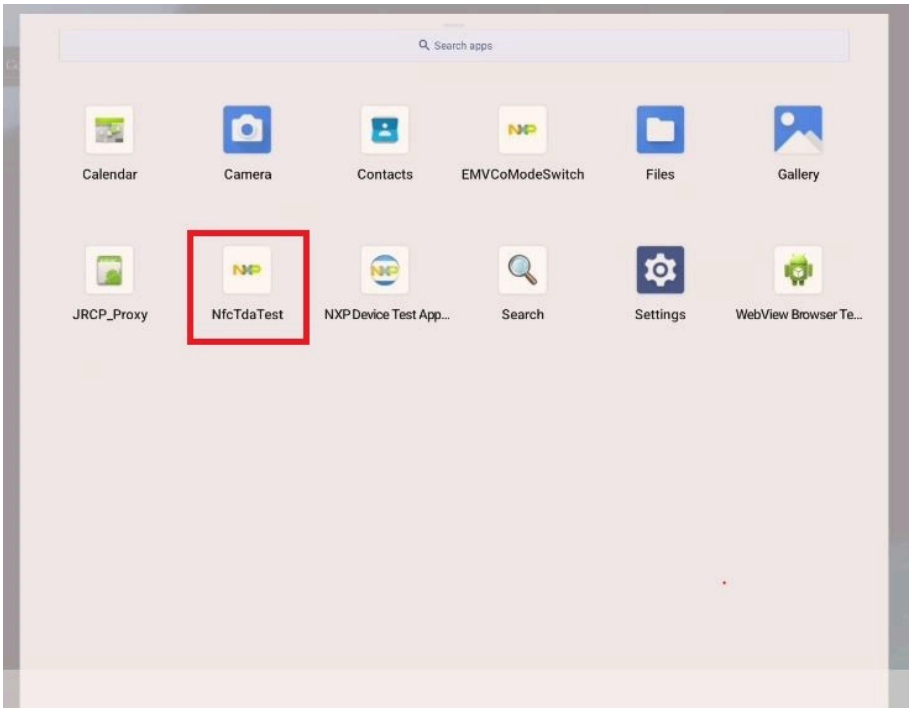

Figure 38. Open NfcTdaTest application

2. 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 Figure 39.

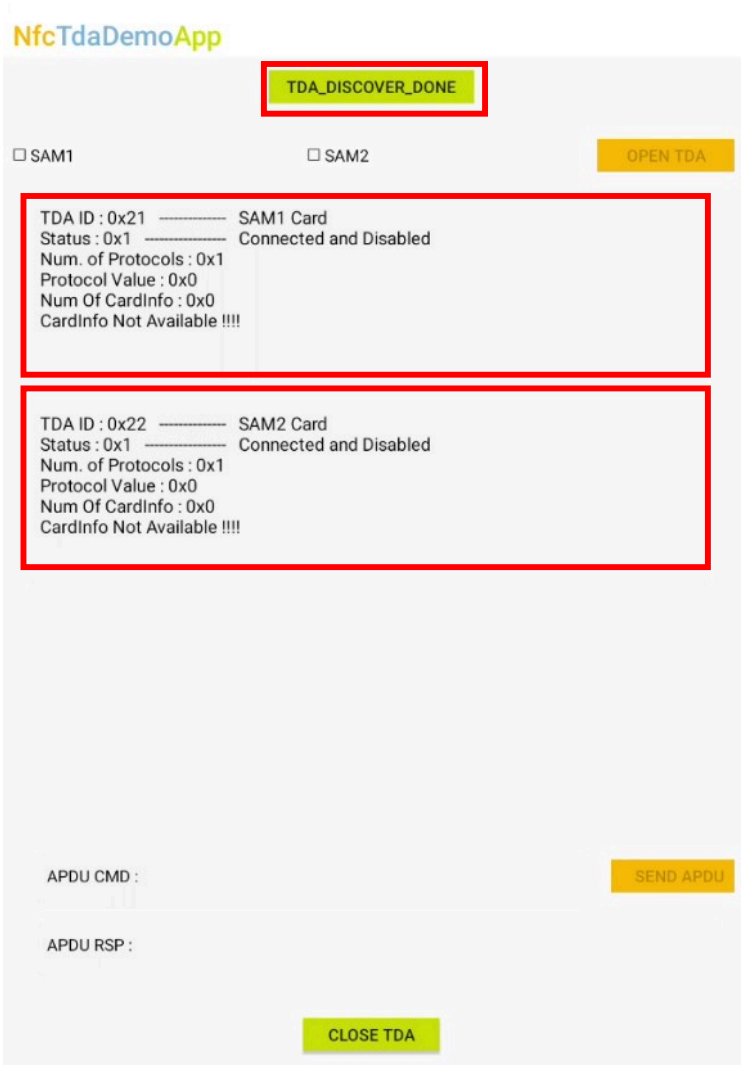

Figure 39. Discover connected smart cards over TDA

3. Select the SAM slot and click "OPEN TDA". See Figure 40.

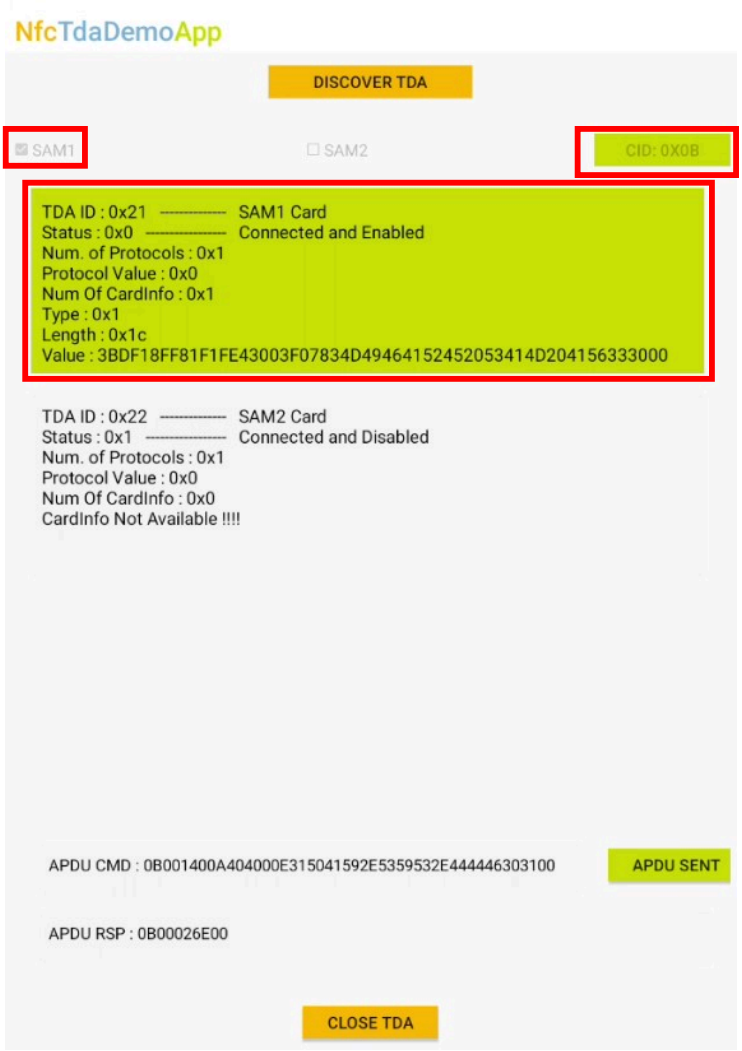

Figure 40. Open a specific SAM slot

4. Send the APDU to Contact card using send "SEND APDU". See Figure 41.

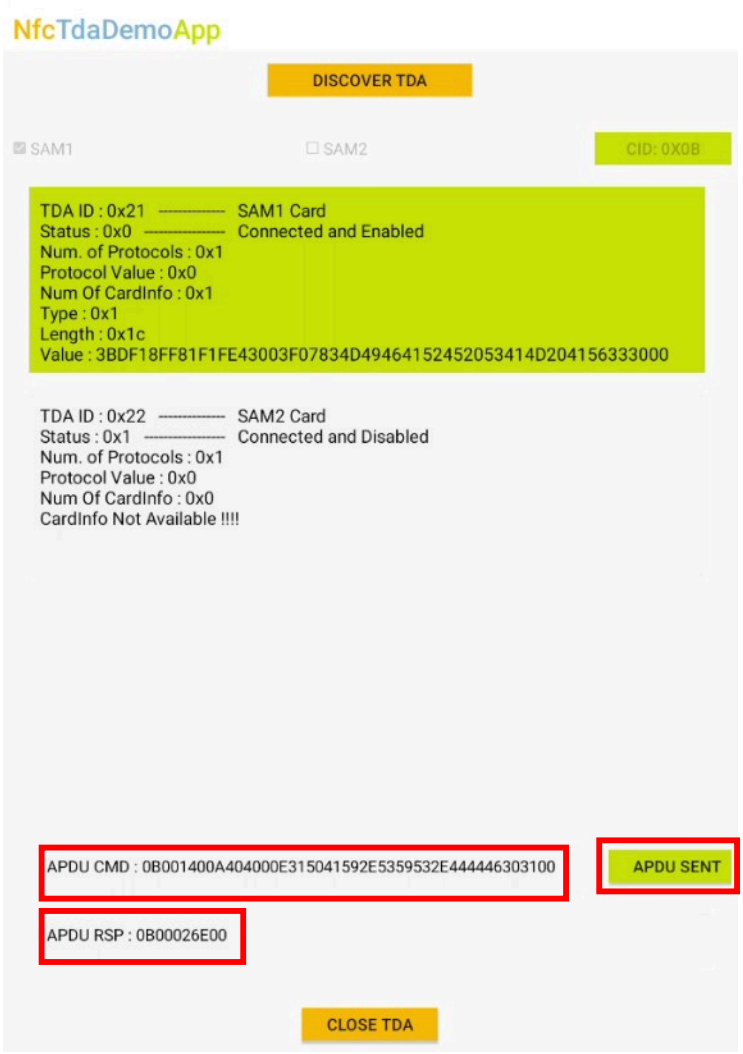

Figure 41. Send APDU to contact card

5. The TDA can be closed with the "CLOSE TDA" button. See Figure 42.

Figure 42. Close TDA

7.3.3 Other useful applications

Other NXP applications can also be used, for example:

- TagInfo Application [9]
- TagWriter Application [9]
- CTS Verifier Application [10]

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

To install the CTS Verifier Application, follow the instructions under [10].

7.3.4 NFC Cockpit

PN722x is the first NXP NCI-based controller that supports the NFC Cockpit tool. The tool can be downloaded from [17].

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_v8.0.0.0\firmware\Secondary_Pn722x\Android*"

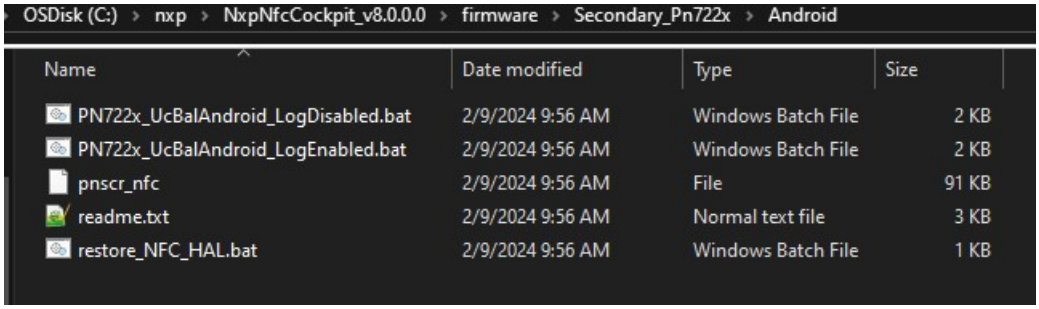

Figure 43. Cockpit location and files

2. Run "PN722x_UcBalAndroid_LogDisabled.bat" or "PN722x_UcBalAndroid_LogEnabled.bat" and follow the instructions in Command Prompt.
   Note: *Review the bat files and change "nfc_nci_nxp_pn72xx.so" and "android.hardware.nfc_pn72xx@1.2-service" if needed. This is needed only if it was changed in the custom Android build.*

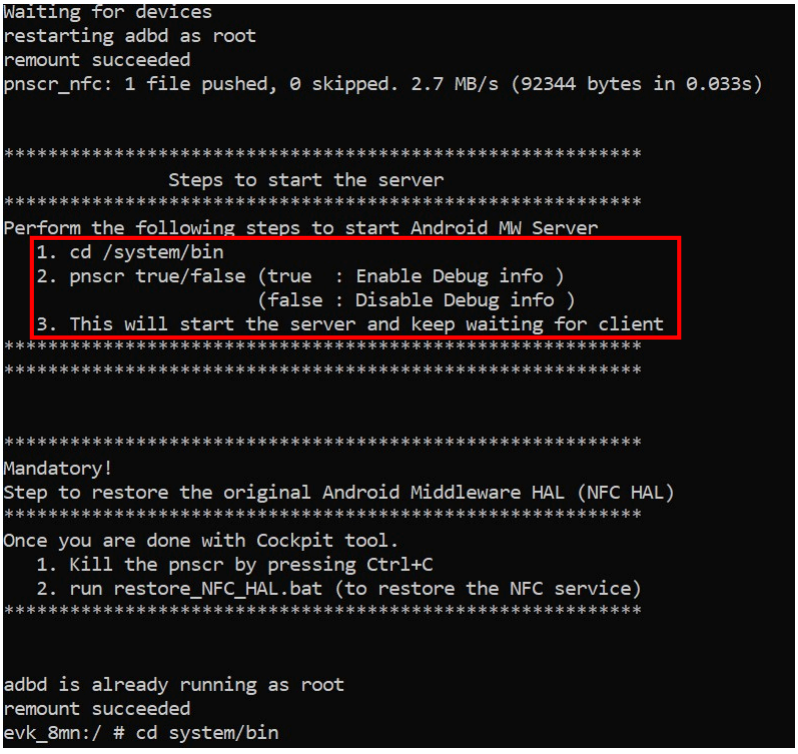

Figure 44. Running bat files

3. Enter:

```
cd system/bin
```

4. Enter the command below (see Figure 45 for expected output):

```
pnscr true //if the nodename is nxpnci
pnscr true -d your_nodename //if node name was changed
```

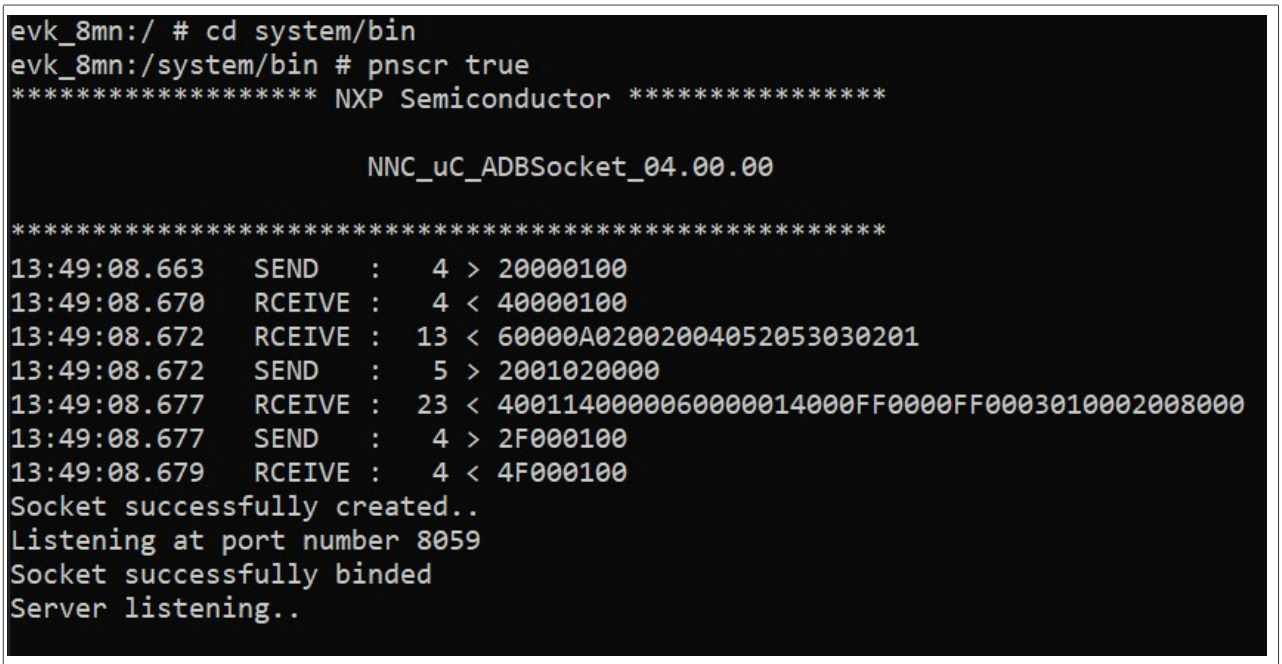

Figure 45. Run pnscr

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
```

Note: *If the bat file has been changed in the customer Android build, review the bat file and change "nfc_nci_nxp_pn72xx.so" if necessary.*

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 to 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 [11].

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
```

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

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.*

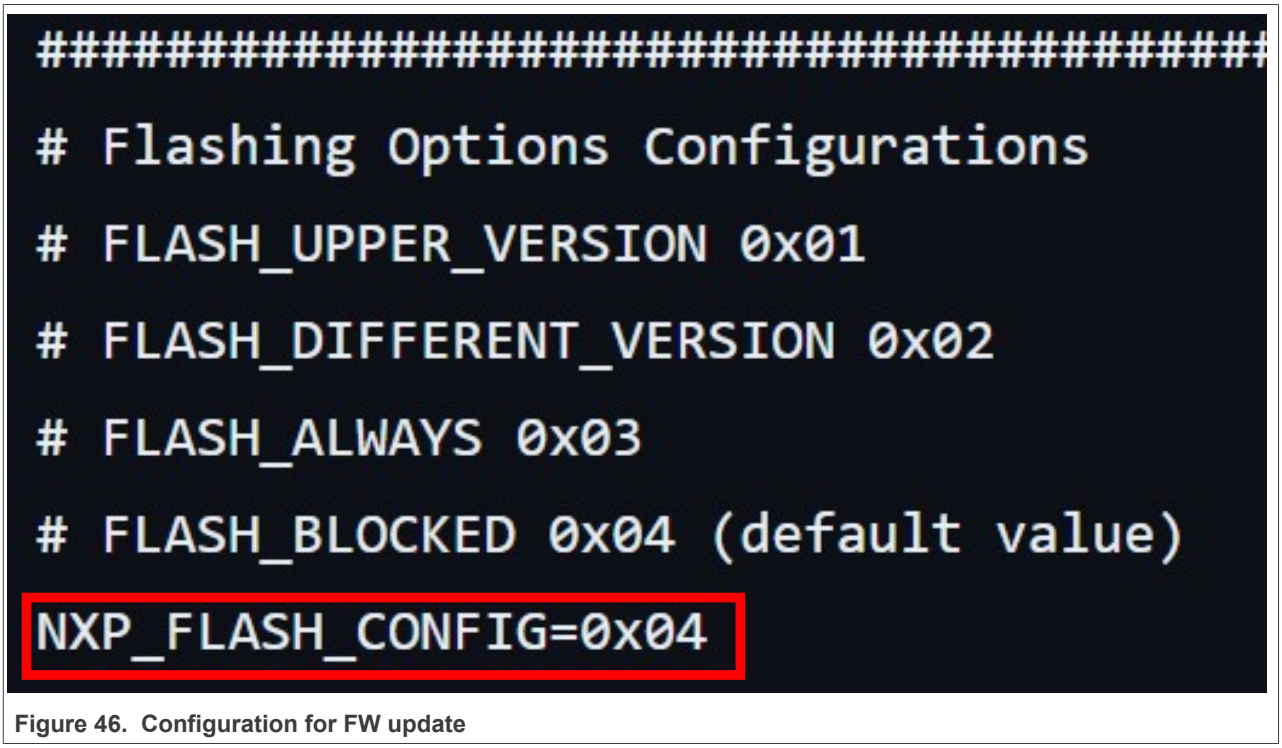

Figure 46. Configuration for FW update

4. 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 shell svc nfc disable
$adb shell svc nfc enable
```

After the reboot is done, the FW update starts.

Checking if FW has been successfully update:

Save the adb logs, as described in Section 6, then open the log file and search for "FW". Figure 47 shows the relevant parts and the current version of FW.

```
2158 D NxpFwDnld: Send Success
2158 D NxpFwDnld: Response timer started
2158 D NxpFwDnld: phNxpNciHal_fw_dnld_chk_integrity_cb - Request Successful
2158 D NxpFwDnld: pn72xx phNxpNciHal_fw_dnld_chk_integrity_cb - Valid Resp Buff!!...
2158 D NxpFwDnld: crc status code area len 0x6
2158 D NxpFwDnld: crc status code data len 0x11
2158 D NxpFwDnld: crc status code area  0xffff803f
 453 D NxpFwDnld: Processing Normal Sequence..
 453 D NxpFwDnld: Initializing Sequence..
 453 D NxpFwDnld: Response Timer Created Successfully
 453 D NxpFwDnld: Inserting FrameId ..
 453 D NxpFwDnld: Frame created successfully
 453 D NxpFwDnld: phDnldNfc_GetDieId Request submitted successfully
2158 D NxpFwDnld: Send Success
2158 D NxpFwDnld: Response timer started
 453 D NxpFwDnld: phNxpNciHal_fw_dnld_complete: Download Status = 0x0
 453 E NxpFwDnld: Fw Download success..
 453 E NxpFwDnld: Returning Download Failed Status to Caller!!
 453 E NxpFwDnld: free library SUCCESS !!
 453 D NxpFwDnld:  phNxpNciHal_fw_dnld_complete : SUCCESS
 453 D NxpFwDnld: fragment len set 22a
 453 D NxpFwDnld: Freeing Mem for Dnld Context..
 453 D NxpHal   : phNxpNciHal_UpdateFwStatus Enter
 453 D NxpHal   : property_set_intf, key[nfc.fw.downloadmode_force], value[0]
 453 D NxpHal   : phNxpNciHal_setSystemProperty : Enter Key = nfc.fw.downloadmode_force, value = 0
2158 D NxpHal   : NxpNci> FW Version: 3.1.0
2158 D NxpHal   : NxpNci> FW Version: 3.1.0
 453 D NxpFwDnld: gphNxpNciHal_fw_IoctlCtx.bClkSrcVal = 0x1
 453 D NxpFwDnld: gphNxpNciHal_fw_IoctlCtx.bClkFreqVal = 0x8
2158 D NxpHal   : NxpNci> FW Version: 3.1.0
2158 D NxpHal   : NxpNci> FW Version: 3.1.0
```

Figure 47. FW Update logs

FW version can be checked anytime, even if the FW update procedure was skipped (see red highlight square in Figure 47).

8.2 Dual host

FW update is executed via a secure MCU. NXP provides the NciLib ([18]) 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 [14].

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 6. 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/

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
```

For more information check [15].

10 Abbreviations and acronyms

Table 7. 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
SE	Secure Element

11 References

[1] Web page – PN7220 – EMV L1 Compliant NFC Controller with NCI Interface Supporting EMV and NFC Forum Applications (link)
[2] Datasheet – PN7220 – EMV L1 Compliant NFC Controller with NCI Interface Supporting EMV and NFC Forum Applications (link)
[3] Web page – Prebuild Android images (link)
[4] Resources – ADB (link)
[5] Resources – Platform-Tools (link)
[6] Webpage – Add Tool Locations to the PATH Environment Variable (link)
[7] Resources – Win-usb driver (link)
[8] Web page – Vysor (link)
[9] Web page – TagInfo and TagWriter applications(link)
[10] Webpage – CTS Verifier (link)
[11] Resources – PN722x FW (link)
[12] Application note – AN13971: Android porting guide (link)
[13] Resources – PN722x MW (link)
[14] Application note – AN14224: How to use PN7220 in Dual-Host mode (link)
[15] Application note – AN14431: PN7160/PN7220 configuration files (link)
[16] Application note – AN14225: How to use PN7220 with contact cards (link)
[17] Web page – NFC Cockpit (link)
[18] 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 8. 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: 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.

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 2023-2025 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.

14 Revision history

Table 9. Revision history

Document ID	Release date	Description
UG10068 v.4.0	30 January 2025	• Section 2 "General description of PN722x" updated. • Section 3 "PNEV722xBPx integration steps" updated, removed all information related to ULP. • Section "I.MX 8M ULP" removed. • Section 3.1 "Hardware description" updated. • Section 3.1.1 "i.MX 8M Nano and Mini" updated. • Section 3.1.2 "PNEV722xBP1" updated. • Section 3.1.3 "PNEV722xBP2" updated. • Section 7.1.1 "EMVCo Compliance Execution" updated. • Section 7.1.3 "EMVCo CT test application" updated. • Section 7.1.4 "EMVCo Transac test" updated. • Section 7.2 "Test applications only for PNEV722xBP2" updated. • Section 7.2.1 "Secure MCU mode switch application" updated. • Section 7.3.2 "NfcTdaTest" updated. • Section 7.3.4 "NFC Cockpit" updated. • Section 8.1 "Single host" updated. • Section 9 "Configuration files" updated.
UG10068 v.3.0	28 May 2024	• Section 7.3.4 "NFC Cockpit" updated. • Section 7.1.3 "EMVCo CT test application" updated. • Section 7.3.2 "NfcTdaTest" updated. • Section 7.3.1 "DTA application" updated.
UG10068 v.2.0	04 April 2024	• Section 3 "PNEV722xBPx integration steps" updated. • Section 5 "Installing prebuild images" updated. • Section 7 "Test applications " added. • Section 8 "Firmware update" added. • Section 9 "Configuration files" added.
UG10068 v.1.0	11 July 2023	• Initial version

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 third party 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.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

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.

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 PSIRT@nxp.com) 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.

Tables

Figures

Contents