AN13134

PN76 family evaluation board quick start guide

Rev. 3.0 — 18 March 2025 Application note

Document information

Information	Content
Keywords	PN76, NFC, open controller, PN7642, LPC55S16, PN76xx
Abstract	This document describes the setup of the development environment and usage of the PN7642 evaluation board.

1 Introduction

This document describes the usage of the PN7642 evaluation board and its development environment, which allows for simple evaluation of the features and functions of the PN7642.

After reading this document, the user should have a basic understanding of the following topics:

1. How to bring up your PNEV7642 development board.
2. Install and set up your development environment (MCUXpresso).
3. Working with the PN76 SDK and its examples.
4. Install and use the NXP NFC Cockpit.

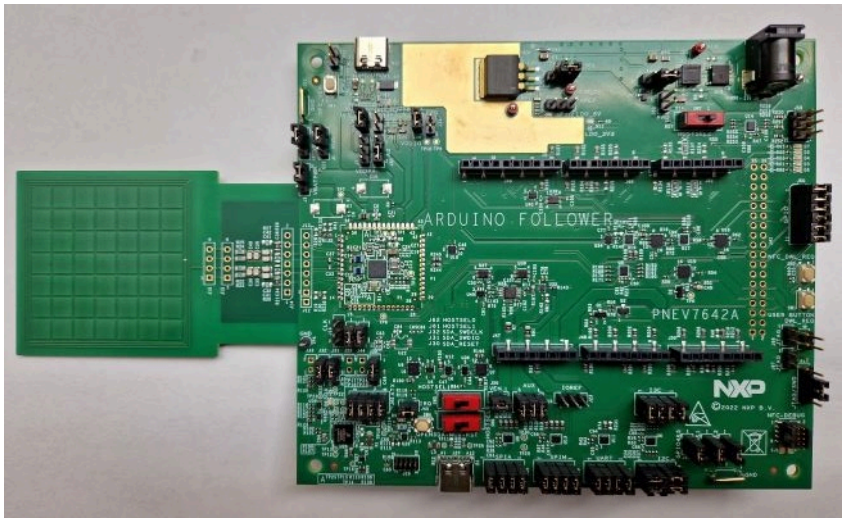

Figure 1. PNEV7642

1.1 Quick map

- There are many documents available for the PN7642, refer to Section 2 "Documentation overview" for an overview.
- For a quick start with the development board, see Section 3 "Quick start with USB as supply"
- Section 4 "System overview" gives a brief introduction to different development setups.
- Section 5 "Firmware overview" provides information about the PN7642 firmware release history.
- Proper configuration of the PN7642 is important, and this configuration involves EEPROM settings. Section 6 "PN7642 registers and EEPROM concept" highlights this.
- The PNEV7642 board is explained in Section 7 "Hardware".
- Software available for the PN7642 is explained in Section 8 "Software".
- To develop software, a development environment is necessary. Setup and usage of which is explained in Section 9 "Development environment".
- Section 10 "NFC Cockpit" is a tool which is recommended to use for analog configuration (Antenna tuning, DPC, etc.).

2 Documentation overview

Below is an overview of documents for the PN7642:

- Data sheet – PN7642
  - This document describes the functionality and electrical specification of the NFC open controller PN7642.
- AN13467 – PN76 family antenna design guide
  - This document describes the NFC antenna design and tuning related to the PN76 family. This includes the dynamic power control 2.0 functionality. It gives some layout recommendations and some guidelines on how to adjust (“calibrate”) the DPC.
- AN13925 – PN76 EEPROM configuration handling
  - This document explains how to enable the OpenSDA interface for debugging on the PN76 development boards.
- AN13996 – How to use the low-power features of the PN76 family NFC controller
  - This application note explains and shows how to use different low-power modes on the PN76 family NFC controllers using the NFC low-power modes example of the SDK.
- AN14060 – How to use the PN76 family cryptographic features (
  - Within this application note it is shown how to use the internally stored keys, in the PN7642 secure key store, with the mbedTLS APIs.
- AN13720 – PN7642 Secure Key Mode demo application
  - This document describes a demo application to showcase the Secure Key Mode commands of the PN7642 front-end controller. The scope of this document is to describe the way of working with the demo application on PN7642 FAMA board.
- PN7642 NFC controller user API documentation
  - This documentation holds all information about the PN7642 firmware APIs. It is part of the MCUXpresso SDK and found in the docs folder. Unpack it and open the "index.html" to read it in your browser. It is generated out of the inline comments with DoxyGen.
- AN13936 – PN7642 frequently asked questions
  - A FAQ document with common questions and their answers. This is an active and frequently updated document and worth a look if found in a dead end.
- AN14540 – Firmware update on PN7642
  - This application note explains the different possibilities how the NXP firmware and user application can be updated on the PN7642.
- AN14518 – Crystal Oscillator Design Guide
  - This document covers the design process of the XTAL oscillator in NFC Reader circuits. In this context, design means the selection of the correct XTAL unit and its implementation into the customer design.

3 Quick start with USB as supply

Follow the instructions below to bring-up the PNEV7642 development board and its environment. For further details about the development board, the development environment, the product support package, and other related topics, refer to the following sections of this document.

1. Unpack your PNEV7642A Rev-B development board.
2. Set jumper "J3 - 5V_SEL" to 3-2 to use USB 5V as supply:

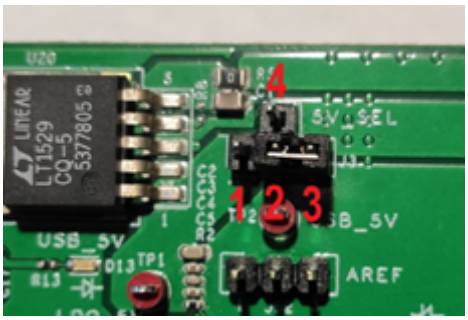

Figure 2. Supply selection

   See Section 7.1.1.1 "Power supply selection".
3. Plug in a USB-C cable to J5 (top USB-C connector) and to the PC (active USB hub recommended due to the high in-rush current).
   See Section 4.1 "PN76 and external debugger".
4. Download, install, and open the latest MCUXpresso IDE.
   a. Create a new workspace for the PN7642 examples.
   See Section 9.1.1 "MCUXpresso IDE".
5. Download the PN7642 MCUXpresso SDK from the PN7642 webpage [10]. By default, the PNEV7642 comes with firmware version v01.00. If users do not want to update the firmware first, download *SDK v02.12.01*.
   See Section 5 "Firmware overview" for firmware and SDK compatibility. Refer to [8] for instructions on updating the PN7642 firmware.
6. Install the SDK by dragging and dropping the .zip file in the "Installed SDKs" tab, found at the bottom of the MCUXpresso:

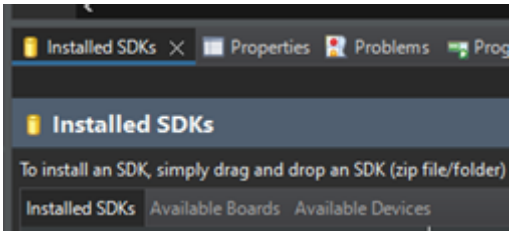

Figure 3. Installed SDKs tab

   See Section 9.1.1.2 "Install SDK".

7. Click "Import SDK example(s)" in the "Quick Start Panel", to import the first SDK example.

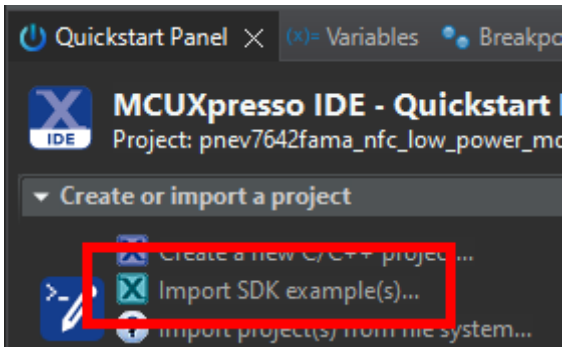

Figure 4. Quickstart Panel - Import SDK examples

   a. Import "nfc_examples" → "NfcrdlibEx1_DiscoveryLoop_Freertos" and check "Semihost" in the "Project Options":

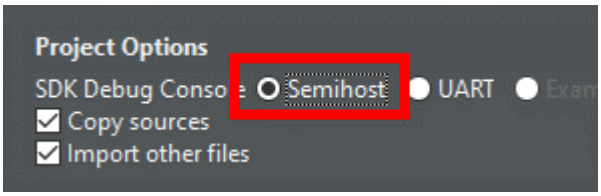

Figure 5. Project Config - Semihost

   b. Click "Finish" to start importing the example into your project workspace.

   See Section 9.1.2 "Import SDK example".

8. Connect a debugger (NXP MCU-Link, LPC-Link2, or SEGGER J-Link) to jumper "J21 - NFC_DEBUG" (bottom-right corner of the development board).
   See Section 7.1.2.1 "JTAG/SWD interface".
9. Click the blue "Bug"-icon in MCUXpresso to compile and start debugging the example:

Figure 6. MCUXpresso start debugging

   a. A pop-up window should appear to choose your debugger. Click the debugger of choice, which is connected to the PN7642, and click "Ok".

   See Section 9.1.3.2 "Debug example".

10. The program should be automatically halted at the void main(). Click "Resume" or press "F8" to let the example run:

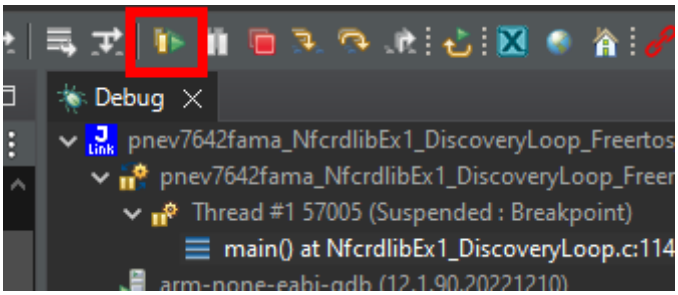

Figure 7. MCUXpresso - Resume button

11. Place an NFC-Tag on the antenna of the PNEV7642A development board. The example prints the information of the detected tag:

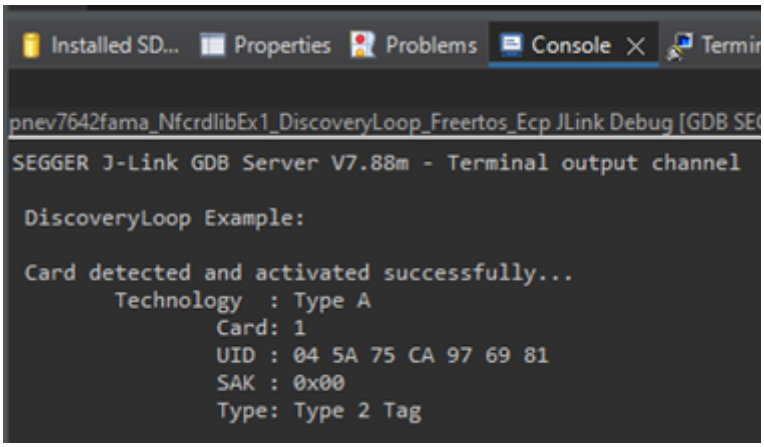

Figure 8. MCUXpresso - Console output of ex1

4 System overview

There are multiple setups to get started with the development board. Choose the setup which best fits your purpose, way of working, and available hardware the best.

- Section 4.1 "PN76 and external debugger"
- Section 4.2 "PN76 and OpenSDA"
- Section 4.3 "PN76, LPC55, and external debugger"

4.1 PN76 and external debugger

The following figure indicates how an evaluation setup with PNEV7642A and an external programmer can be set-up.

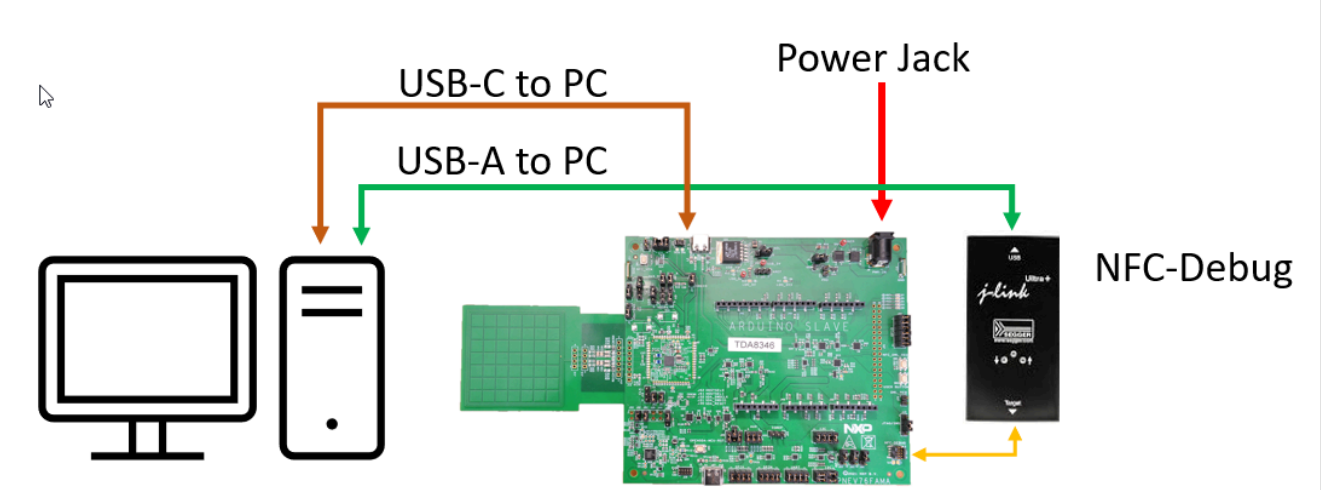

Figure 9. System overview - PN76 and external debugger

PNEV7642A connections:

- J5 "Top USB-C" connected with a USB-C cable to the developers PC for communication (for example, VCOM for NFC Cockpit).
- J9 "Power Jack" connected to a power supply (5V, min. 1A).
  - If you power it directly via USB "J5" and not with the power jack, an active USB hub is recommended, due to the high in-rush current.
- J21 "NFC-Debug" to connect a Debug Probe like a SEGGER J-Link or MCU-Link.

Table 1. Boards and utility overview

What	Why
PNEV7642A Board	The PNEV7642A is the evaluation board of the PN7642 chipset. This whole document describes how to configure and program it.
Debug Probe	In the picture above, a SEGGER J-Link is shown as an external debugger. If you want to flash an application and debug it onto the PN7642, you need an external debug probe. This can be a MCU-Link or a SEGGER J-Link.
Personal Computer (Host)	A Windows, Linux, or MacOS computer is necessary to run the IDE for developing applications for the PN7642 chip. Currently only Windows is fully supported and described within this application note.

4.2 PN76 and OpenSDA

OpenSDA is **an open-standard serial and debug adapter**. It bridges serial and debug communications between a USB host and an embedded target processor. The following figure explains how an evaluation setup with PNEV7642A and the onboard OpenSDA can look like and how to connect it.

By default, the OpenSDA option on the PNEV7642A Rev-B development board is disabled. How to use OpenSDA is described in [11].

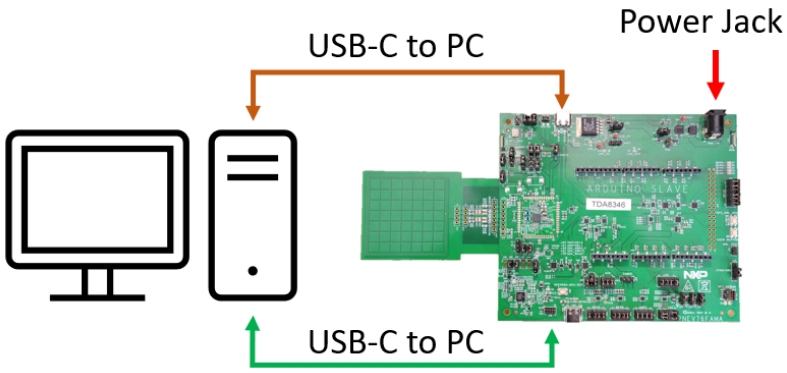

Figure 10. PN76 and OpenSDA

PNEV7642A connections:

- J5 "Top USB-C" connected with a USB-C cable to the developers PC for communication (for example, VCOM for NFC Cockpit).
- J9 "Power Jack" connected to a proper power supply to supply the PNEV7642A board.
- J27 "Bottom USB-C" connected for the OpenSDA debug probe.

Table 2. Boards and utility overview

What	Why
PNEV7642A Board	The PNEV7642A is the evaluation board of the PN7642 chipset. This whole document describes how to configure and program it.
Personal Computer (Host)	A Windows, Linux, or MacOS computer is necessary to run the IDE for developing applications for the PN7642 chip. Currently only Windows is fully supported and described within this application note.

4.3 PN76, LPC55, and external debugger

The following indicates how an evaluation setup with PNEV7642A, LPC55S16-EVK, and an external programmer can be set-up.

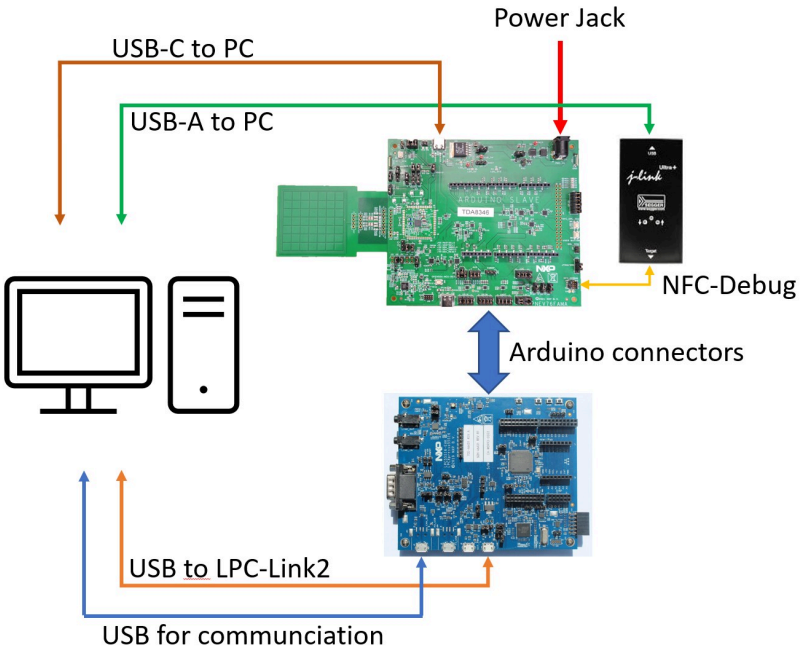

Figure 11. System overview

PNEV7642A connections:

- J5 "Top USB-C" connected with a USB-C cable to the developers PC for communication (for example, VCOM for NFC Cockpit).
- J9 "Power Jack" connected to a proper power supply to supply the PNEV7642A board.
- J21 "NFC-Debug" to connect a Debug Probe, SEGGER J-Link or LPC-Link2, to program the PN76.

LPC55S16-EVK

- J4 "Main USB" connected to the PC for communication with the PC and/or as power supply for the LPC55S16-EVK board.
- J1 "LINK2" connected to the PC and shown as an LPC-Link2 debug probe. Used to program the LPC55S16.

Table 3. Boards and utility overview

What	Why
PNEV7642A Board	The PNEV7642A is the evaluation board of the PN7642 chipset. This whole document describes how to configure and program it.
LPC55S16-EVK	The LPC55S16-EVK is an optional board. It is only necessary for some of the provided examples like HIF-based examples. The PNEV7642A is fully operational without this board. Within the provided software package, you find "host software". This software is dedicated to run on this LPC55S16 board to complement another example running on the PNEV7642A Rev-B development board. One of the major reasons to use a LPC55S16 besides showcasing the HIF encryption is to update the PN7642 firmware. This currently cannot be done without the LPC55 board.
Debug Probe	In the picture above, a SEGGER J-Link is shown as an external debugger. If you want to flash an application and debug it onto the PN7642, you need an external debug probe. This can either be a LPC-Link2 or a SEGGER J-Link.
Personal Computer (Host)	A Windows, Linux, or MacOS computer is necessary to run the IDE for developing applications for the PN7642 chip. Currently only Windows is fully supported and described within this application note.

5 Firmware overview

The PN76 firmware and tool compatibility are listed below in Table 4.

Firmware version

A firmware version is always 2 bytes: XX.YY

Where '*XX*' is the major version and '*YY*' is the minor version. A firmware version written as "2.1" is equal to "02.01", "1.0" is equal to "01.00" and so on.

It is not possible to downgrade a major version. A chip with firmware version 02.xx cannot be downgraded to firmware version 01.xx. A downgrade is only possible within minor version: 02.02 → 02.01; 01.0A → 01.00; and so on.

A firmware version with “F” in its minor version is the same as without the “F” except it does not update the RF and protocol settings. Firmware version "01.00" and 01.F0" are equal from a functional point of view, but one overwrites the settings while the other one, with “F”, will not modify RF settings.

SDK version

An SDK is bound to a certain firmware version. Make sure to use the right SDK version for the firmware version. An SDK upgrade is necessary if a firmware upgrade is made. Or the firmware is upgraded to the firmware version within the SDK.

This guarantees that the compiled application is running on the target firmware version, see Table 4.

NFC Cockpit version

The NFC Cockpit application binary (.bin), within the installation folder of the NFC Cockpit, is compiled for a certain firmware version.

Make sure to flash the correct NFC Cockpit application according to your firmware version.

Firmware backward compatibility

A firmwares backwards compatibility is not guaranteed. While a backward compatibility within minor versions is given (02.02 → 02.00), this is unlikely for major version changes (02.00 → 01.00).

Before performing a firmware update, check the release notes.

Compatibility table

Table 4. Firmware compatibility

FW version	SDK version	Comment
01.00 / 01.F0	2.12.1	Standard firmware on C100[1] ICs. DWL_REQ pin necessary to enter into the bootloader.
02.00 / 02.F0	2.12.3	Standard firmware on C101 ICs. DWL_REQ pin not necessary. Pin-less download active as default.
02.01 / 02.F1	2.12.4	Firmware officially withdrawn due to backward compatibility issues.
02.02 / 02.F2	2.12.5	Minor bug fixes. Backward-compatible to FW v02.00.
	2.15.000	New SDK structure to support VSC. Same content as v2.12.5
02.03 / 02.F3	2.15.002	See release notes of the SDK.
02.05 / 02.F5	2.15.003	See release notes of the SDK.

[1] C100/C101 being the part numbers of the PN7642.

Firmware release notes

See RN00257 for the latest PN7642 firmware release notes.

5.1 Firmware update

How to perform a firmware update of the NXP firmware and user application space is explained in [8].

6 PN7642 registers and EEPROM concept

The PN7642 uses internal registers to adapt and optimize the functionality and performance for each of the supported protocols and data rates dependent on the connected antenna, matching network and receiver path. It offers an EEPROM, which contains the default settings for all the supported protocols. These settings are loaded into the registers with the PN76_Sys_LoadRfConfiguration(…) method for each supported protocol and data rate.

The default EEPROM configuration are optimized for the 45 mm x 45 mm antenna of the development board and can be changed by the user in case a customized antenna and matching network is used. The load RF configuration method allows initializing multiple registers with an efficient single API call and allows distinguishing between transmit and receive configurations. Update of the registers relevant for a selected protocol is done by copying the content of EEPROM addresses to registers. Not all protocols require the initialization of all or the same registers, the API *"PN76_Sys_LoadRfConfiguration(…)"* considers this.

Some of these settings **must** be adapted toward a new antenna design (e.g. the dynamic power control). All those design-specific settings must be stored in the PN76 family EEPROM to allow a proper functionality.

Some EEPROM configuration data is independent from the used protocols and defines, e.g., the startup behavior of the PN7642 or the functionality of low-power card detection (LPCD). This configuration data can also be adapted for optimum performance of the chip.

See AN13925 for further detailed information how to handle the EEPROM configuration.

Note: *EEPROM writes must be used with extreme care, due to limited write cycles (100k) and the potential for corruption if power is lost during the operation! A reset or power loss during an EEPROM write can corrupt the currently written EEPROM page. If an EEPROM write is performed, make sure that stable power is available and no reset is performed until the write is complete.*

To check on EEPROM and Flash integrity run the API *"PN76_EDL_Get_Integrity_NXP_Fw(…)"*.

7 Hardware

The PN76 PSP (Product Support Package) builds upon the PNEV7642A board and LPC55S16-EVK board.

For the PN7642, a couple of different board revisions have been available. The latest one is the PNEV7642A Revision B. Previous boards cannot be bought any longer and are not described in this document.

Table 5. PN76 family boards

Board name	Description
PNEV7642A Rev.B	Section 7.1
PNEV7642A Rev.A	The main difference to Rev.B is the CT interface for the TDAEV8035 Arduino Shield, which had an issue. This board is no longer available for customers.
PNEV76FAMA	No longer available for customers.

7.1 PNEV7642A Revision B

The PNEV7642A Revision B is easily identified by its large golden ground plane, added to help heat dissipation.

The schematics and layout files of the development board can be found at [20].

The development board assembled of two major components:

- The based board which hosts all the periphery and connectors.
  - Schematic: [13]
- The module board which hosts the PN7642 chipset and its minimum required components (See Section 7.2 "PNEV7642MA").
  - Schematic: [14]
  - The module board can be used for rapid prototyping on a custom PCB.

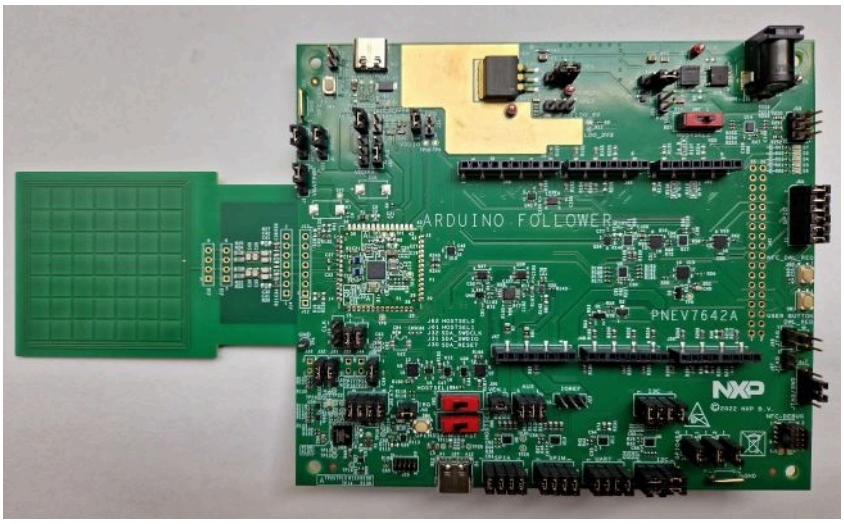

Figure 12. PNEV7642A Rev.B

7.1.1 Power supply

The development board can be either powered by the USB connector "J5", or from a power jack at connector "J9". The default settings use the power jack, which must be also used if the DC-DC is enabled (default).

Note: The PN7642 in default configuration requires an external power supply, i.e. the USB supply does not provide enough current (in-rush current spikes). If you want to power it via the J5 USB-C connector, make sure that your USB port can handle the necessary power (up to 1A). With active USB-Hubs, this usually works while using the ports on a notebook might be insufficient.

The external power supply must be between 5 V and 7.5 V DC (polarity does not matter), since the board provides a rectifier and LDO to supply the circuit with 6 V (optional, supply voltage ≥7.5 V), 5 V, 3.3 V, and 1.8 V.

The DC power input can cover a DC voltage around 7.5 V with a current of at least 1A. The polarity is not a concern since there is a built-in rectifier. The inrush current of the DC-DC can be up to 1.6 A, when enabling the RF-Field.

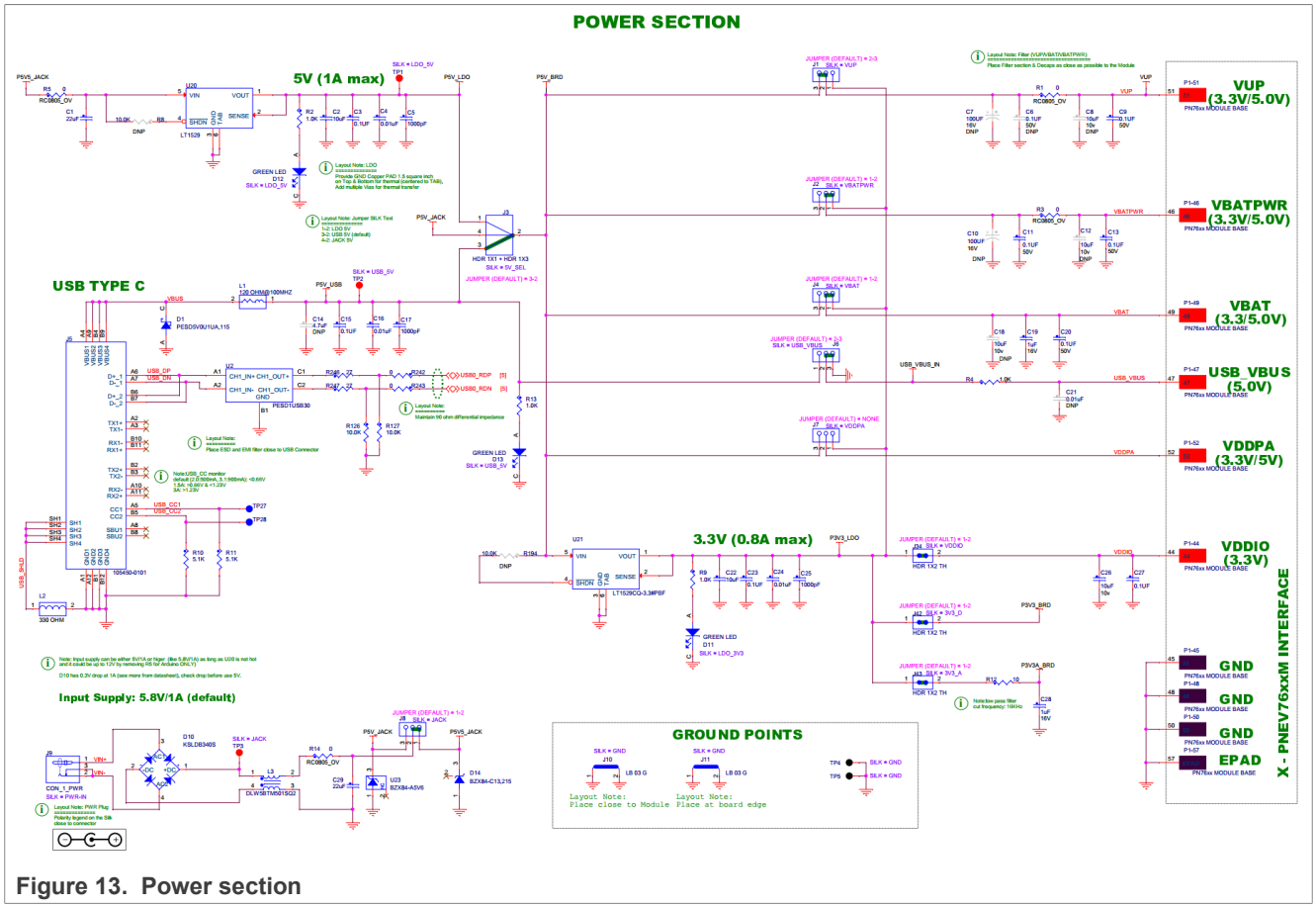

Figure 13. Power section

7.1.1.1 Power supply selection

To switch the power source, J3 has to be set according to below table.

Table 6. Power selection

Pins	Description
1-2	LDO 5 V
3-2	USB 5 V (default)
4-2	JACK 5 V - 7.5 V

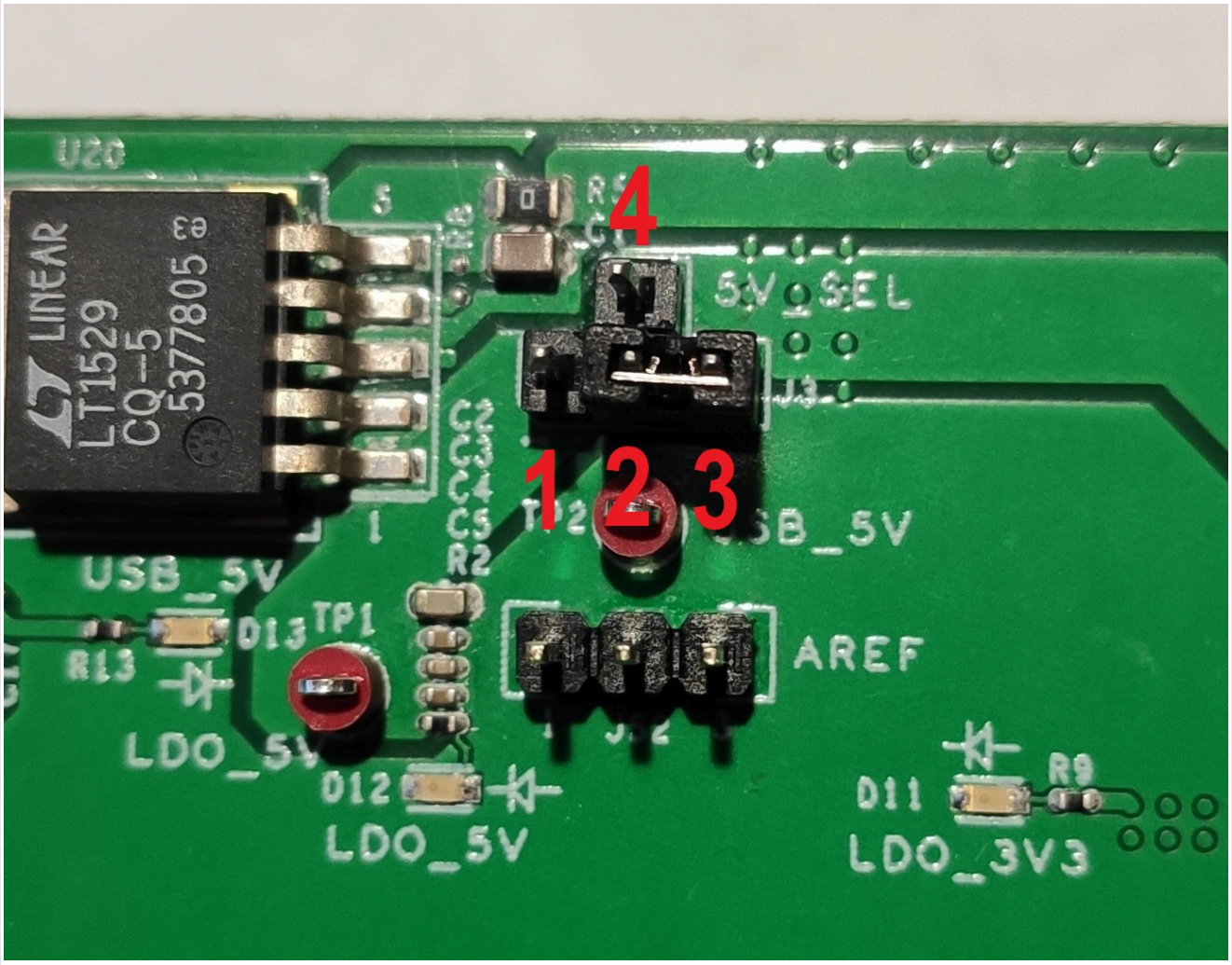

Figure 14. Power selection jumper

7.1.1.2 Power jack

The recommended supply method is via the power jack. Jumper "J8" determines if the power is fed into the 5 V LDO or directly supplied to the board net.

If the connected power supply does not output stable 5 V, the jumper must be set to 1-2 to make use of the onboard LDO.

Table 7. Input supply jumper setting

Pins connected	Description
1-2	External power supply delivers >5 V and is connected to the onboard LDO.
3-2	External power supply delivers exactly 5 V and is directly connected to board net!

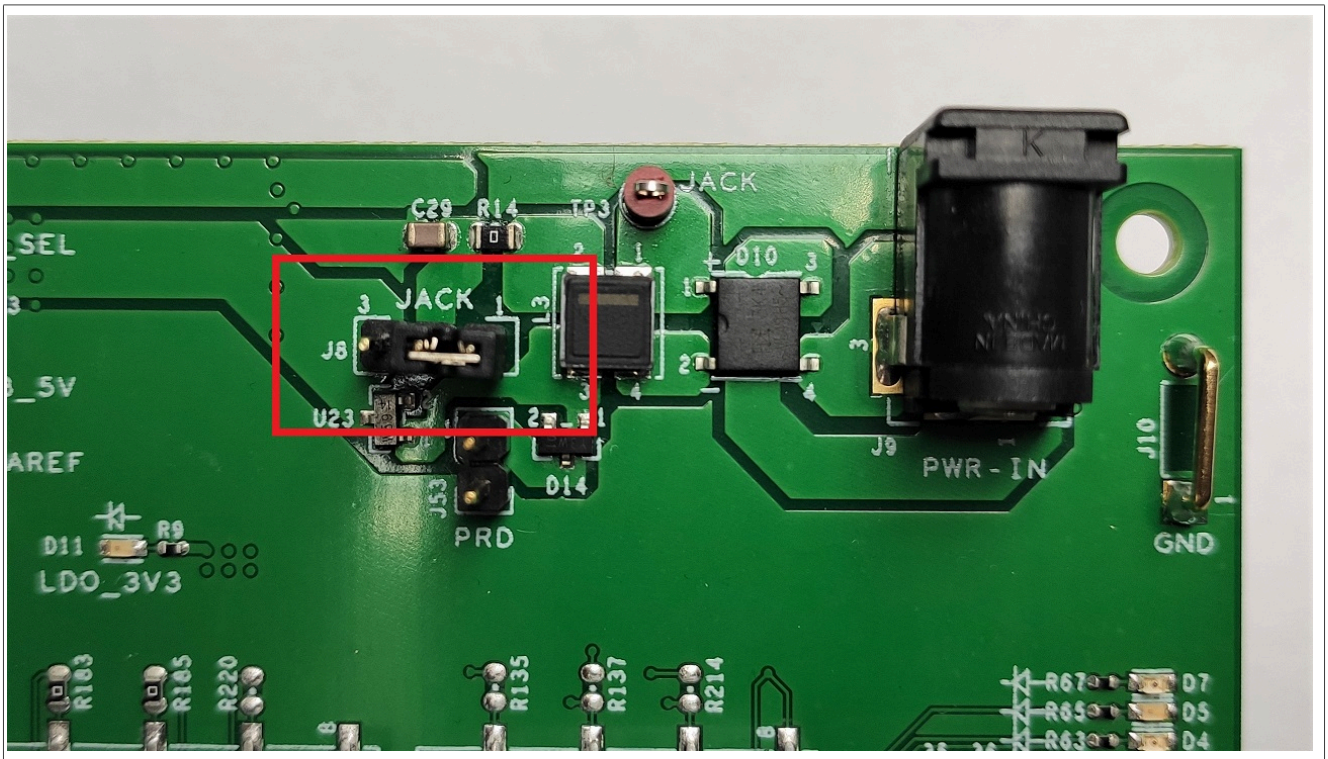

Figure 15. PNEV7642A development board power jack jumper

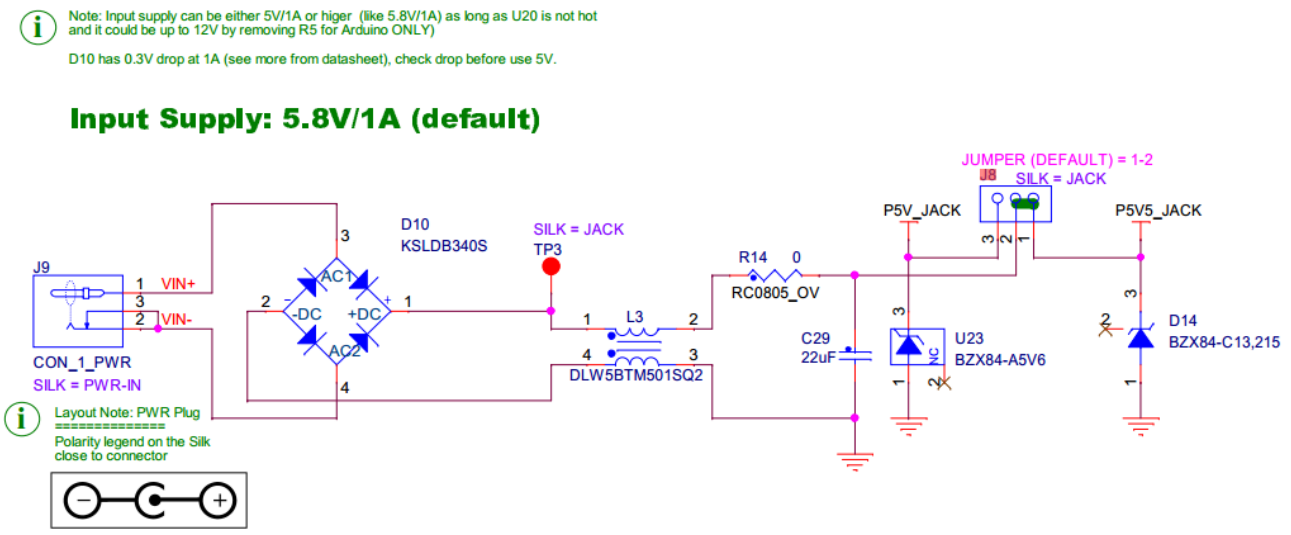

Figure 16. PNEV7642A development power jack schematic

7.1.2 Debug interfaces

The board offers two options of debugging interfaces, JTAG/SWD and the onboard OpenSDA. They are mutual exclusive. Hardware changes have to be made to switch from one to the other.

Per default, the JTAG/SWD interface (Section 7.1.2.1) for an external debugger is active.

How to switch to the onboard OpenSDA see [11].

7.1.2.1 JTAG/SWD interface

The PN76 controller can be programmed via an SWD interface, which is exposed as J21 "NFC-Debug" on the PNEV7642A board. To access the SWD interface, J17 has to be set to 2-3 (default).

To switch from SWD to JTAG pins 1-2 and 3-4 need to be closed. See Table 8.

The current SDK is built to use a SEGGER J-Link, MCU-Link, or a LPC-Link2. Those external debuggers are connected to J21 with a ribbon cable. In case you use a SEGGER J-Link make sure to have installed the latest SDK (>V7.84).

Table 8. J17 Interface jumper setting

Pins connected	Description
1-2	JTAG Interface
2-3	SWD Interface (recommended)
3-4	JTAG Interface

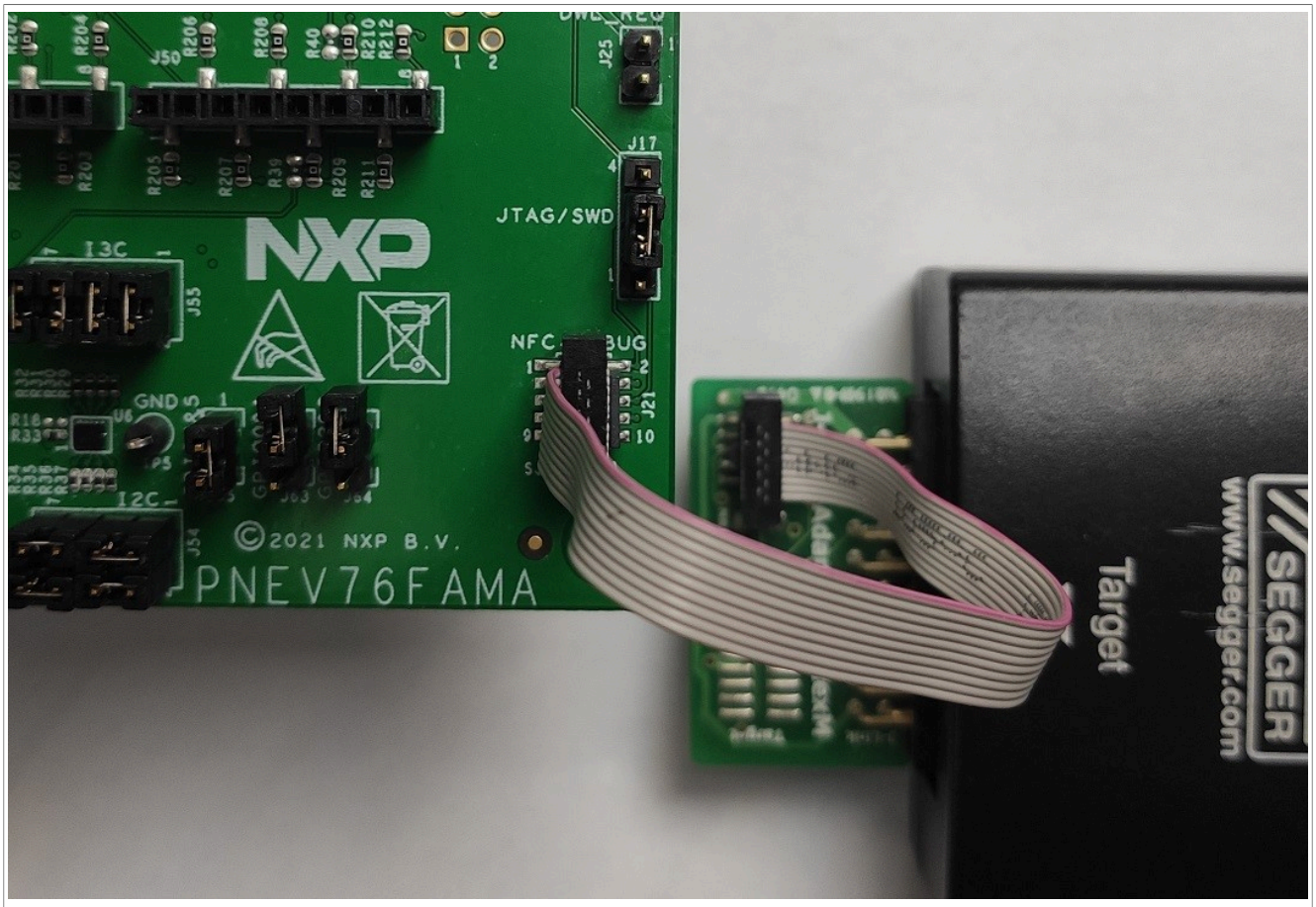

Figure 17. Debug interface PNEV76FAMA and PNEV6742A

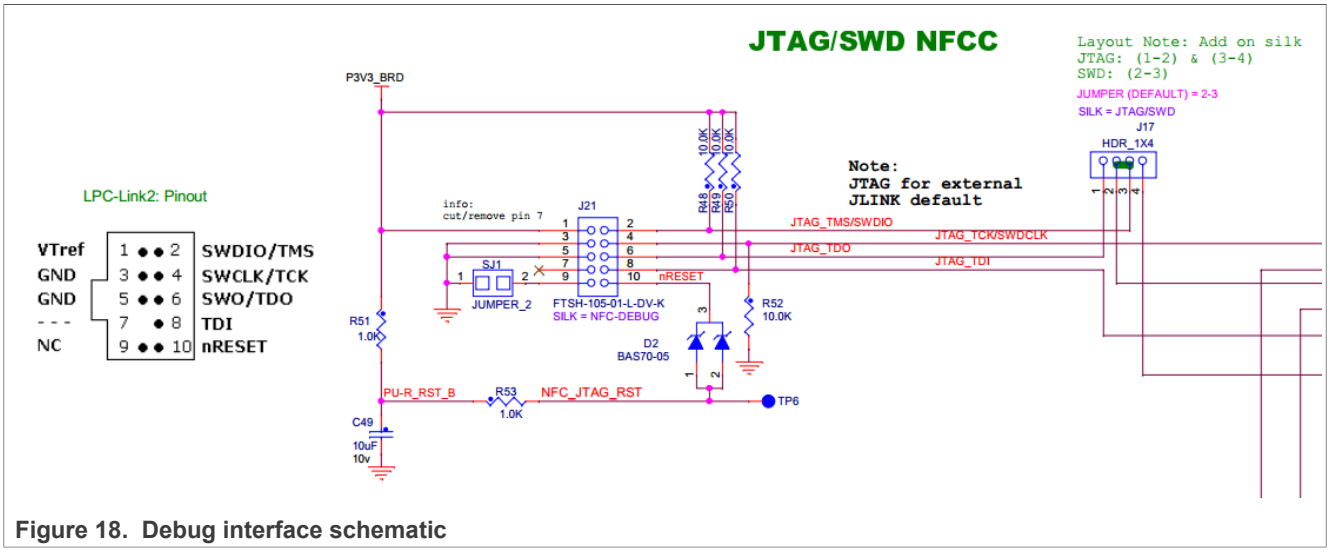

Figure 18. Debug interface schematic

7.1.2.2 OpenSDA interface

The PNEV7642A hosts a OpenSDA interface on the bottom-left corner. OpenSDA is an open-standard serial and debug adapter. It bridges serial- and debug communications between a USB host and an embedded target processor. J27 is the USB-C connector for OpenSDA while J29 is the JTAG interface for the microcontroller running the OpenSDA software.

By default, OpenSDA is not enabled. Refer to AN13898 – OpenSDA on PN76 development boards for more information.

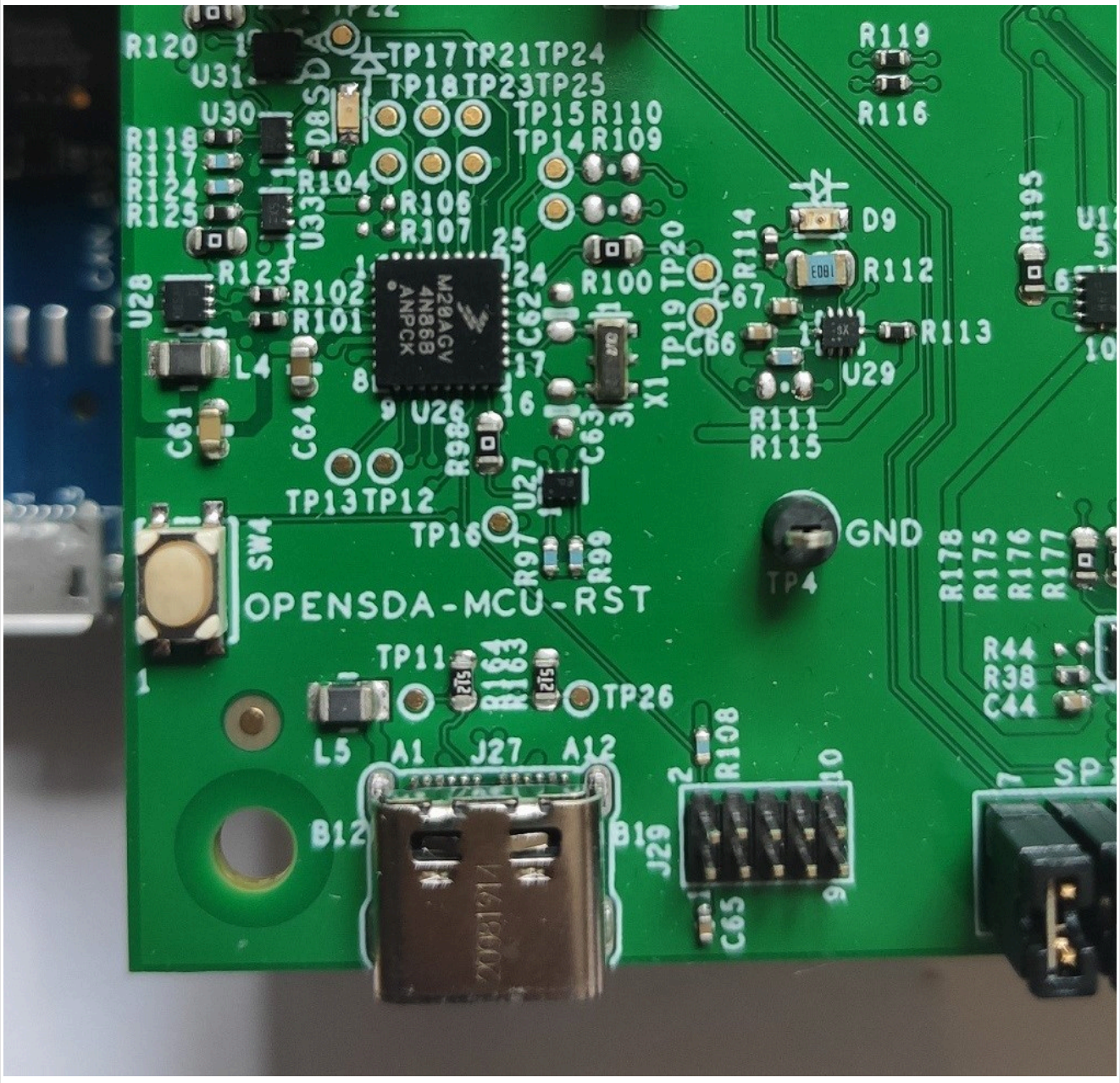

Figure 19. OpenSDA connectors

7.1.3 Antenna

The default antenna is a 45 mm x 45 mm antenna with some metal layer inside the antenna area. This antenna is not an optimum antenna as such, but intends to demonstrate the performance and register settings of the PN7642 under typical design constraints like LCD or some metal (e.g., PCB) inside the antenna area. The default settings provide an EMVC0 3.0 L1 analog compliance under the assumption that the antenna surface is a few mm above the antenna PCP.

The antenna connection uses the standard tuning circuit. The EMC filter is designed with a cut-off frequency of fEMC = 14.25 MHz, and the antenna impedance is tuned to Z = 15…16 Ω.

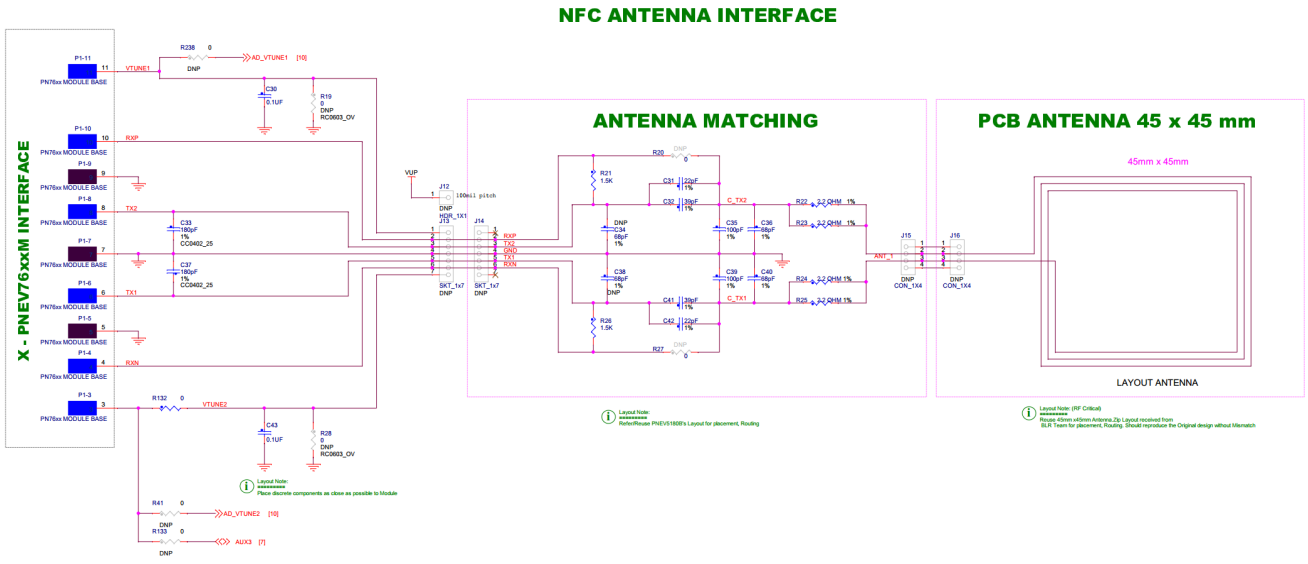

Figure 20. Antenna schematic

7.1.4 Jumper settings

The following chapters describe the most important jumpers for your daily debugging. For more information about other jumpers, consolidate the schematics of the base board [13].

7.1.4.1 Host interface selection

On this board, the HIF (host interface) can be chosen by three red switches. Those act either as pullup or pulldown with 10 kOhm toward the signal IOREF or GND.

The switches are not grouped together but spread over the board (SW7 near power jack, SW5 and SW6 near OpenSDA).

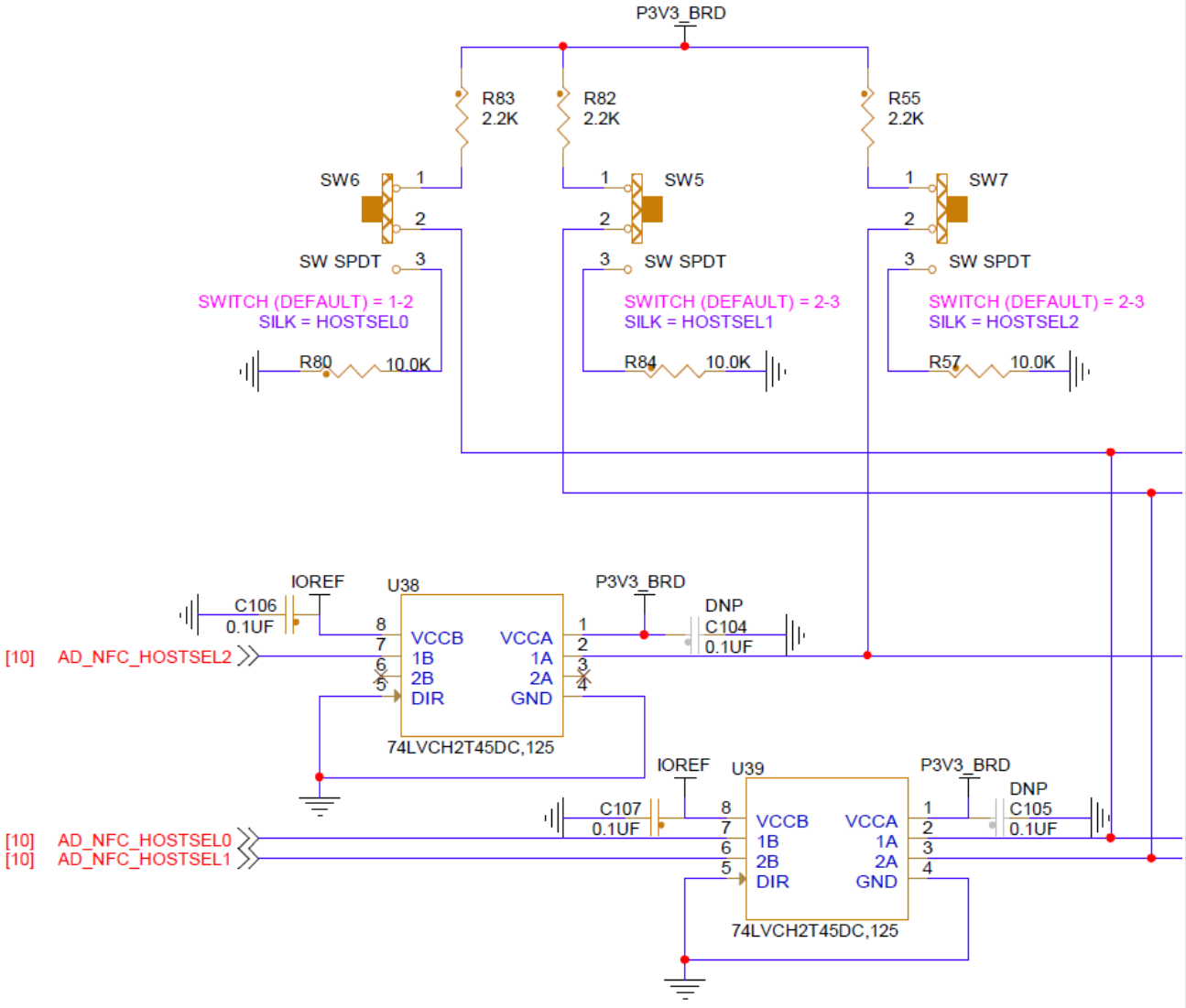

Figure 21. PNEV7642A HIF selection switches schematic

Following table defines the chosen host-interface. Where the value "0" represents a Low signal (GND) and "1" a high signal ("IOREF").

Table 9. PNEV7642A HIF selection

Interface	HOSTSEL2 (SW7)	HOSTSEL1 (SW5)	HOSTSEL0 (SW6)
I2C	0	0	0
SPI (default)	0	0	1
UART	0	1	0
I3C	0	1	1
USB	1	0	0
SPI	1	0	1
USB	1	1	0
USB	1	1	1

Note: *There are multiple options for USB. It does not matter which one is taken. Ultimately the signal to the controller is the same.*

7.1.4.1.1 Floating inputs on level shifters

As seen in Figure 21, there are two level shifters, U38 and U39 (74LVCH2T45DC), also connected to the lines. Those level shifters ensure that whatever voltage the Arduino Leader (connected on the bottom, PN7642 is the follower) device has, is translated to the actual working voltage of the PN7642 board.

Unfortunately, without an Arduino Leader/Controller, the inputs of those two chips are floating. This can lead to unwanted behavior. If the floating input is detected as low, the chip drives the output 1A/2A as low. This can interfere with the chosen settings of the red switches (SW5, SW6, and SW7), especially in a USB case, where SW7 has to be high.

7.1.4.2 I/O Reference (J23)

By default, IOREF (J23) is not connected and this represents the "standalone mode" of the board (no Arduino controller installed - like the LPC55S16 board).

- **Standalone:** Not populated.
- **Follower:** Populated on 1-2.
  - This is the case, if a host is installed. E.g., using the LPC55S16 as host.
- **Leader:** Populated on 2-3.
  - In case you have an arduino shield on top. Like the TDAEV8035 board.

7.1.4.3 LED control J63 and J64

J63 and J64 control the lines to the LEDs. If connected, the LEDs are controlled by the Arduino controller interface. If the PNEV76FAMA, or PNEV7642A Rev-B, is used in standalone mode, disconnect those two jumpers to use the onboard LEDs.

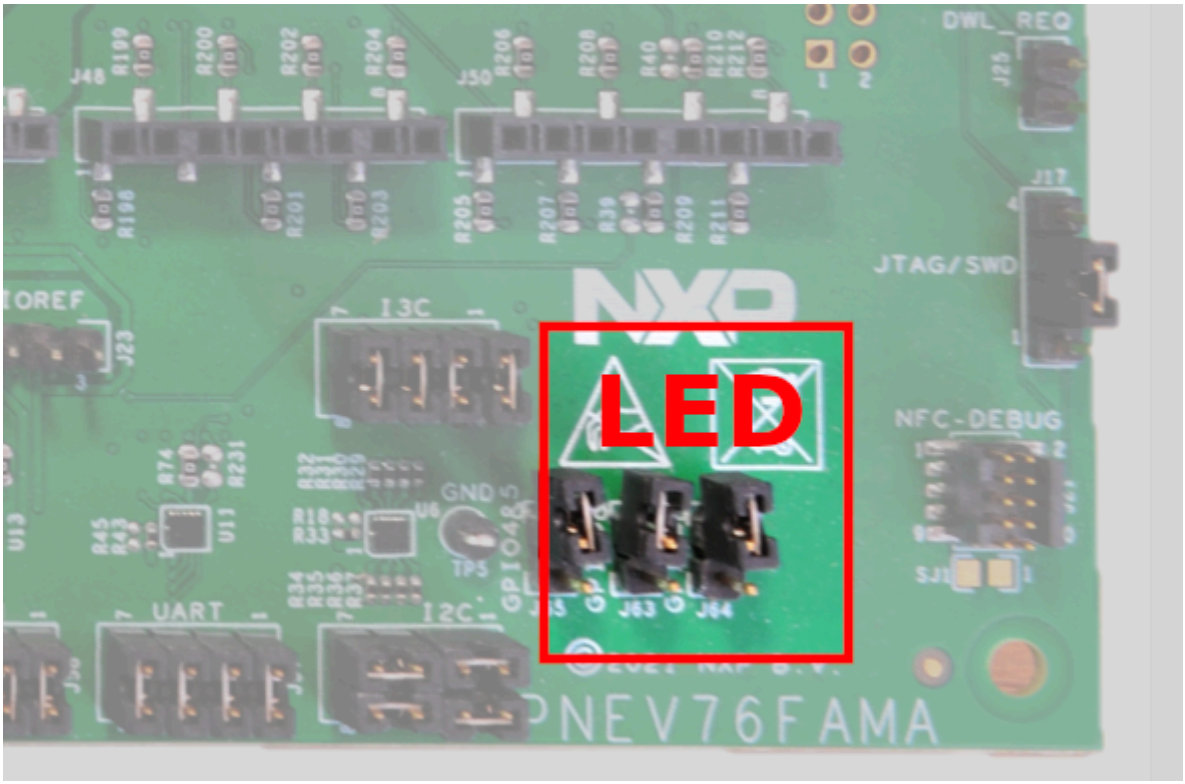

Figure 22. LED Control J63 J64

7.2 PNEV7642MA

The development board includes a small rectangle module board which hosts the PN7642 chip. This module implements all relevant components required to operate the PN7642, including crystal, DC/DC, blocking capacitors and the main part of the EMC filters.

The thin multilayer PCB allows the soldering on low-end PCBs, lowering the overall system cost and serving as an example of how PCB routing and component placing can be done for the PN7642 in commercial designs.

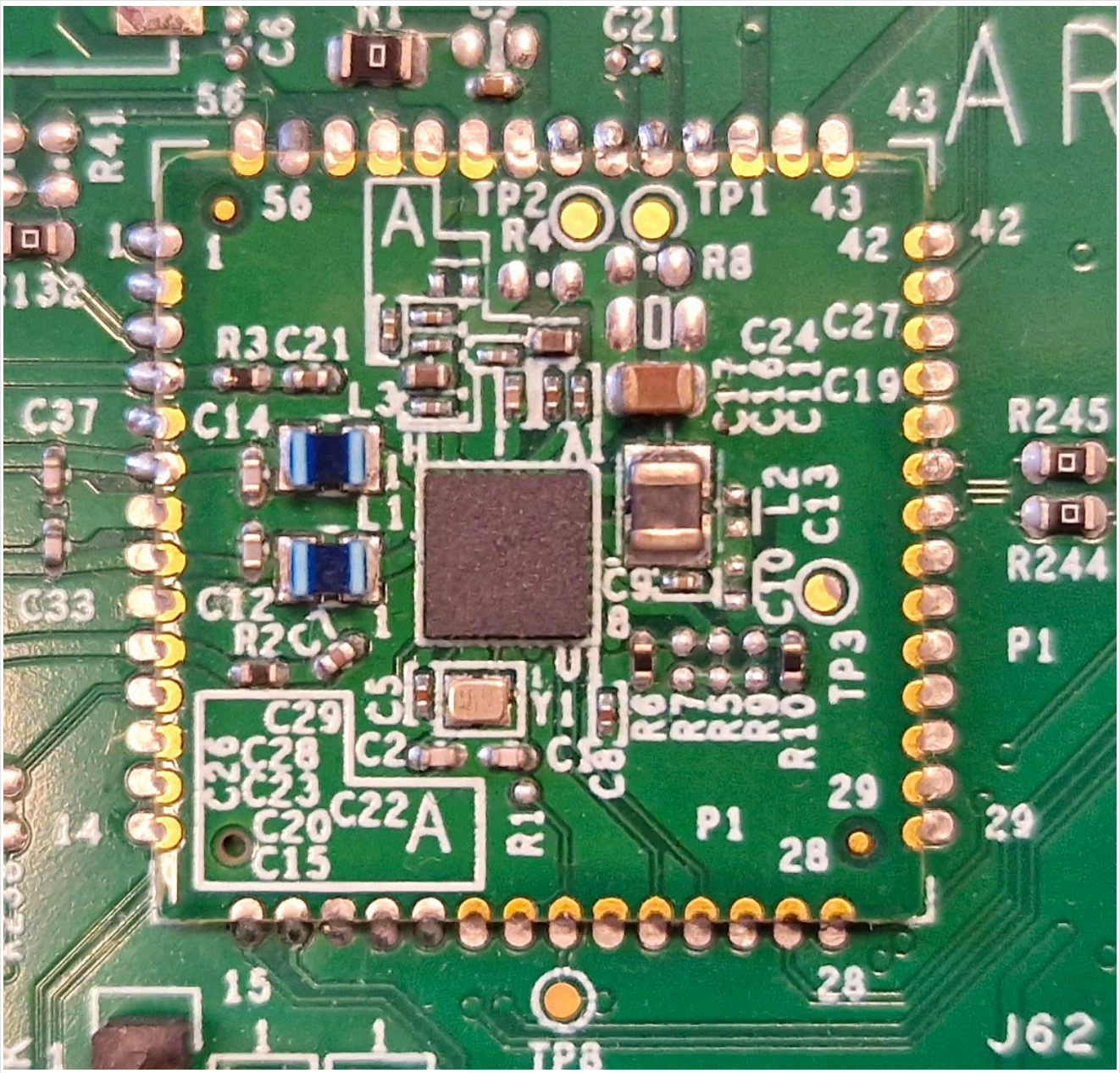

Figure 23. PNEV7642MA

The module board shown in Figure 23 is according to schematic file "spf-54814_a1".

DCDC disabled: On this module board, the DC-DC is disabled. R4, C29, and C13 are not populated (see Figure 24). VUP is connected to P5V_BRD through "J1" of the development board.

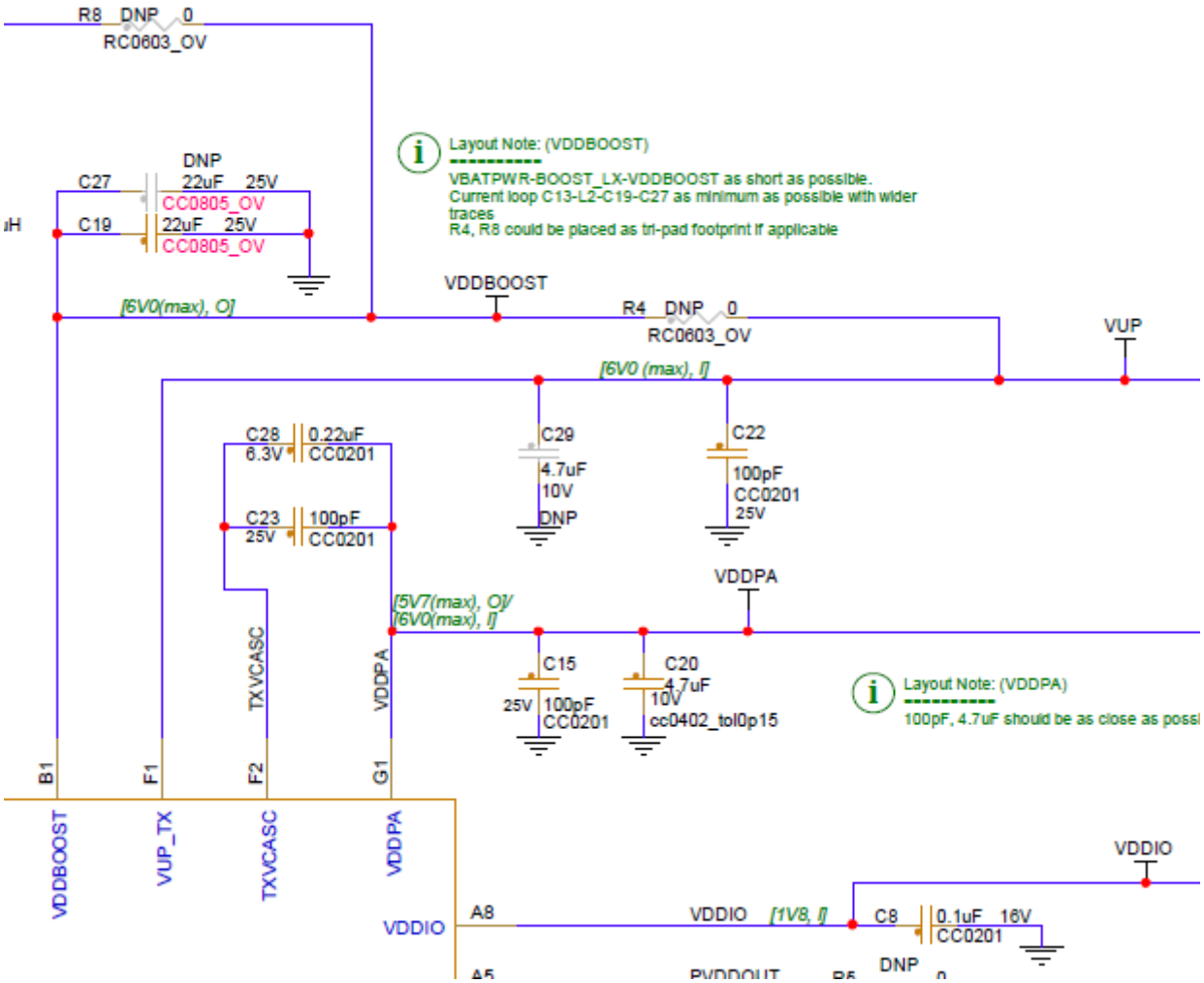

Figure 24. PNEV7642MA schematic snippet VUP

7.3 LPC55S16 host MCU

The default host controller for the PN76 SDK is the NXP LPC55S16. It is an Arm Cortex-M33 architecture MCU with built-in advanced security features. The examples (Host Software) are preconfigured to be run on the LPC55S16-EVK, which plugs in on the bottom side (Arduino controller) of the PNEV7642A Rev-B development board.

Development board link: [15]

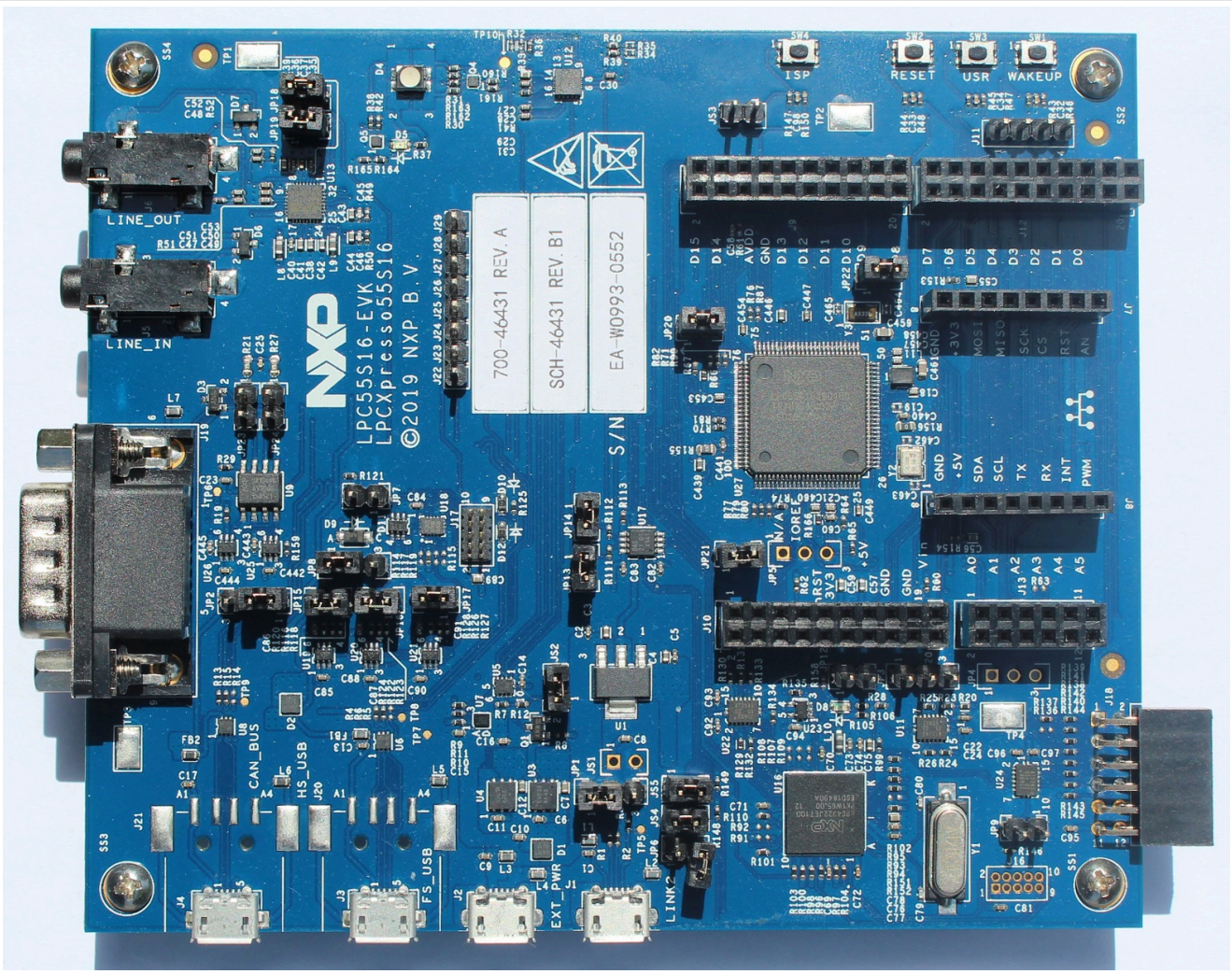

Figure 25. LPC55S16-EVK top view

7.3.1 USB connections

The LPC55S16-EVK is used within the PSP as a host microcontroller to showcase the host interface functionality of the PN7642 as well as updating the PN7642 firmware. The LPC55S16-EVK board does not need an external debugger as it has an LPC-Link2 onboard.

For any example within the PSP, which needs communication to the PC the High-speed USB connection on J4 (green square) is used.

The LPC-Link2 can be used by connecting a micro-USB cable to J1 (red square).

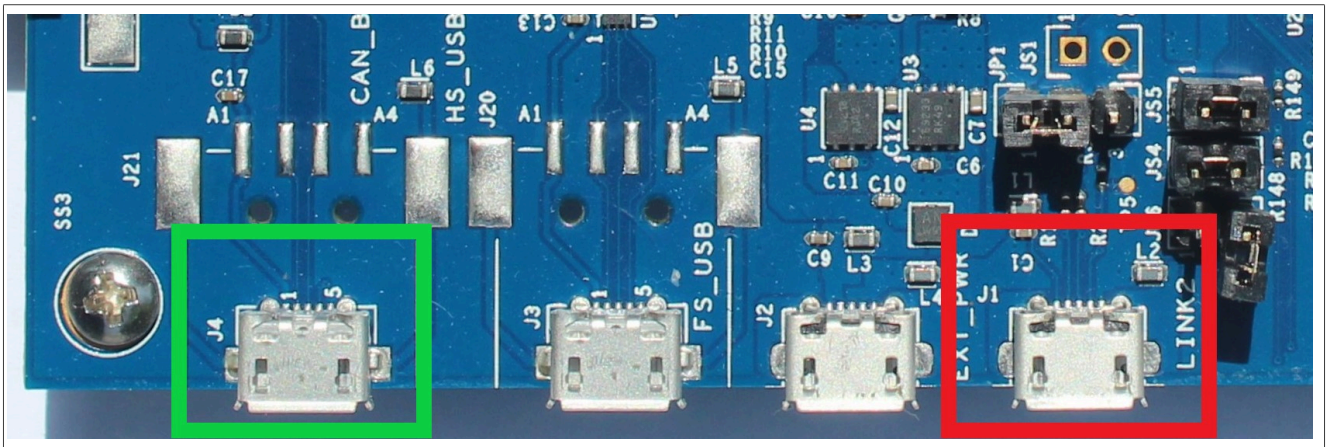

Figure 26. USB connectors

An external power supply on J2 is not necessary as the PNEV7642A development board shall be supplied with an external power supply.

For further information about the LPC55S16-EVK, checkout the website of NXP: https://www.nxp.com/design/:LPC55S16-EVK.

7.4 Connecting the LPC55 with the PNEV76 development board

Some of the PN7642 PSP examples require the LPC55S16-EVK as host MCU. One of them is the firmware download example, which is used to update the PN7642 firmware. For this, we have to stack the PNEV7642A Rev-B development board on top of the LPC55S16-EVK board.

Align Pin.1 of J36 of the PNEV7642A Rev-B development board with Pin.1 of J9 of the LPC55 board. The last 4 pins, 17 - 20, of J12 of the LPC board are not connected. As well as pin 1-4 of J10 stay unconnected.

The green-marked area in Figure 27 is to be connected with the PN76 board while the red area stays unconnected.

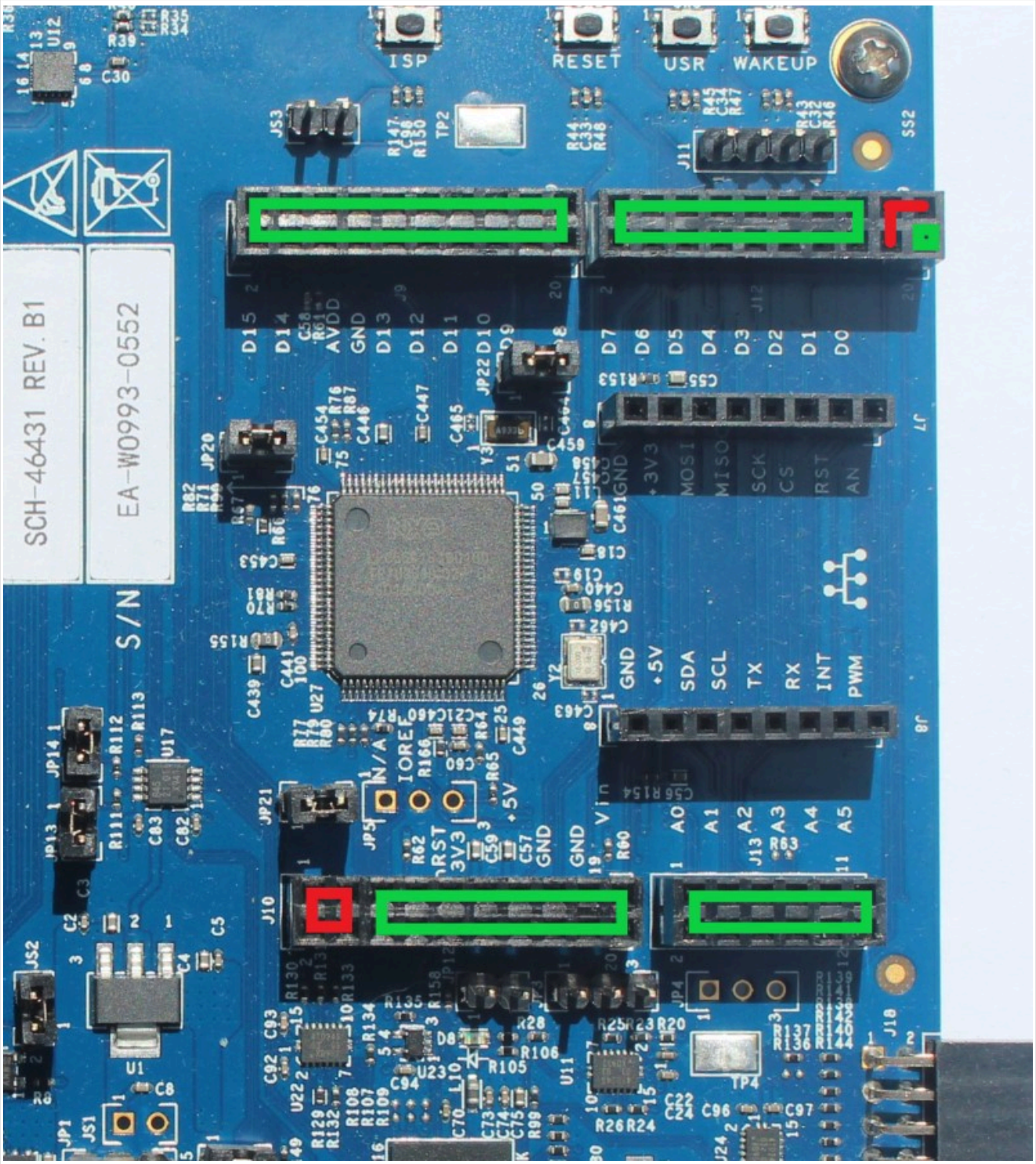

Figure 27. LPC55 arduino alignment

The following pictures help to understand how to align the PNEV7642A Rev-B development board on top of the LPC55S16-EVK.

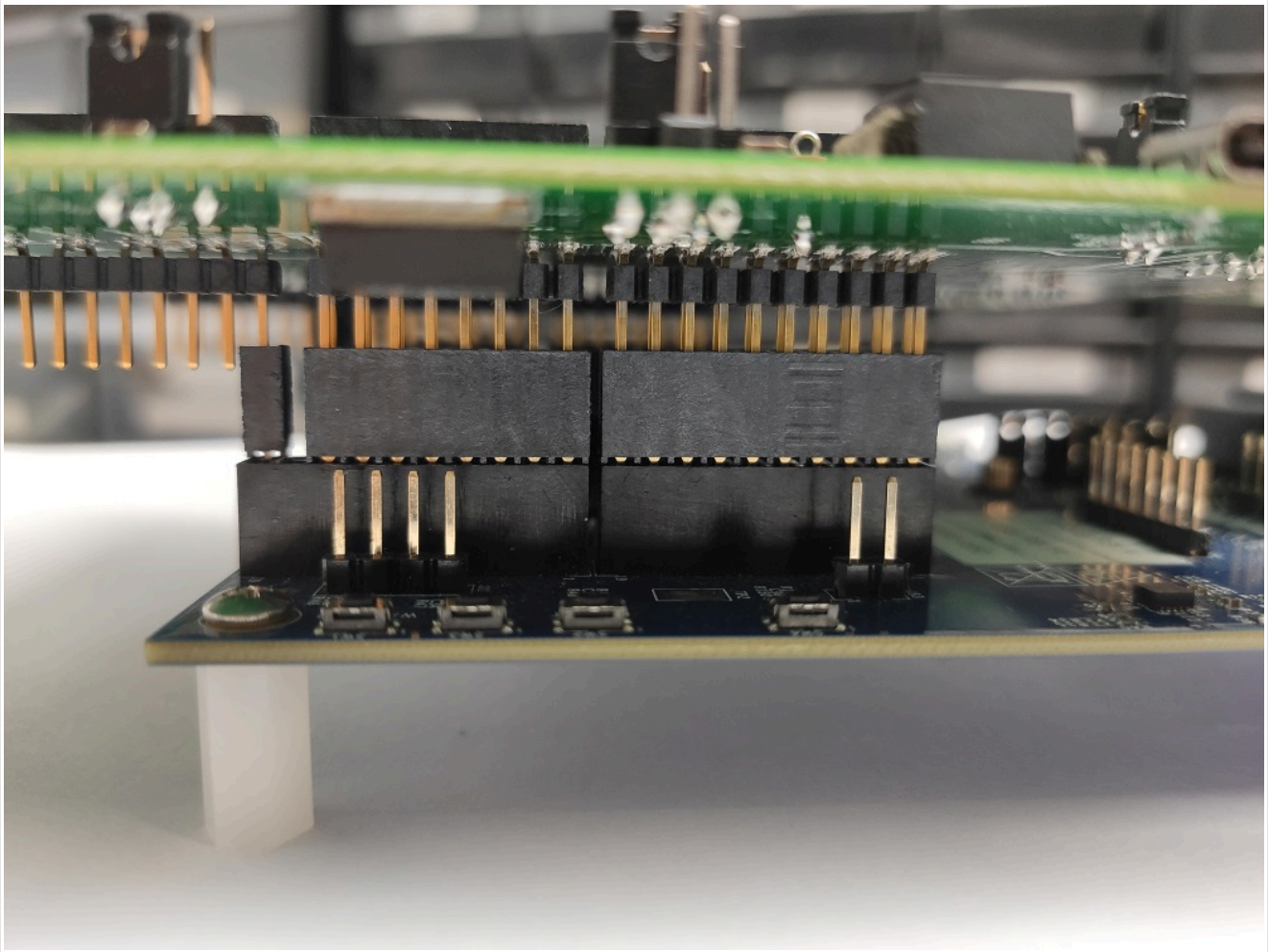

Figure 28. Alignment J9

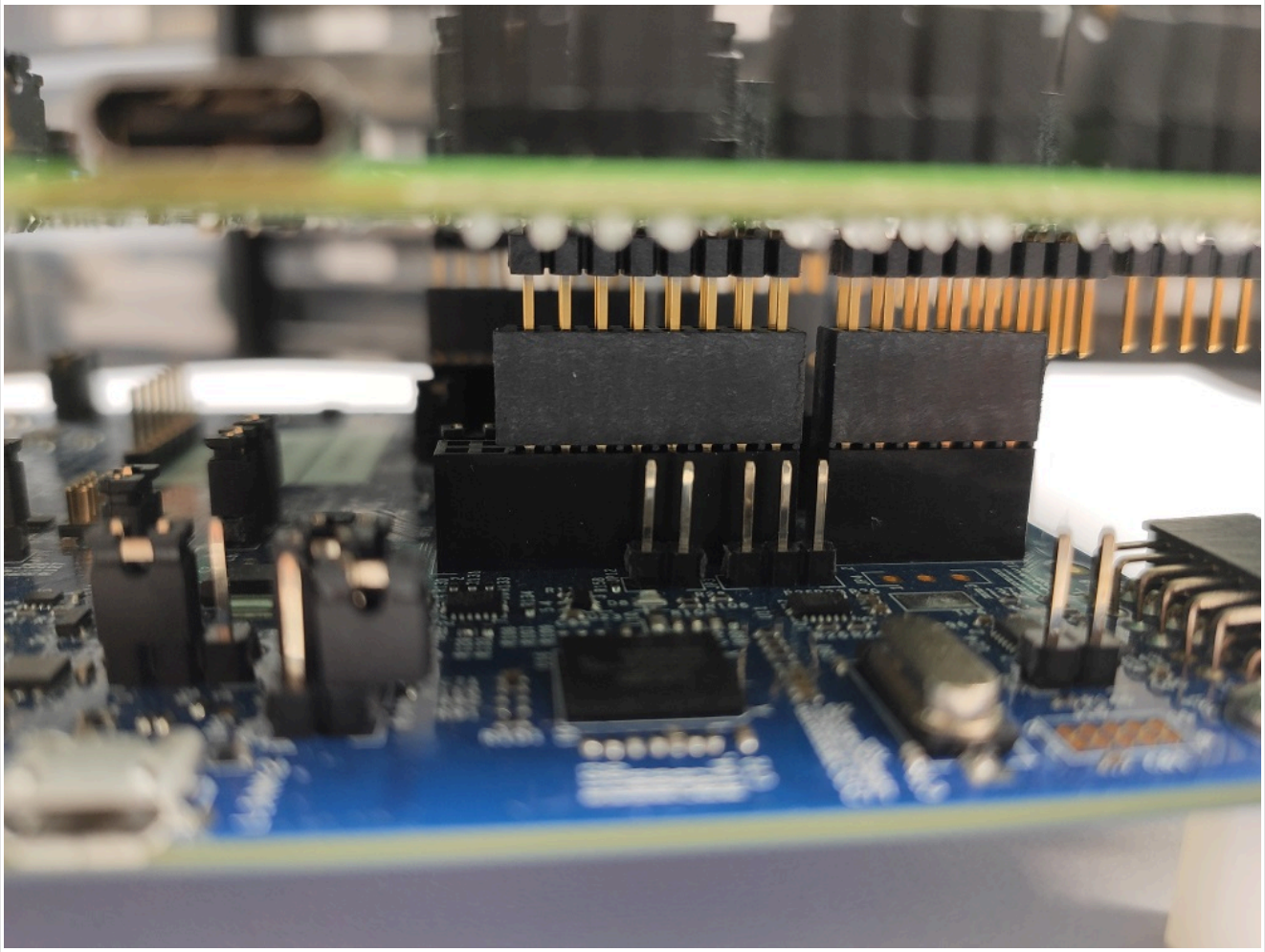

Figure 29. Alignment J10

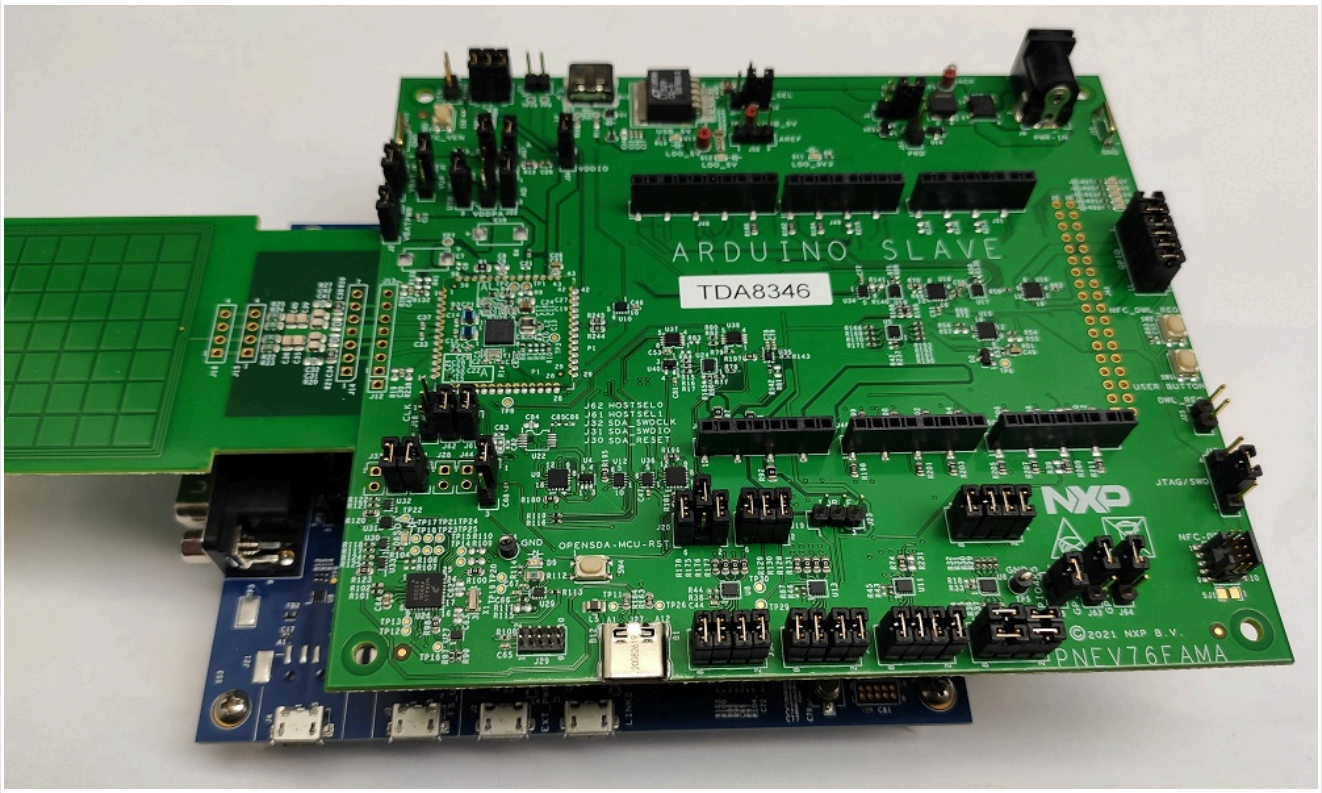

Figure 30. Stacked PNEV76FAMA on LPC55S16-EVK

7.5 TDAEV8035

The TDA8035 is a contacted card reader. It supports class A (5 V), class B (3 V), and class C (1.8 V) and a single contacted card. For the PN7642 development boards, a TDAEV8035 board is available which plugs-in on top of it, as Arduino follower. It hosts two TDA8035 chips and supports two contacted cards.

Two of the TDAEV8035 boards can be plugged on each other to host a maximum of 4 TDA8035 chips in parallel.

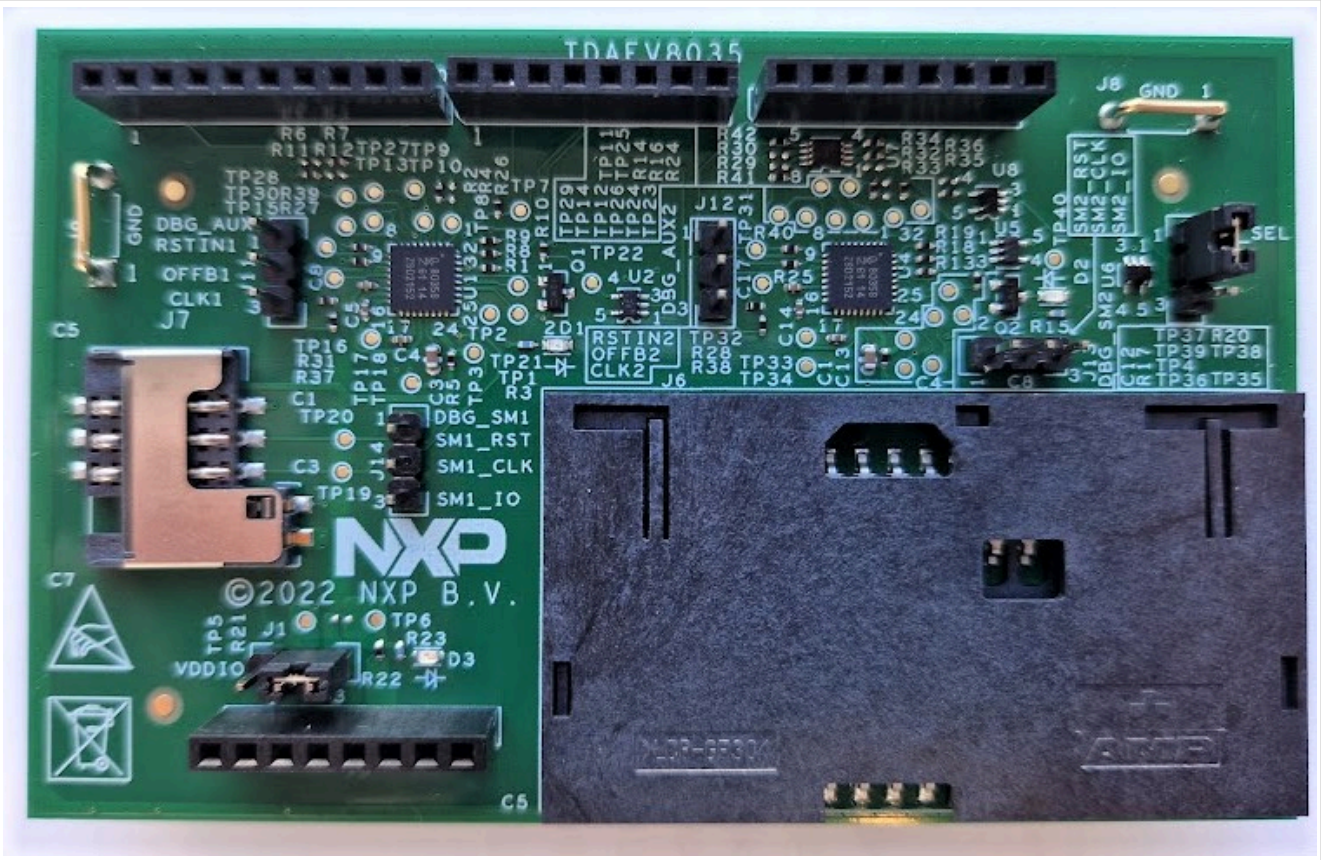

Figure 31. TDAEV8035

All CT examples of the MCUXpresso SDK require at least one TDAEV8035 to be connected.

7.5.1 Stacking of boards

The TDAEV8035 is plugged in on top of the PNEV76 development boards (connector "Arduino follower").

J3-1 of TDAEV8035 is aligned with J46-1 of PNEV7642A. The rest falls into place correctly.

Table 10. TDAEV8035 to PNEV7642A alignment

TDAEV8035		PNEV7642A
J3	aligned with	J46
J4	aligned with	J49
J5	aligned with	J51
J2	aligned with	J47
	not connected	J48
	not connected	J50

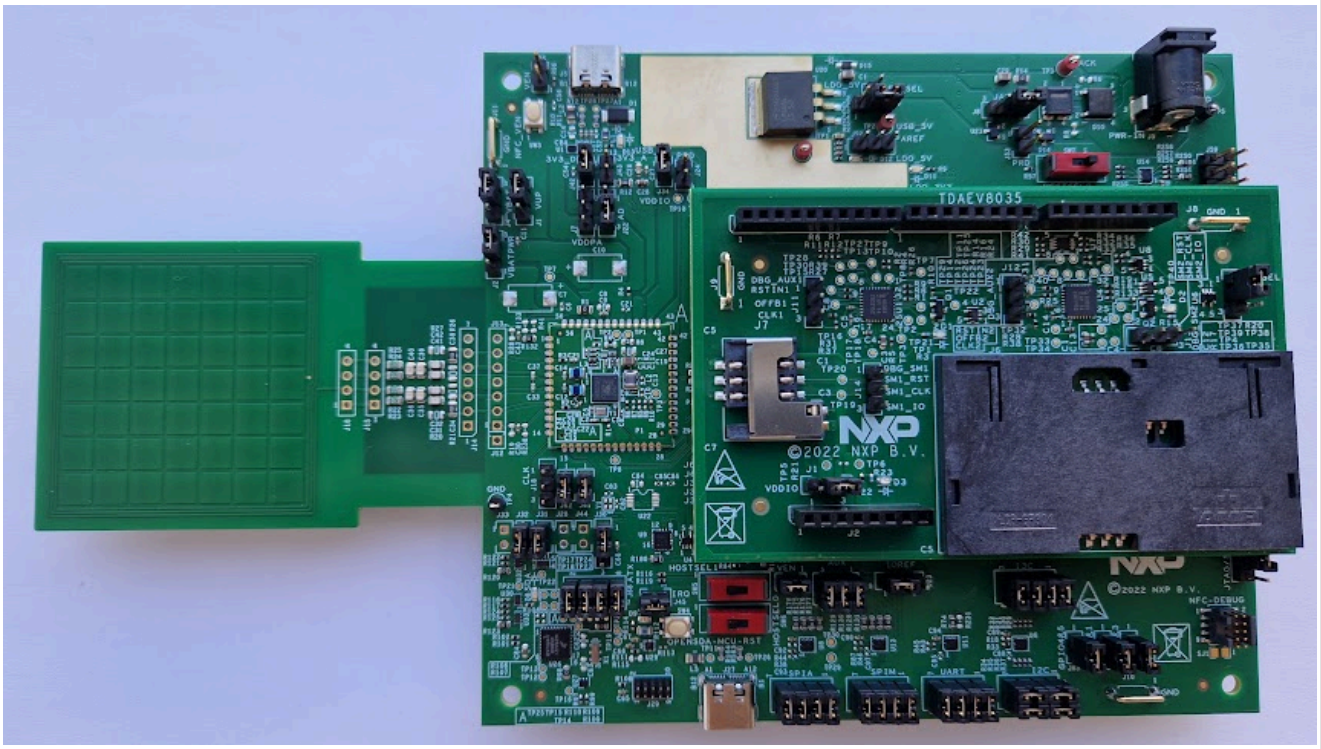

Figure 32. TDAEV8035 on PNEV7642A

7.5.2 Jumper settings

To make the TDAEV8035 board functional, some jumpers on the PNEV7642A board must be set.

Table 11. Jumper settings

Jumper	Note
J59 - "CT"	Found in the top-right corner. A 2x6 jumper array where 1-2, 3-4, and 5-6 have to be connected.
J23 - "IOREF"	Must be set to leader configuration (2-3 connected).

7.6 Setup without DC-DC

A typical use of the PN7642 includes the DC-DC, which then is controlled automatically by the DPC (dynamic power control). This controlled DC-DC provides the optimum input for the TxLDO, resulting in minimal loses, i.e. the minimal self-heating.

For some use-cases, such as the ULPCD (Ultra Low-Power Card Detection), the DC-DC cannot be used. For this some settings must be considered.

7.6.1 TxLDO voltage drop

When the DC-DC is not used, the self-heating must be checked carefully. Especially, if the VUP is supplied externally with levels higher than VBAT. The TxLDO losses can increase a lot, if the DPC reduces the VDDPA.

To avoid overheating, the firmware uses a maximum limit of the TxLDO voltage drop, which is set in the EEPROM setting DPC_TXLDO_MAX_DROPOUT. When this voltage drop is reached, the DPC does not further reduce the VDDPA.

For the default value and register description, see [1].

Example: The VUP is externally supplied with 5.5 V and DPC_TXLDO_MAX_DROPOUT is configured to 3.6 V. Then the VDDPA cannot be lower than 1.9 V.

Caution: Be careful when increasing the DPC_TXLDO_MAX_DROPOUT. This might overheat the PN7642 in some corner case operation.

7.6.2 Maximum VDDPA

The maximum VDDPA shall only be set to a value of "VUP - 300 mV", if the TxLDO is used.

A typical use case might be the operation without DC-DC, but with TxLDO being used, as, for example, for ULPCD. The VBAT = VUP drives the TxLDO, which provides the VDDPA.

Example: The VBAT is 5.0 V, as a result the maximum VDDPA (set in TXLDO_VDDPA_MAX_RDR) shall not be higher than 0x20 → VDDPAmax = 4.7 V.

8 Software

All PN7642 relevant software can be found on nxp.com.

This list is an overview of existing software packages, their purpose, and required environment. How to obtain and work with the different packages is explained in later chapters of this document. Always check for the correct version of your SDK package. As different firmware versions of the PN7642 need different SDK versions (or changes within them).

The latest versions are found online. See Section 5 for the version compatibility.

Table 12. PN7642 SDK overview

Name	Description	Environment and equipment required
PN7642 MCUXpresso SDK	This software package contains the MCUXpresso SDK, which offers a fast variety of examples with the known workflow of any other MCUXpresso SDK.	MCUXpresso v11.6.1 (or later)
PN7642 Keil SDK	An SDK for the Keil IDE with a fast variety of examples.	
PN76 VCOM FW	User application, which implements the NFC Cockpit counterpart.	MCUXpresso v11.6.1. (or later)

Table 13. PN7642 complementary software

Name	Description	Environment and equipment required
Host Crypto Scripts	Containing scripts to encrypted customer application firmware and generate key material for the secured key store provisioning.	Python

Table 14. LPC55S16 host software overview

Name	Description	Environment and equipment required
LPC55S16 Host Software	Containing the LPC55S16 host software examples such as the secure firmware download example to update the PN7642 firmware as well as Host-Interface examples. This software package runs on the LPC55S16 counterpart of the PN7642.	LPC55S16-EVK

8.1 MCUXpresso SDK

MCUXpresso homepage: [16]

The PN7642 MCUXpresso SDK is a full-integrated SDK, which supports not just software examples but also the configuration tool. The SDK itself has to be downloaded from the NXP website and installed manually. It is not listed in the SDK manager of MCUXpresso.

The installation of the SDK is described in: Section 9.1.1.2 "Install SDK"

The SDK offers many examples designed for the PNEV7642A development board:

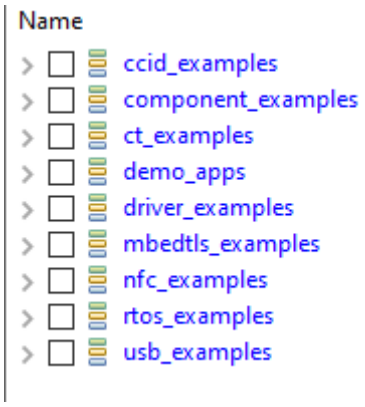

Figure 33. SDK examples

How to import and build examples is described in: Section 9.1.2 "Import SDK example"

8.2 Host software (LPC55S16)

The host software package holds examples to be used with the LPC55S16 and MCUXpresso, to interact with the PN7642. For this, the PN7642 has to be connected with the LPC55 as explained in Section 7.4.

The host software can be downloaded on nxp.com at PN7642 software: [17]

8.2.1 Import projects

Download and extract the software package. Create a separate folder for the MCUXpresso workspace and open MCUXpresso using this folder as the workspace.

Choose import existing projects and make sure not to check "copy into workspace" by doing so.

In case you have not installed the LPC55S16 SDK in MCUXpresso previously, you have to do so now as it is a prerequisite to work with this software package. The installation is done via the SDK wizard on the bottom right. Click "Install New plug-in SDKs".

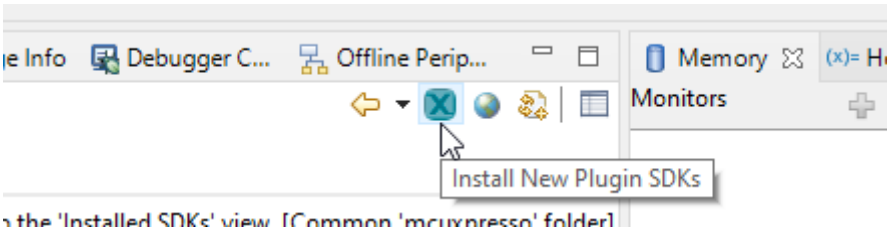

Figure 34. Install new plug-in button

Filter for "55S16" (as the actual SDK board is not called LPC55S16 filtering for this term will not give you any results) and chose the "lpcxpresso55s16" board. Click "Install" at the bottom.

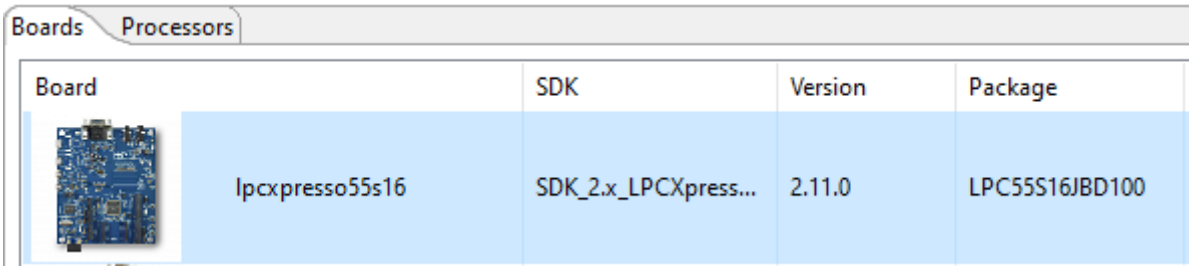

Figure 35. Choose LPC55S16 board

Having the projects imported and the SDK installed, the workspace is ready to run the host utils examples.

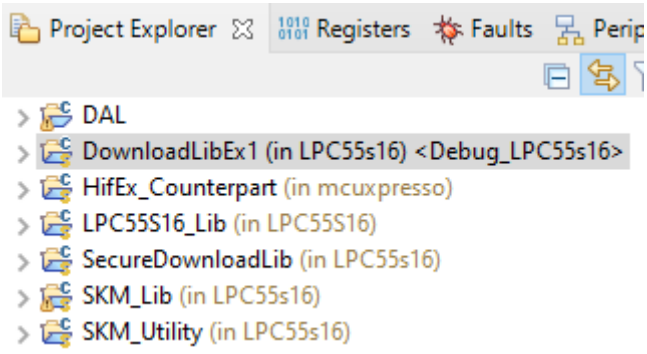

Figure 36. Project explorer

8.2.2 Secure firmware download

The secure firmware download example "DownloadEx1" is updating the PN7642 NXP secure firmware. In this chapter, it is explained how to change, and run the example in your environment.

Prerequisite: You must have a firmware file (.esfwu), which can be downloaded at Secure Files on NXP.com.

On the PNEV76FAMA board: Check USB_VBUS_J6 regarding the correct VBUS setting. As this example uses SPI/I2C host interface and not USB.

On the PNEV7642A Rev-B board: no actions to be taken.

8.2.2.1 Edit firmware location

At the top of the main file of *DownloadEx1*, the path to the firmware files is defined. This has to be changed to meet the individual path. For this example, a folder at the root of "C:" with the name "FW_Download" has been created, in which the downloaded *.esfwu* firmware file has been copied.

Users can either rename the firmware file to match the definition in the example or edit the definition in the example to match the firmware files name. The latter has the benefit of not losing important information in the name, such as the version, which can be useful if users have to repeat the update process later and might have forgotten what firmware file was previously used.

In this example, the path has been changed to match the firmware filename:

```
58 /****************************** IMPORTANT ******************************************/
59 /* Specify the full Absolute File path for the Encrypted FW binary, to be downloaded */
60 #define FIRMWARE_FILE_PATH      "C:\\FW_Folder\\pn7640_v01.01.01_prod.esfwu"
61 #define NS_FIRMWARE_FILE_PATH   "C:\\FW_Folder\\pn76xx_cl_edl_customer.esfwu"
62 /****************************** IMPORTANT ******************************************/
```

Figure 37. Firmware file path editing

```
55 /****************************** IMPORTANT ******************************************/
56 /* Specify the full Absolute File path for the Encrypted FW binary, to be downloaded */
57 #define NXP_FIRMWARE_FILE_PATH      "C:\\FW_Folder\\NXP_Firmware_v03.05.esfwu"
58 #define CUS_FIRMWARE_FILE_PATH      "C:\\FW_Folder\\Customer_App_vXX.YY.esfwu"
59 /****************************** IMPORTANT ******************************************/
```

Figure 38. Firmware file path editing, new

For updating the PN7642 firmware, the macro "FIRMWARE_FILE_PATH" has to match the location of the NXP secure firmware file.

Pay special attention to using a double backslash as a single backslash is marking a special character.

Note: *Depending on the Host Software package and its version the naming can be slightly different.*

8.2.2.2 Run firmware update

By running the example, there are several options printed in the console to chose from.

```
******** Secure Firmware Update ********

 Select the Option
        - Enter 1 for FW Version.
        - Enter 2 to Get DieID.

        - Enter 3 to perform SOFT RESET.
        - Enter 4 to CheckSessionState.

        - Enter 5 for Secure Firmware Update
        - Enter 6 for Non-Secure Firmware Update

Select Option:
(For MCUXpresso, you may have to press many enter keys after your input)
```

Figure 39. Firmware update options

Choosing the first option reads out the current firmware version of the PN7642. It is also a good check to see if the connection is working properly.

```
(For MCUXpresso, you may have to press many enter keys after your input)1
Option 1 selected
GetFirmwareVersion func
Secure FW ver     : 1.0

Non Secure FW ver : 0.0
```

Figure 40. Read firmware version

- Secure FW ver: represents the NXP firmware version on the PN7642
- Non Secure FW ver: can be assigned by updating the user application space via download mode.

The connected PN7642 is running firmware version v01.00. If the path to the firmware file is correct and the firmware file itself is valid, running option 5 "Secure firmware update" starts the update process.

```
Select Option:
(For MCUXpresso, you may have to press many enter keys after your input)5
Option 5 selected
Secure Firmware upload func
Please Wait..
Warning : Trying to either Downgrade or load same Fw
Current
MajorVersion = 1 bMinorVersion = 0
updating to
MajorVersion = 1 bMinorVersion = 1
**********************************************************************
Successful firmware upload
```

Figure 41. Update firmware version

The update may take a while. At the end, a successful update is indicated by the prompt of "Successful firmware upload". To verify, a read firmware version can be executed once more.

9 Development environment

MCUXpresso version 11.6.0 (or later) is supporting the PN7642.

Table 15. List of SW tools for Windows environment using MCUXpresso IDE

Tool	Recommended minimum Version
MCUXpresso IDE	Version 11.6.0 or later.
Debugger/Programmer	SEGGER J-Link v7.8 (and later), MCU-Link, or LPC-Link2

9.1 MCUXpresso environment

The PN7642 MCUXpresso SDK is a full-integrated SDK, which supports not just software examples but also the configuration tool. The SDK itself has to be downloaded from the NXP website and installed manually. It is not listed in the SDK manager of MCUXpresso.

The installation of the SDK is described in Section 9.1.1.2 "Install SDK".

The SDK offers many examples designed for the PNEV7642A development board:

9.1.1 MCUXpresso IDE

The latest version of MCUXpresso can be downloaded from the NXP website at: https://www.nxp.com/design/software/development-software/mcuxpresso-software-and-tools-/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE

This chapter describes how to import and install the PN76 SDK into MCUXpresso. How to import examples from the SDK as well as flashing and debugging those. For additional information, look into the MCUXpresso documentation available on the MCUXpresso website.

9.1.1.1 Install IDE

MCUXpresso is available for Windows, Linux, and MacOS. Follow the installation instruction of your operation system. The installation of MCUXpresso is not described further in this document. If you encounter issues or have questions about the MCUXpresso installation, consult the MCUXpresso user manual and website.

9.1.1.2 Install SDK

1. Make sure that you have either downloaded the PN7642 SDK from the nxp website or received it from an NXP official.
2. Close the "Welcome page" of MCUXpresso.
3. Go to the "Installed SDKs" tab on the bottom. In the following picture, multiple SDKs are already installed (highlighted in red). The following installaiton procedure does not change on depending on SDKs previously installed.

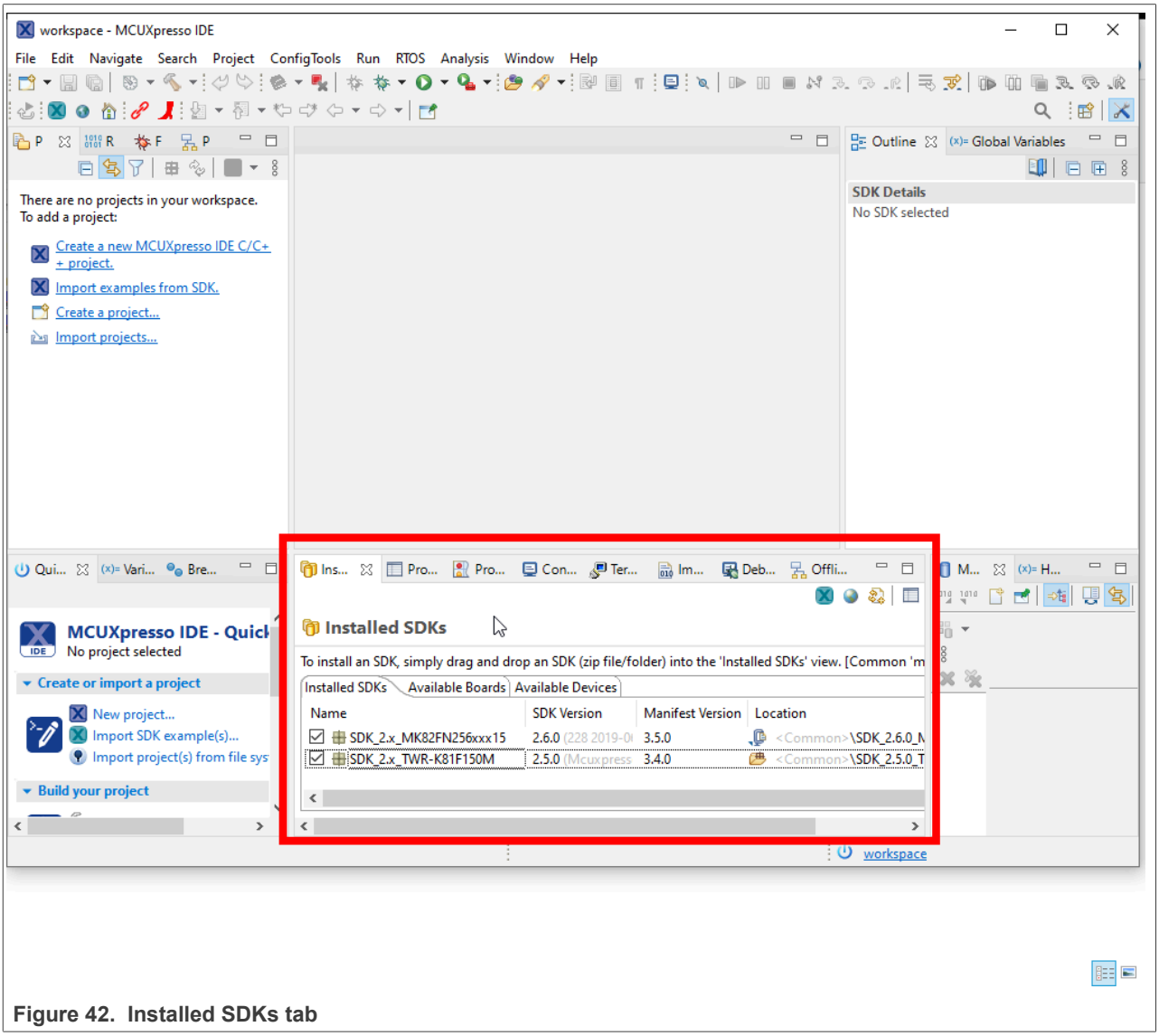

Figure 42. Installed SDKs tab

4. Drag and drop the PN7642 SDK file "PN7642_MCUXpresso_SDK_…. .zip" into the "Installed SDKs" tab. You most probably get asked if you allow installation by drag and drop. Press **OK** to proceed with the installation.

5. After a successful installation, the SDK will be listed in your 'Installed SDKs' tab. Where you can see the name, SDK Version and other information.

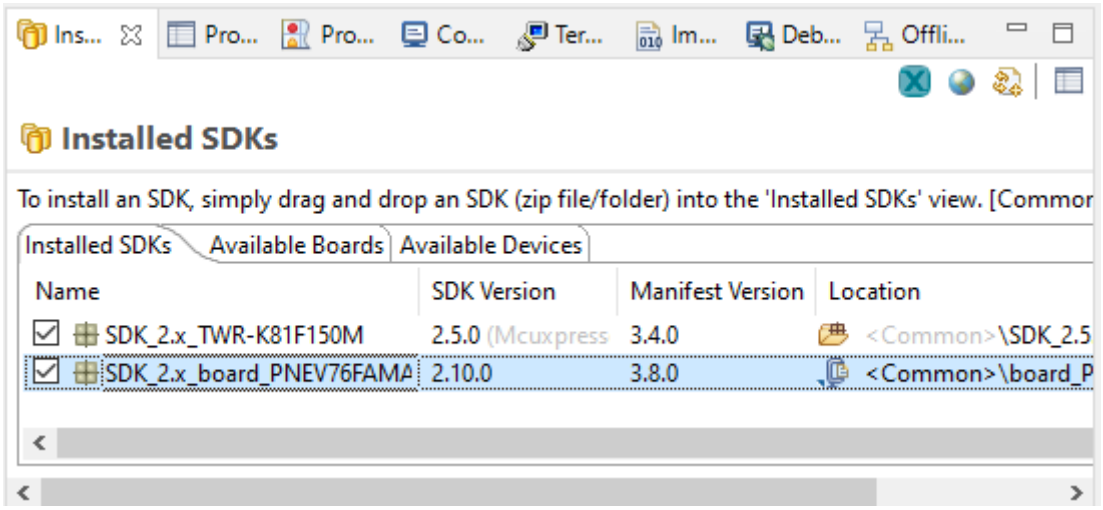

Figure 43. Installed PN76 SDK

9.1.2 Import SDK example

After the successful SDK installation, you can import examples from it by clicking "Import SDK example(s)..." in the Quickstart panel or by clicking "Import examples from SDK." in the Project Explorer.

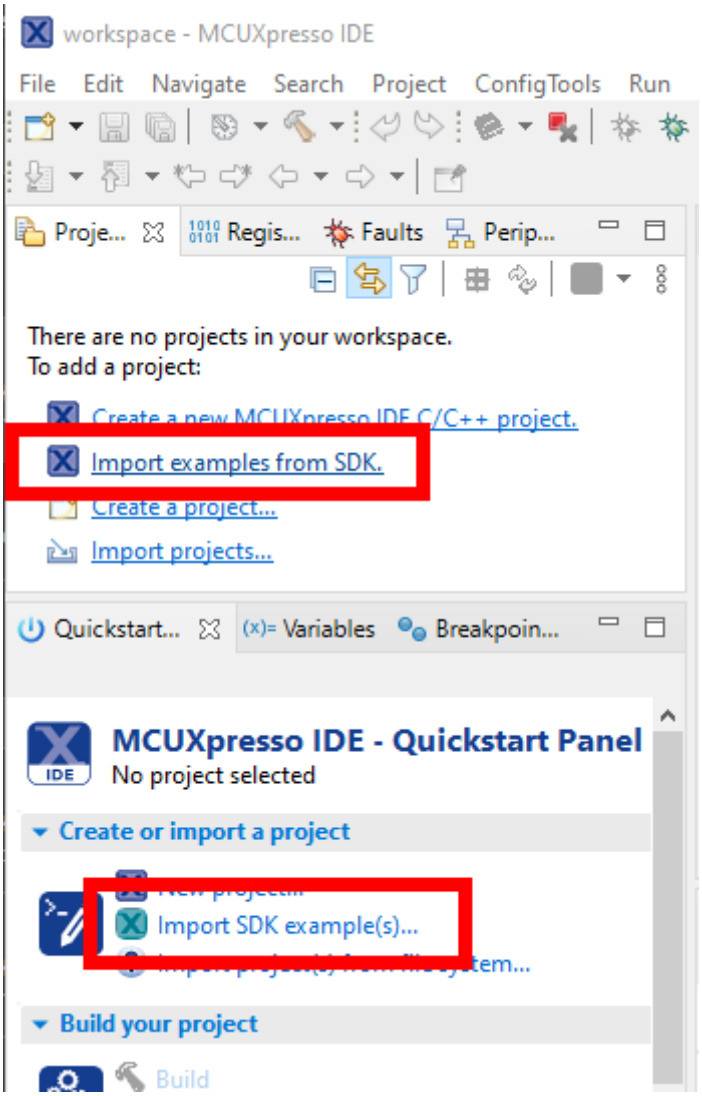

Figure 44. Import SDK Examples

In the following window, the installed SDKs are listed. Choose PN76 → PN7642, and click on the board image. The button "Next >" is now available. Press "Next >" to see a list of available examples.

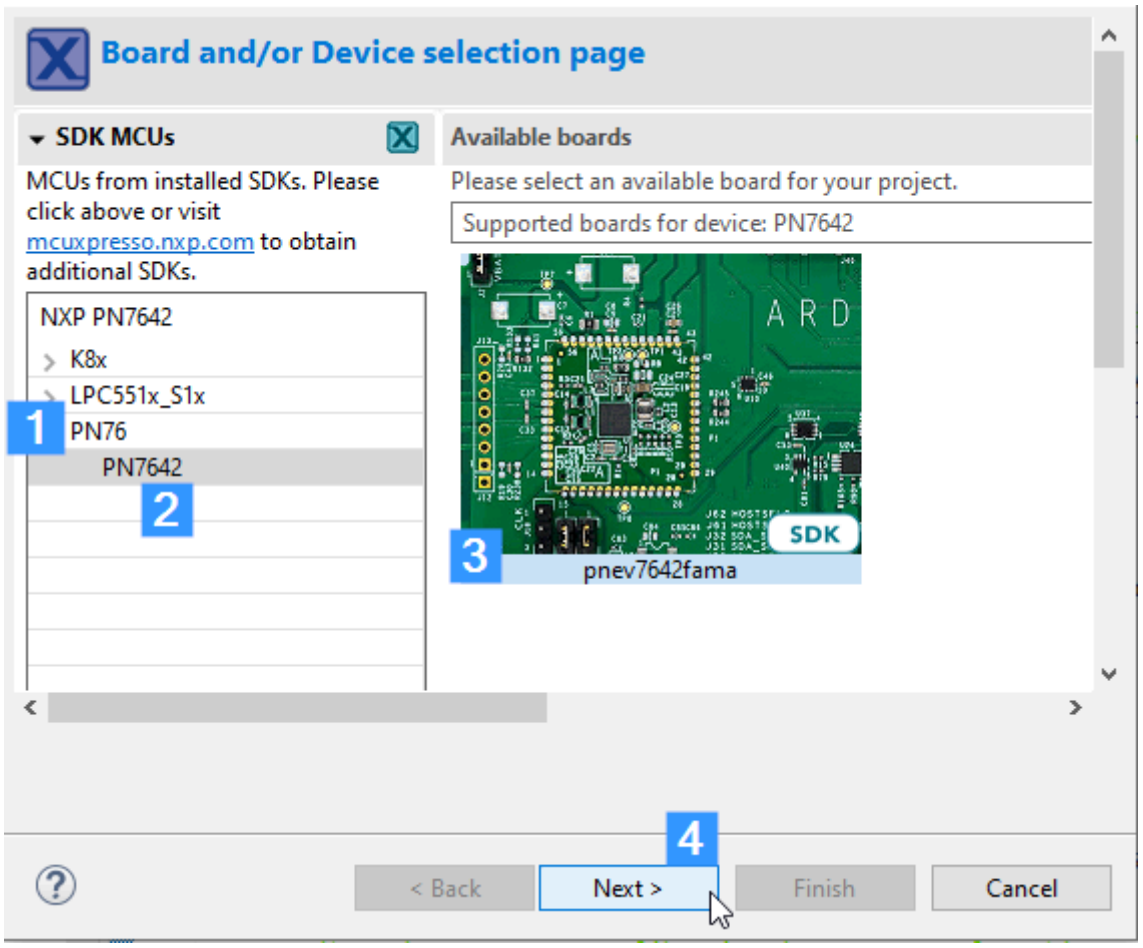

Figure 45. Choose PN76 SDK

Choose the specific examples which suits your needs. For the instructions shown in Section 9.1.3, the 'led_blinky' example is used.

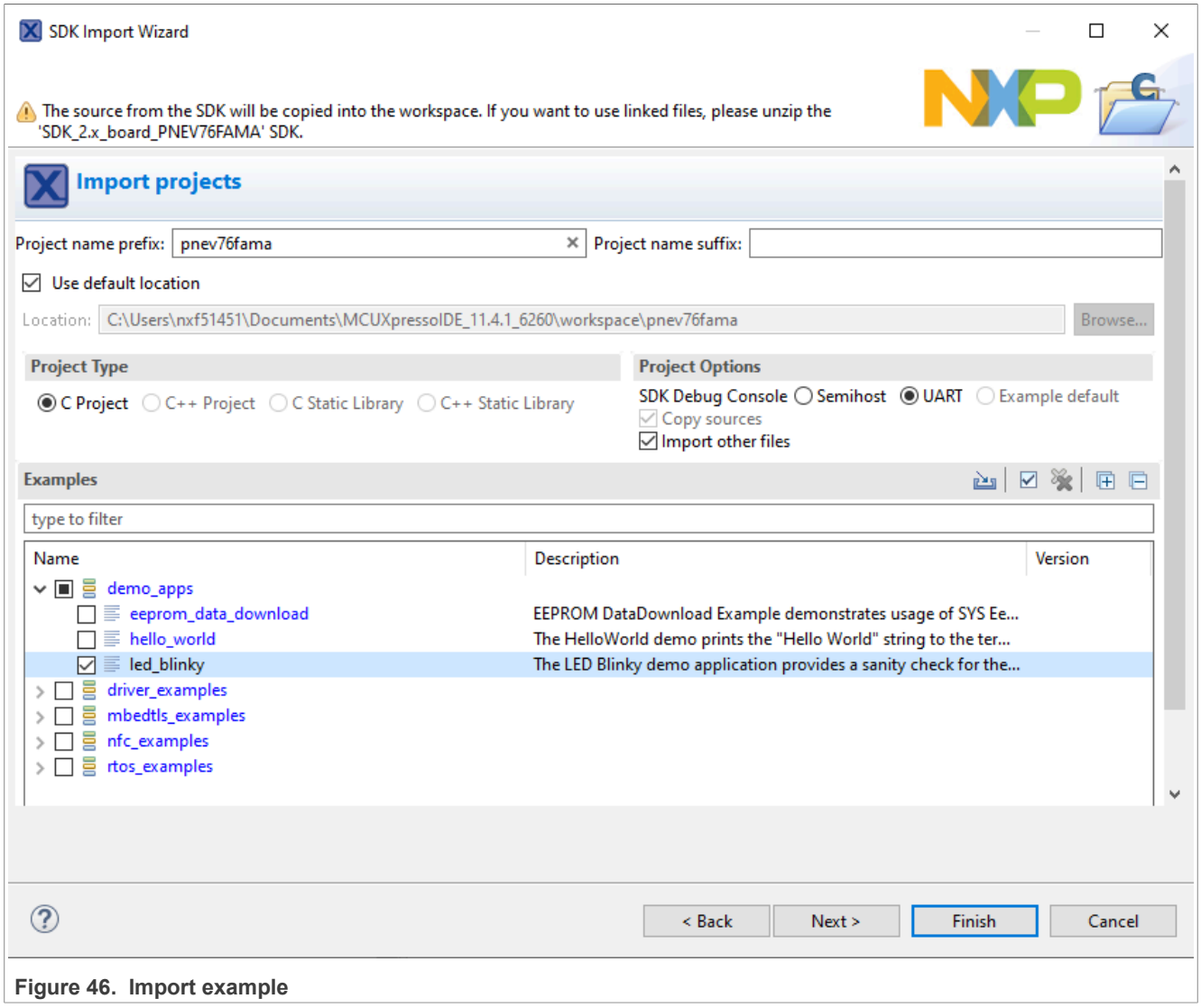

Figure 46. Import example

Note: *By default, "UART" is selected at the Project Options, which is the SEGGER RTT UART. If you want to have the serial output in the MCUXpresso console, check "Semihost" instead of "UART".*

By clicking "Finish" after choosing the example you want to import, it appears in your Project Explorer.

9.1.3 Build example

Note: *For the example shown in this section, the 'led_blinky' example is used.*

Every example has two build configurations, "Debug" and "Release".

The two build configurations serve different purpose. While "Release" is targeted to run standalone on the MCU and optimized in size for the option, "Debug" will disable optimization and also enable (via Preprocessor defines) the debug output. For development and debugging purpose, we recommend to chose "Debug". This ensures that stepping through the code will not result in optimized out jumps and hard to follow program flows.

Make sure to build your project in "Release" if you let it run without a debugger connected. Else it can result in unwanted, unpredictable behavior or even hard faults.

To build the project in the desired preconfigured option, choose either or at the build button at top:

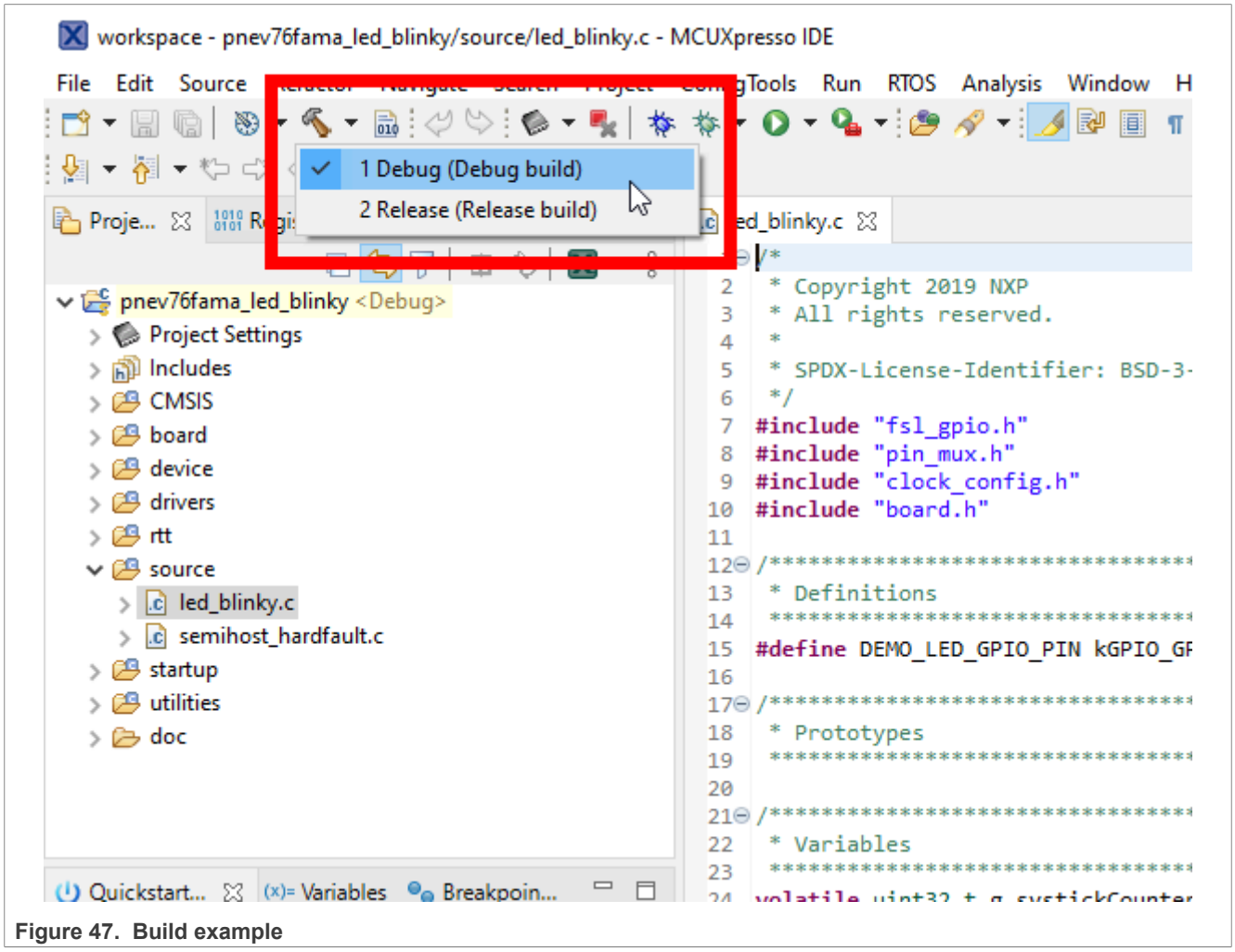

Figure 47. Build example

You can also choose the active build configuration by right-click on the project → Build Configurations → Set Active → Debug/Release.

After the build has finished, you will see 'Build Finished …' in the console.

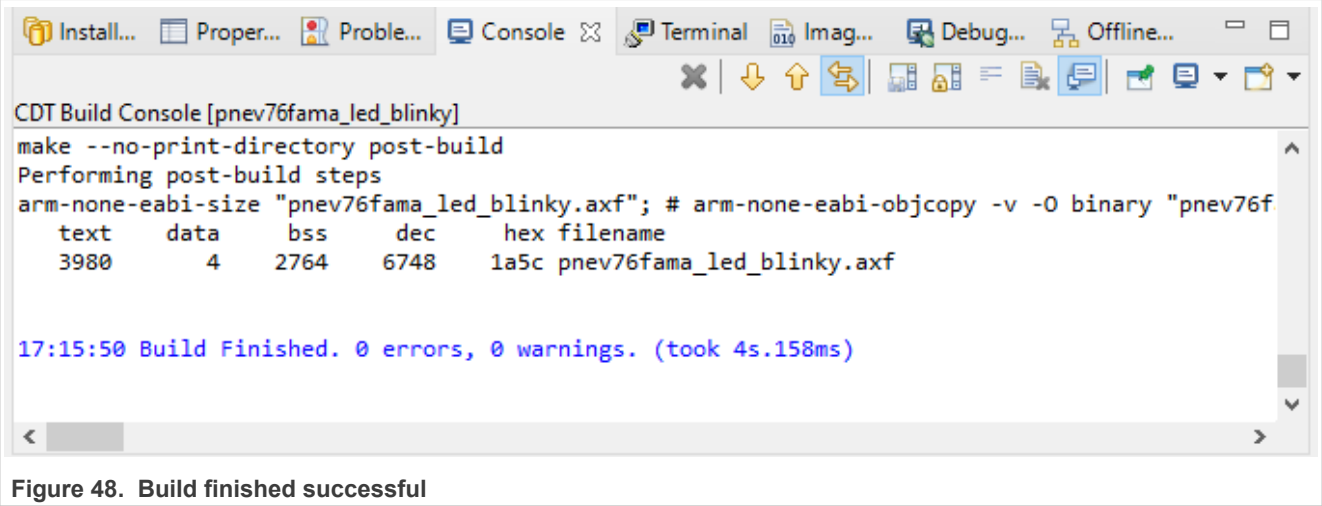

Figure 48. Build finished successful

Now, you can either flash or debug the build project.

9.1.3.1 Flash example

If you build your example in "Release" configuration, you can now flash it with the GUI Flash Tool by clicking the "GUI Flash Tool" button at the top toolbar:

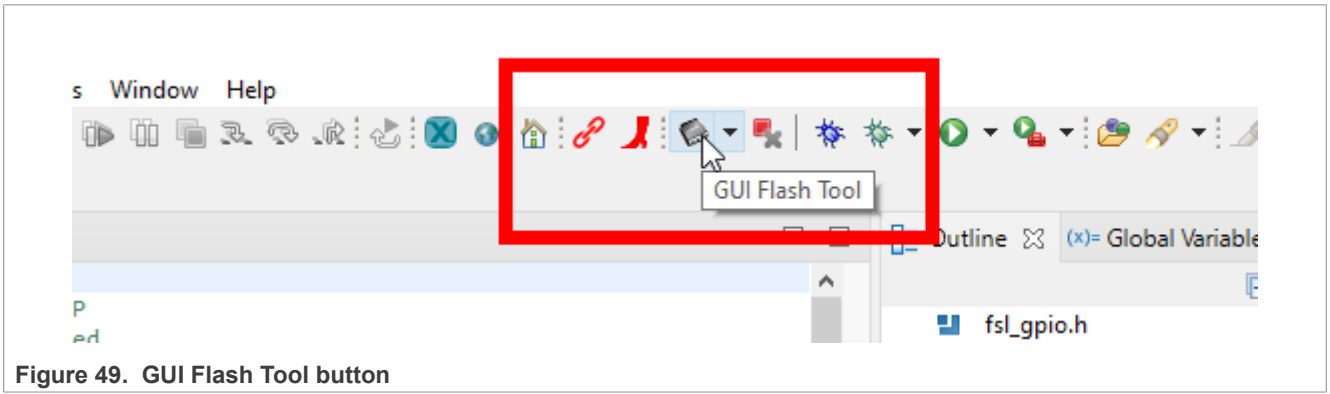

Figure 49. GUI Flash Tool button

MCUXpresso searches available programmers like a LPC-Link2 or SEGGER J-Link now. In case multiple probes are detected, choose the one connected to the NFC-Debug port on the PNEV7642A Rev-B development board and click "OK".

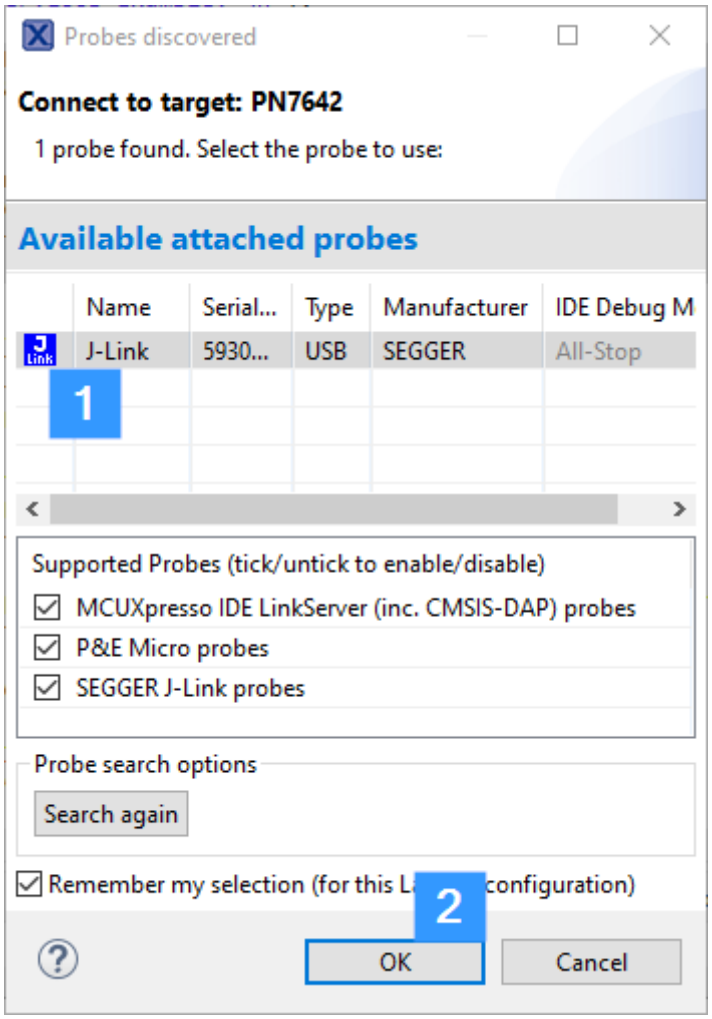

Figure 50. Choose available probes

The GUI Flash Tool now lists several Options how, with and where to program the flash space of the PN76. It should be preconfigured and you do not have to change anything. Click "Run..." at the bottom to start flashing the PN76.

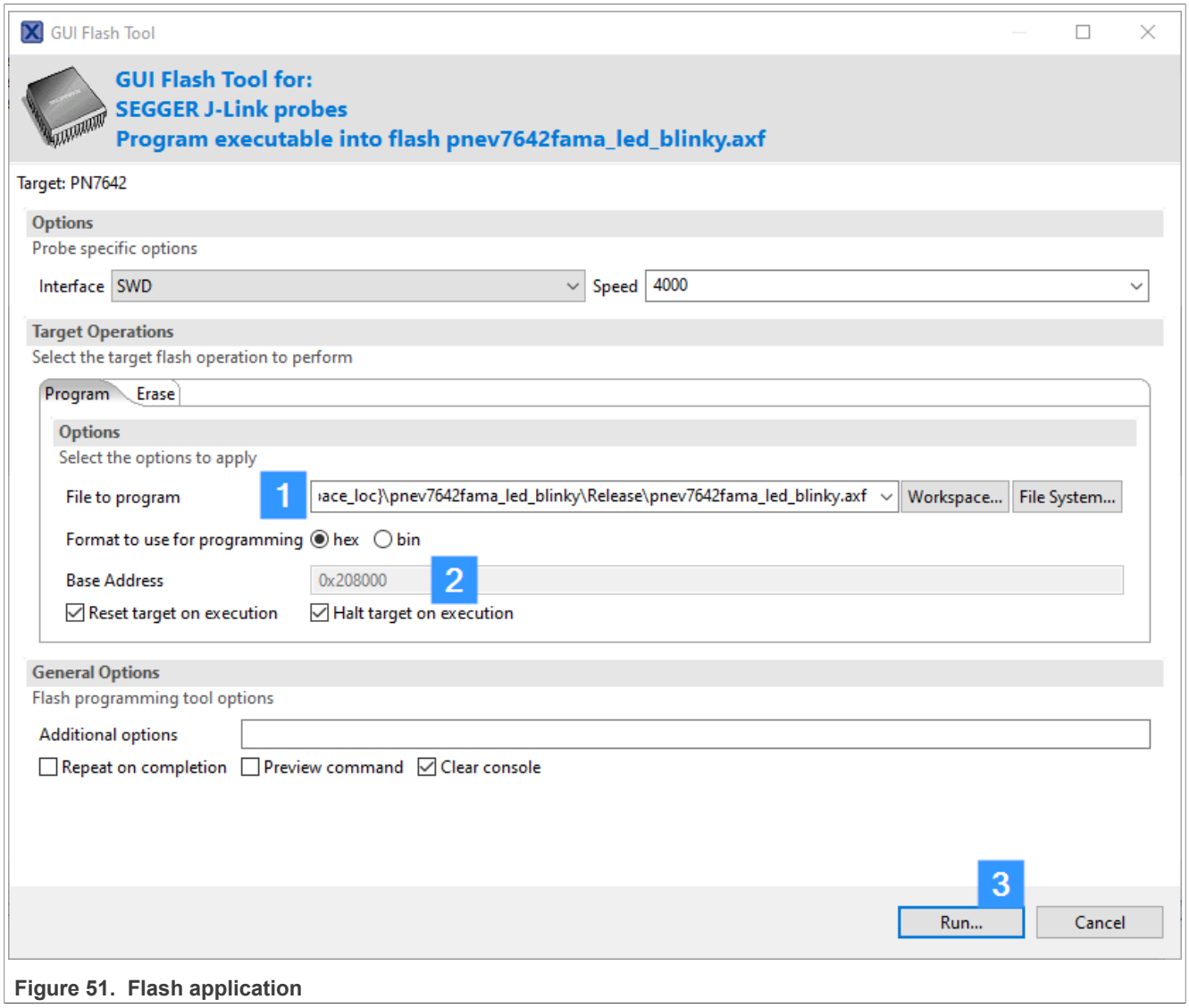

Figure 51. Flash application

In case you want to flash another application you build you can do so by browsing to the .hex or .bin file at (see "1"). The base address (see "2") is the start address of the user application space and shall not be changed.

Press "Run..." (see "3") to start the programming.

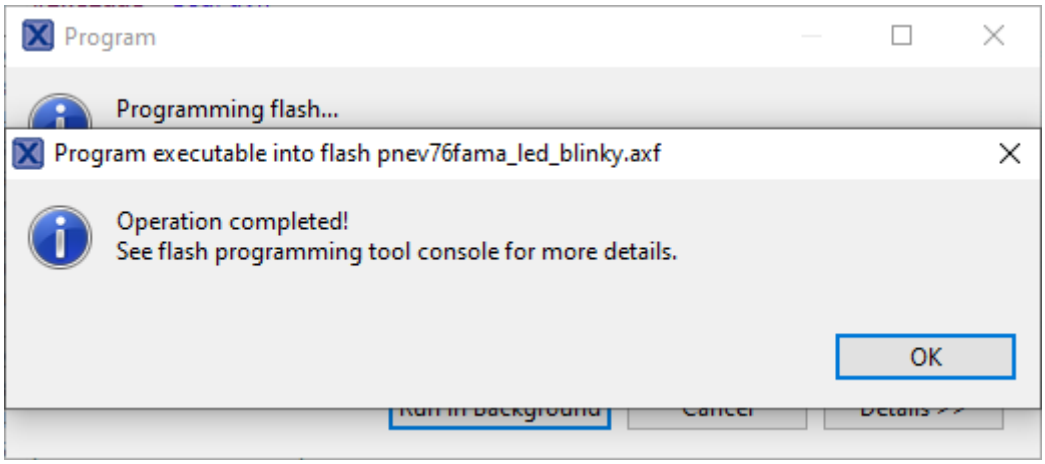

Figure 52. Flash successful

9.1.3.2 Debug example

If you build your example in "Debug" configuration, start debugging by clicking "Start debugging..." in the top toolbar.

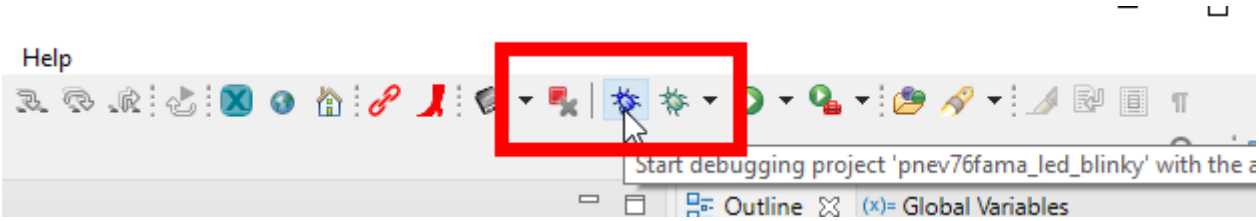

Figure 53. Start debugging

After the application has been flashed, the debugger halts automatically at the first line in the main() function.

With the buttons in the toolbar, you can step through the program or let it run freely by pressing "Resume (F8)".

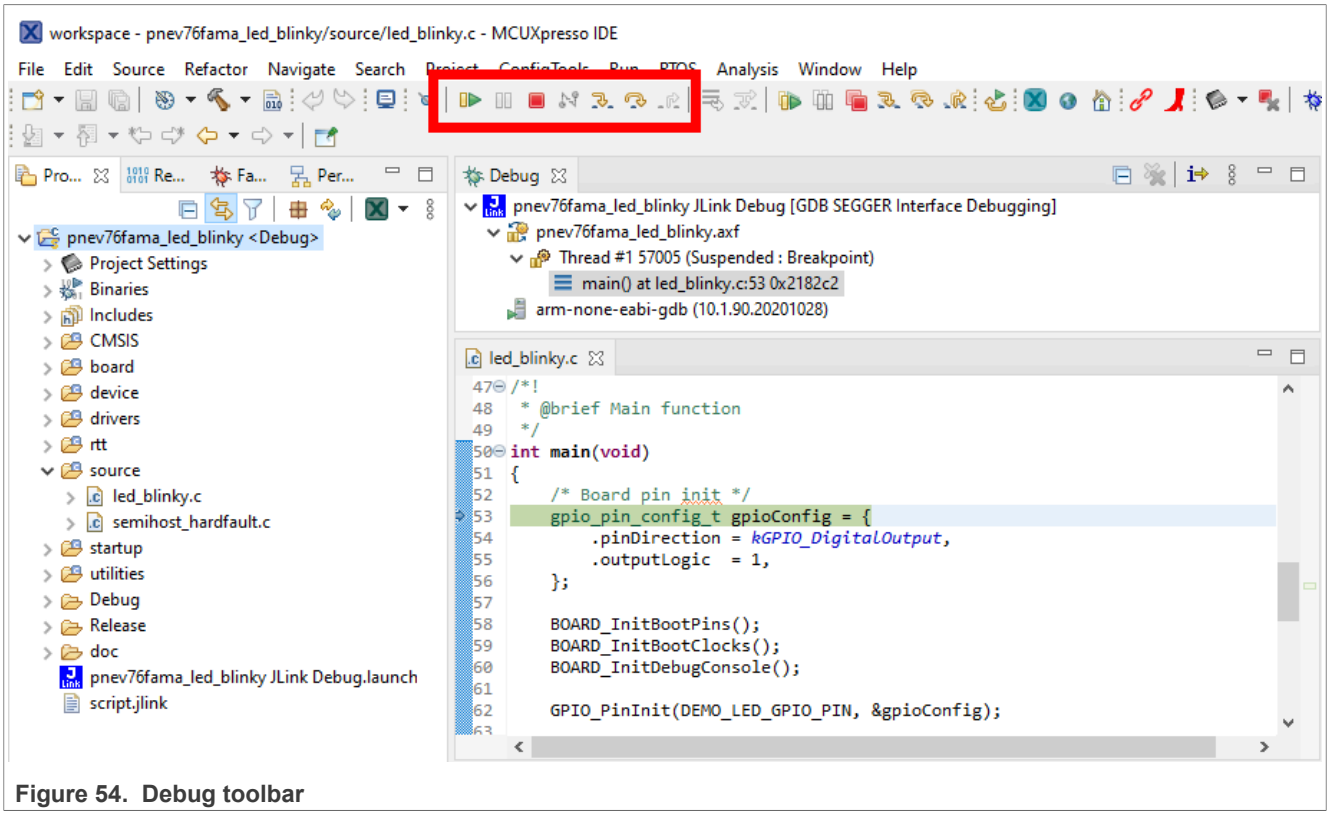

Figure 54. Debug toolbar

9.1.4 Bin file generation

To run an application standalone on the chip, it must be compiled in the release. To flash applications in mass-storage mode, you need a binary file. In case the binary (.bin) file is not generated automatically, you can do it manually by doing following:

1. Build your target application in release.
2. Open release folder in MCUXpresso.
3. Right-click on the *.axf file.
4. Choose 'Binary Utilities' → 'Create binary' in the menu.
5. The .bin should appear.

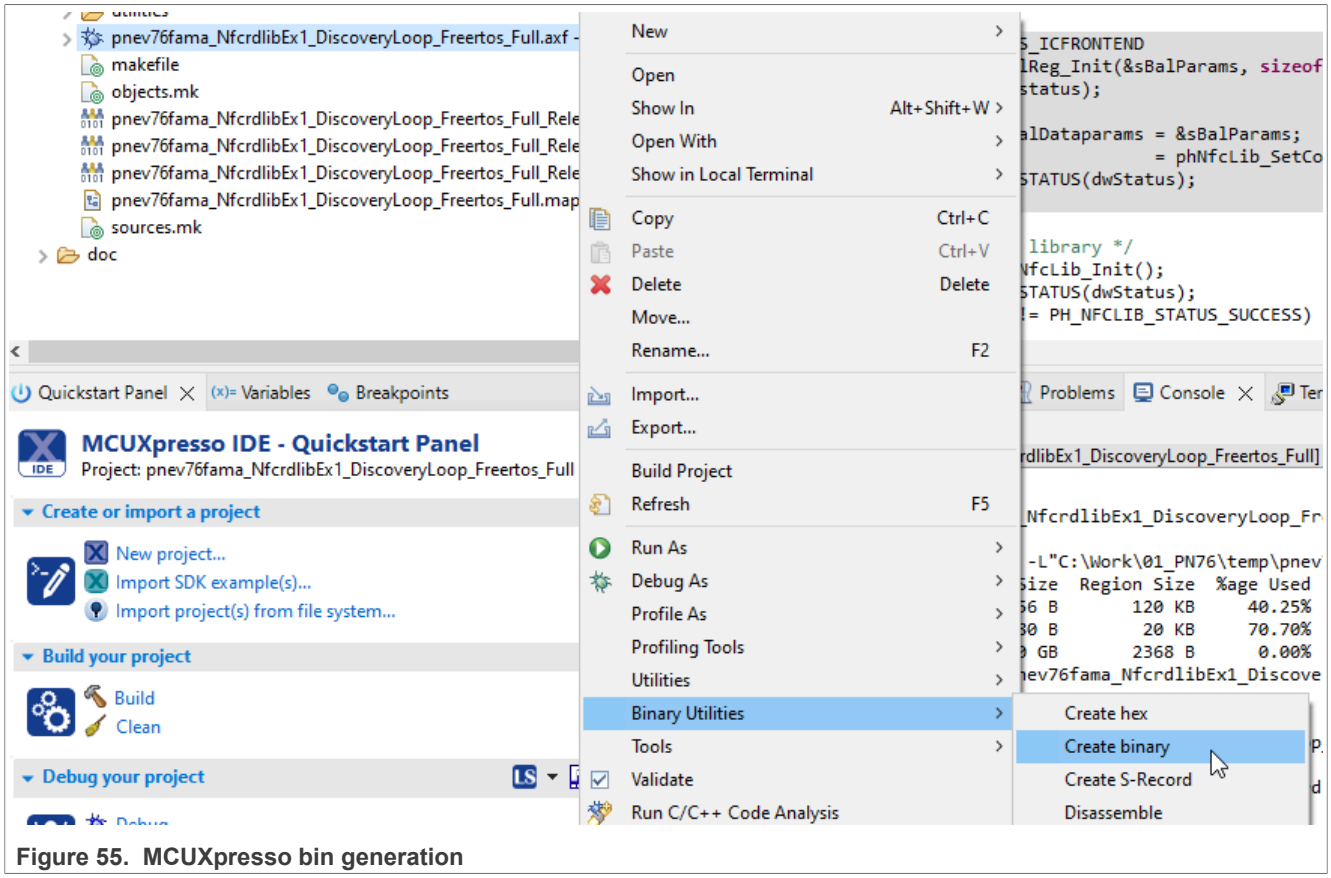

Figure 55. MCUXpresso bin generation

10 NFC Cockpit

NFC Cockpit is an intuitive graphical user interface that lets you configure and adapt IC settings without writing a single line of software code. From version v07.01.00 onwards, the PN7642 is supported as well. You can download the NFC Cockpit at the official NXP website (see [18]).

This chapter describes how to configure, set up, and use the NFC Cockpit with the PNEV7642A board.

Version limitations

As the PN7642 firmware itself evolves, same as the NFC Cockpit, some version combinations are not working. Check the Firmware Overview and Table 4 regarding valid combinations.

While the NFC Cockpit GUI supports all NFC Cockpit applications on the PN7642, the NFC Cockpit application on the PN7642 has to match the firmware on the PN7642.

NFC Cockpit is using VCOM. Users have to set the host interface to use USB. The NFC Cockpit counterpart integration on the PN7642 is also available as source code (see [19]).

This VCOM library can be integrated in a user application to provide the full NFC cockpit experience for configuration of the device.

Firmware Update

NFC Cockpit is checking the firmware of the connected device and informs the user to update to a later version in case a newer version has been found. For the PN7642, this is not always recommended. Refer to Section 5 "Firmware overview" and Section 5.1 before performing a firmware update.

10.1 Install NFC Cockpit

Download the latest version of NFC Cockpit at the NXP website. Execute the installer and follow the instructions. After the successful installation go to the installation folder.

Within the installation folder, you find a couple of folders. Two of them are from particular interest to progress further.

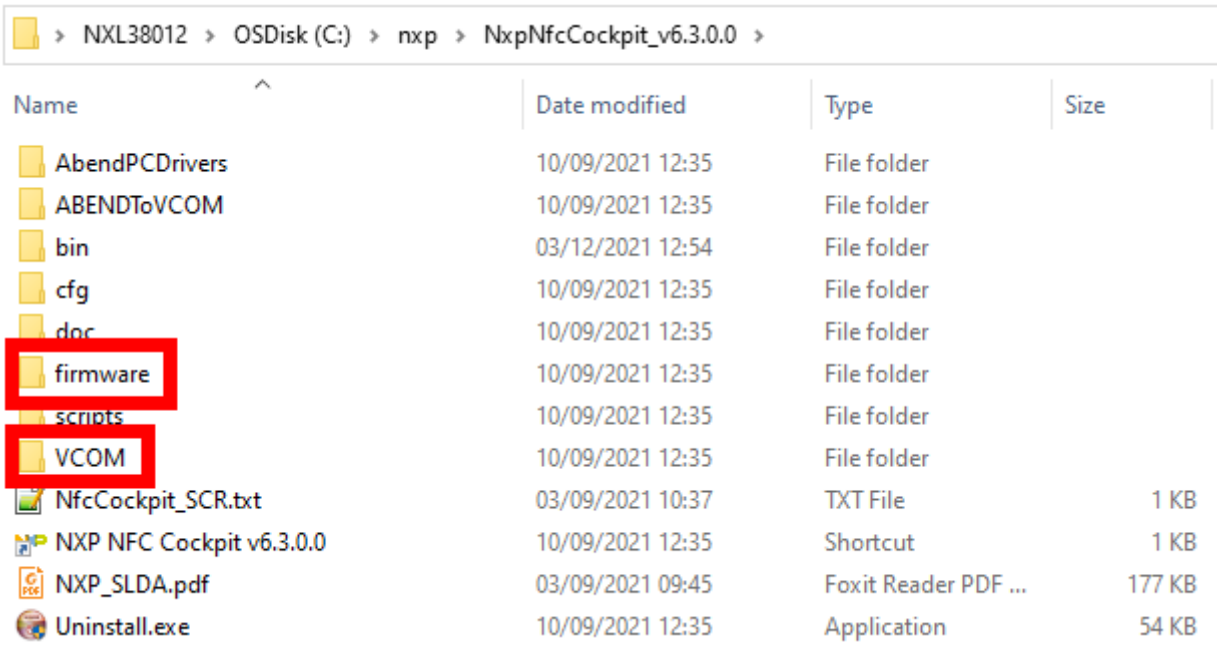

Figure 56. NFC cockpit folders

Table 16. NFC Cockpit folder content

Folder	Content
firmware → PN7642	'NxpNfcCockpit_05_03_00_Flash.bin', which is the PN7642 application for NFC Cockpit. This application has to match the PN7642s firmware. Check the compatibility table in Section 5.
VCOM	'install_PN76xx_vcom.bat', which installs the proprietary PN7642 VCOM driver.

10.2 Install VCOM driver

The PN7642 makes use of a proprietary VCOM driver. To install this, execute the 'install_PN76xx_vcom.bat' from your NFC Cockpit installation folder → VCOM.

Depending on the security policy of your device, maybe an execution with administrator rights is necessary (right click → run as administrator).

10.3 Flash NFC Cockpit firmware

Per default, the PNEV7642A is not flashed with the NFC Cockpit firmware, but with the NFC Discovery Loop Example application. This application will not enumerate a VCOM port and cannot communicate with the NFC Cockpit.

To flash the NFC Cockpit firmware, no external debugger is necessary. The PN7642 has a USB Mass Storage Mode where we can drag and drop the firmware to install.

Note: *To be able to put the device into mass-storage mode, make sure you set the host-interface to USB (see Section 7.1.4.1 "Host interface selection").*

10.3.1 Bring to mass storage mode

To bring the PN76 development board into mass storage mode, the user must bring the DWL_REQ pin HIGH while resetting the board.

Follow the instructions below:

1. Press and hold SW3 "NFC_VEN".
   a. Found in the upper left corner. A small white pushbutton.
2. Press and hold SW2 "NFC_DWL_REQ".
   a. Found on the right side edge of the board. A small white pushbutton.
3. Release SW3 "NFC_VEN".
4. Release SW2 "NFC_DWL_REQ".

A new mass storage device will appear in your explorer, which holds two files *"CRP_00.BIN*" and *"CRPSTA_3.BIN*".

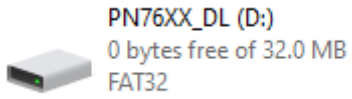

Figure 57. PN7642 mass storage device

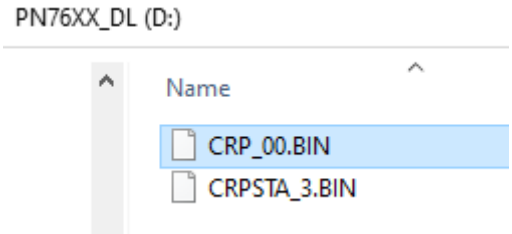

Figure 58. PN7642 content

10.3.2 Flash NFC Cockpit firmware

1. Open the PN76xx_DL mass storage device and delete the *"CRP_00.BIN"* file.

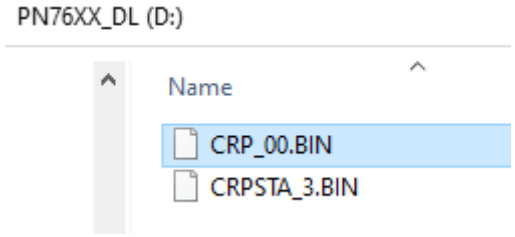

Figure 59. Mass storage content

2. Go to your NFC Cockpit installation folder → firmware → PN7642 → v1.00/v2.00, see README.txt, and copy the *"NxpNfcCockpit_05_03_00.bin"*[1] onto the mass storage device.

Note: *Check chapter Section 5 to find the matching NFC Cockpit application version to the PN7642 firmware version in use.*

For the PN7642, use the PN7642 folder → firmware and read the "*README.txt*" to check which firmware versions are suitable.

Figure 60. NFC Cockpit firmware

3. The PN76xx_DL automatically restarts and a PN76XX VCOM device should appear in your device manager. If no VCOM appears and instead the mass storage device PN76xx_DL show up again, make sure DWL_REQ is low and reset the device again by pressing NFC_VEN.

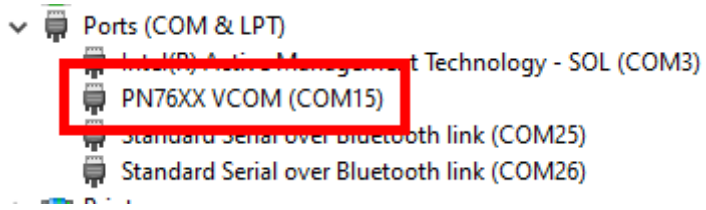

Figure 61. PN7642 VCOM device

The PN76 has been successfully flashed with the NFC Cockpit firmware and can now be used with the NFC Cockpit application.

Depending on your system and previously installed drivers, there is a possibility that the PN76 VCOM port is displayed within another device category. This usually is no problem and the NFC Cockpit will function as expected. In rare cases, it might be necessary to look closer to similar drivers and de-install/install them manually.

1 The naming and/or version might be different depending on which NFC Cockpit version is used.

10.4 NFC Cockpit usage

The NFC Cockpit is a powerful GUI tool to control any NXP NFC chips. This chapter only explains a quick check if it is working. For further usage and details on its capability, read the NFC Cockpit user manual.

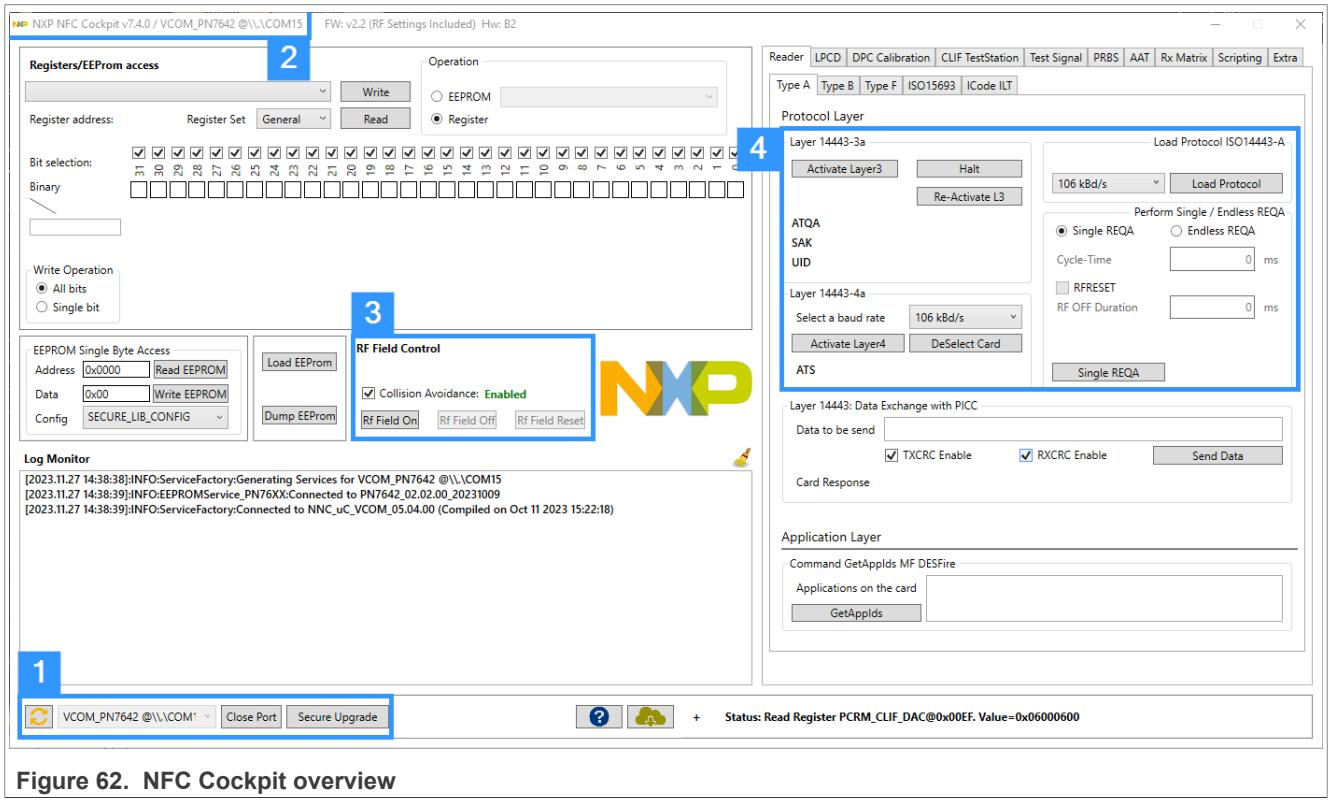

Figure 62. NFC Cockpit overview

Highlighted in the above image are our main areas of interest.

1. The VCOM port, which we are currently connected to.
2. NFC Cockpit version and firmware version of the connected device. In this case FW v2.2 of the PN7642
3. RF Field control. To turn the RF-Field on and off as well as doing an RF-Field Reset (Off → On).
4. Protocol Layer control. To activate a Type A tag.

For a quick sanity check, follow the below steps:

1. Place a Type-A Tag on the antenna of the PNEV7642A Rev-B development board. This can be, for example, a MIFARE Ultralight, MIFARE DESFire, MIFARE Classic, or any other Type-A tag.
2. Press "RF Field On" in the RF Field Control area (3).
3. Press "Load Protocol" in the Protocol Layer Area (4). This loads all necessary Type-A-related parameters at the PN7642.
4. Press "Activate Layer3" in the Protocol Layer Area (4). It performs a Layer-3 (ISO/IEC14443-3) activation, including its anticollision.
   a. If the activation was successful the ATQA, SAK and UID are displayed.

11 Checklist

A checklist to see if you fulfill all prerequisites to start evaluating, developing, and working with the PNEV7642.

1. PNEV7642 correctly powered up, either through an external power supply or USB.
   a. See Section 7.1.1 "Power supply".
2. Debugger connected.
   a. Can be an external debugger like SEGGER J-Link, MCU-Link, or LPC-Link2.
   b. Or the onboard OpenSDA.
   c. See Section 7.1.2 "Debug interfaces".
3. MCUXpresso installed.
   a. See Section 9.1.1 "MCUXpresso IDE".
4. PN7642 SDK installed.
   a. See Section 9.1.1.2 "Install SDK".
5. PNEV7642 jumper settings according to the desired use case.
   a. Especially, if you want to use a host-interface (USB, I2C, SPI, …) the red switches have to be set accordingly.
   b. See Section 7.1.4 "Jumper settings".
6. Firmware version of the chip matches the installed MCUXpresso SDK.
   a. C100 comes by default with FW v01.00 while C101 comes with FW v02.00. The same application will not work on both as the firmware versions are incompatible with each other. The application has to be compiled using the correct SDK.
   b. See Section 5 "Firmware overview".

12 Troubleshooting

A short list of the most common issues and its solutions. The application note AN13936 frequently asked questions contains some additional answers to unexpected behaviour.

Table 17. Troubleshooting

Cause	Comment
IOREF	Depending on the usage of the board (standalone, follower or leader) the jumper IOREF has to be set correctly. See Section 7.1.4.2 "I/O Reference (J23)".
HIF selection	Make sure to set the HIF switches according to the host-interface of your choice (USB, I2C, SPI, etc.). See Section 7.1.4.1 "Host interface selection".
HIF not working	The host interface is determined by either the host select switches (red switches) or via the Arduino Leader (bottom) interface. To ensure compatibility with different voltage leves there are level-shifters inbetween. If no Arduino Leader is used the inputs of those level shifters are floating, which could cause issues. If you encounter issues that, e.g., USB is not populated and working you can try to force AD_NFC_HOSTSEL0/1/2 to the voltage leves necessary for your chosen host-interface. See Section 7.1.4.1.1.
Power supply	Be careful by powering the board. By using an external power supply, make sure to have set the power jack settings correct. Section 7.1.1.2 "Power jack". If you power the board via USB, make sure that the USB port is good enough to supply the PNEV7642 development board. Usually an active USB-Hub works. While a notebook board might not be sufficient. See Section 7.1.1.1 "Power supply selection".
Firmware version	Make sure that the firmware version of your board is the right one for the used MCUXpresso SDK. In case of doubt, it is good practice to update the firmware. This can be easily done with the "check_nxpfw_update" example of the MCUXpresso SDK. See Section 5 "Firmware overview".
Debugger - Cannot connect to core	This can have many reasons. The most popular are insufficient power, the PN76xx being in reset (check VEN), in mass-storage mode or deep-sleep. Make sure that nothing of these options is the case. In case you are enabled OpenSDA on the development board the NFC-DEBUG "J21" is not operational because you had to resolder some resistors.

13 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 18. Characteristics

PN	RF Technology	(a) Freq Ranges (EU)	(b) Max Transmitted Power
OM27642DB	Near Field Communication	13.56 MHz ± 7 kHz	-11 dBm
OM27640DB	Near Field Communication	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 PNEV76FAMA and PNEV7642A development board, conforms to the Radio Equipment Directive 2014/53/EU.

The full EU and the UK Declaration of Conformity for this apparatus can be found at this location: https://www.nxp.com/products/:OM27642.

14 References

[1] Data sheet – PN7642 – Single-chip solution with high-performance NFC reader, customizable MCU, and security toolbox (link)
[2] Application note – AN13467 – PN76 family antenna design guide (link)
[3] Application note – AN13925 – PN76 EEPROM configuration handling (link)
[4] Application note – AN13996 – How to use the low-power features of the PN76 family NFC controller (link)
[5] Application note – AN14060 – How to use the PN76 family cryptographic features (link)
[6] Application note – AN13720 – PN7642 Secure Key Mode demo application (link)
[7] Application note – AN13936 – PN7642 frequently asked questions (link)
[8] Application note – AN14540 – Firmware Update on PN7642 (link)
[9] Application note – AN14518 – Crystal Oscillator Design Guide (link)
[10] Webpage – PN7642 – Single-chip solution with high-performance NFC reader, customizable MCU, and security toolbox (link)
[11] Application note – AN13898 – OpenSDA on PN76 development boards (link)
[12] Release notes – RN00257 – PN7642 firmware release notes (link)
[13] Design resources – Schematics for PN76xx RFP-board (link)
[14] Design resources – Schematics for PNEV7642MB module board (link)
[15] Webpage – LPCXpresso55S16 Development Board (link)
[16] Webpage – MCUXpresso Integrated Development Environment (IDE) (link)
[17] Software – LPC55S16 Host Software (link)
[18] Webpage – NFC Cockpit Configuration Tool for NFC ICs (link)
[19] Software – PN76 VCOM LibNncFw (link)
[20] Webpage – OM27642 – Development Kit for PN7642 Secure, Compact NFC Controller with Programmable Memory and Crypto Acceleration (link)

15 Revision history

Table 19. Revision history

Document ID	Release date	Description
AN13134 v.3.0	18 March 2025	Editorial changes. Restructured document for better user experience. • Section 1 "Introduction": updated. • Section 1.1 "Quick map": added. • Section 2 "Documentation overview": updated. • Section 3 "Quick start with USB as supply": updated. • Removed section "Firmware release notes" and added information to Section 5 "Firmware overview". • Section 5.1 "Firmware update": added. • Section 9.1.1.2 "Install SDK": updated. • Section 9.1.2 "Import SDK example": updated. • Section 9.1.3 "Build example": updated. • Section 10 "NFC Cockpit": updated. • Section 11 "Checklist": updated. • Section 14 "References": updated.
AN13134 v.2.7	6 December 2023	• Added Section 3 "Quick start with USB as supply". • Added Section 7.2 "PNEV7642MA". • Editorial changes: – Consistent naming of PN7642. – Restructuring of chapters to have a better reading flow.
AN13134 v.2.6	18 October 2023	• Added section Setup without DC-DC. • Added section Firmware overview: – Added multiple references to this chapter in applicable chapters. • Added section Documentation overview. • Updated section NFC Cockpit.
AN13134 v.2.5	2 May 2023	• Deleted old boards and environments not relevant for the official released version. • Added section Checklist. • Added section Troubleshooting.
AN13134 v.2.4	13 March 2023	• Added section TDAEV8035. • Added section MCUXpresso SDK. • Updated section Radio Equipment Directive (RED). • Security status changed to public.
AN13134 v.2.3	1 December 2022	• Enhanced section Software. • Updated section Radio Equipment Directive (RED). • Added section PNEV7642A Rev-B. • Editorial changes and optimize picture size. • Updated the terms "Master/Slave" to "Controller/Target" to align with the recommendation of the NXP - I2C standards organization.
AN13134 v.2.2	18 August 2022	• Section IOREF added.

Table 19. Revision history...*continued*

Document ID	Release date	Description
AN13134 v.2.1	9 August 2022	• Jumper settings added. • NFC Cockpit version compatibility list added. • New PNEV7642A board added. • Deprecated status to Eclipse-related topics added. • Binary generation topic to MCUXpresso added. • Section Radio Equipment Directive (RED) added.
AN13134 v.2.0	15 February 2022	• Rearranged chapters for better reading flow. • Section Host software (LPC55S16) added to explain secure firmware download example. • Section Software added to give an overview of existing software packages.
AN13134 v.1.6	18 January 2022	Editorial changes.
AN13134 v.1.5	9 December 2021	• Added NFC Cockpit chapter. • Enhanced system overview.
AN13134 v.1.4	1 December 2021	• Adde system overview. • Changed to new revision D PNEV76FAMA board. • Added important jumper settings. • Added MCUXpresso chapter.
AN13134 v.1.3	21 July 2021	Change from PN5190 RF Performance board to PNEV76FAMA board.
AN13134 v.1.2	22 April 2021	• PN76 wording changed. • PN76 family software package contents added.
AN13134 v.1.1	19 March 2021	Change to GCC version.
AN13134 v.1.0	11 February 2021	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.

Amazon Web Services, AWS, the Powered by AWS logo, and FreeRTOS — are trademarks of Amazon.com, Inc. or its affiliates.

AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile — are trademarks and/or registered trademarks of Arm Limited (or its subsidiaries or affiliates) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved.

DESFire — is a trademark of NXP B.V.

EdgeVerse — is a trademark of NXP B.V.

J-Link — is a trademark of SEGGER Microcontroller GmbH.

MIFARE — is a trademark of NXP B.V.

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

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

Tables

Figures

Contents