AN13134

PN76 family evaluation board quick start guide

Rev. 2.7 — 6 December 2023   **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, which provides an easy evaluation of the features and functions of the PN7642.

It provides the first steps to operate the board and the setup of the development environment.

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.

AN13134

Application note

Rev. 2.7 — 6 December 2023

2 / 67

1.1 System overview

There are multiple setups to get started with the development board. You can choose any of them. Whatever fits your purpose, way of working and hardware at hand the best.

1.1.1 PN76 and external debugger

The following figure shall indicate how an evaluation setup with PNEV7642A and an external programmer can look like.
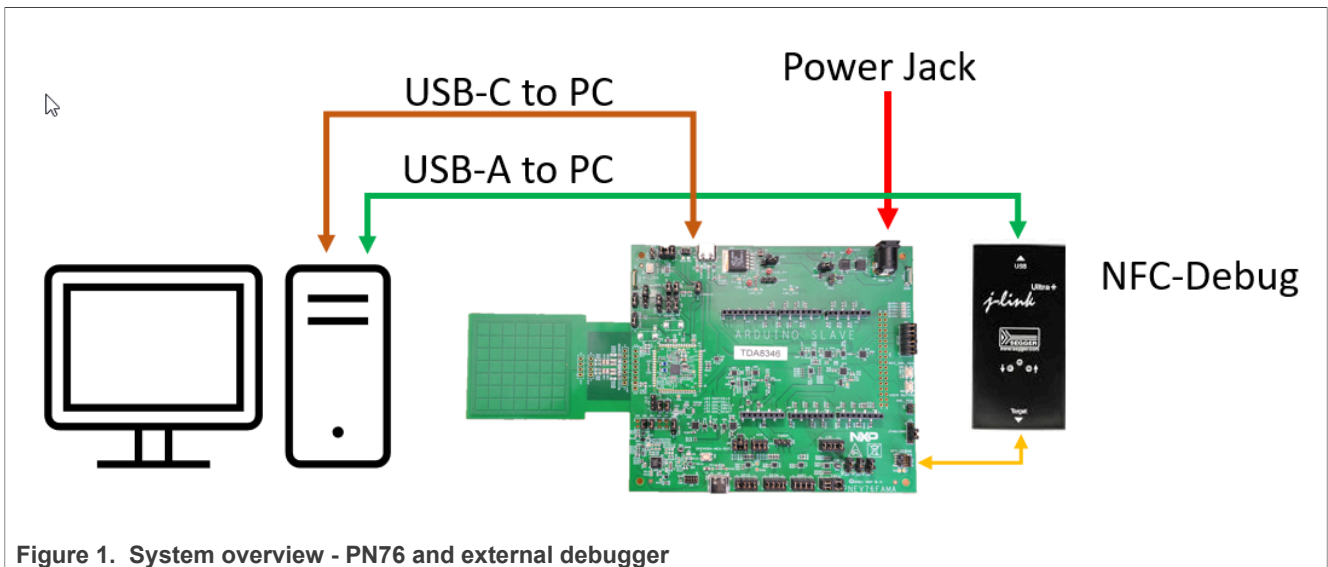
Figure 1. System overview - PN76 and external debugger
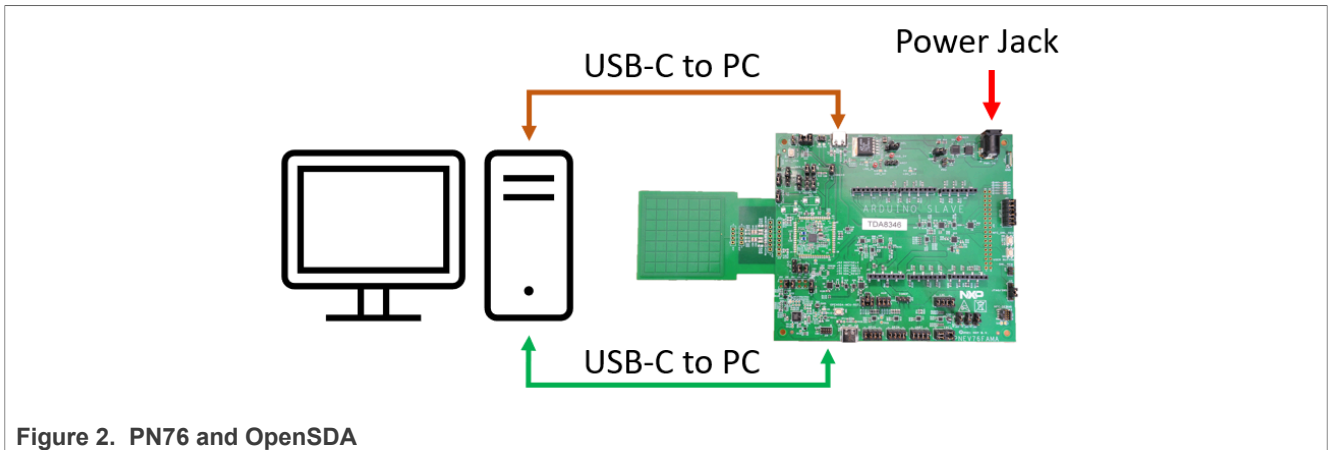
PNEV7642A connections:

- 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 proper power supply to supply the PNEV7642A board.
  – If your USB-C power is good enough (active USB Hub), you might power it directly via USB "J5" and not with the power jack.
- J21 "NFC-Debug" to connect a Debug Probe like a SEGGER J-Link, MCU-Link, or LPC-Link2. To flash and debug the PN76.

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 either be a LPC-Link2, 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.

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

3 / 67

1.1.2 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 the following application note: AN13898 OpenSDA on PN76 development boards.

Figure 2. 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.

1.1.3 PN76, LPC55, and external debugger

The following indicates how an evaluation setup with PNEV7642A, LPC55S16-EVK, and an external programmer can look like.
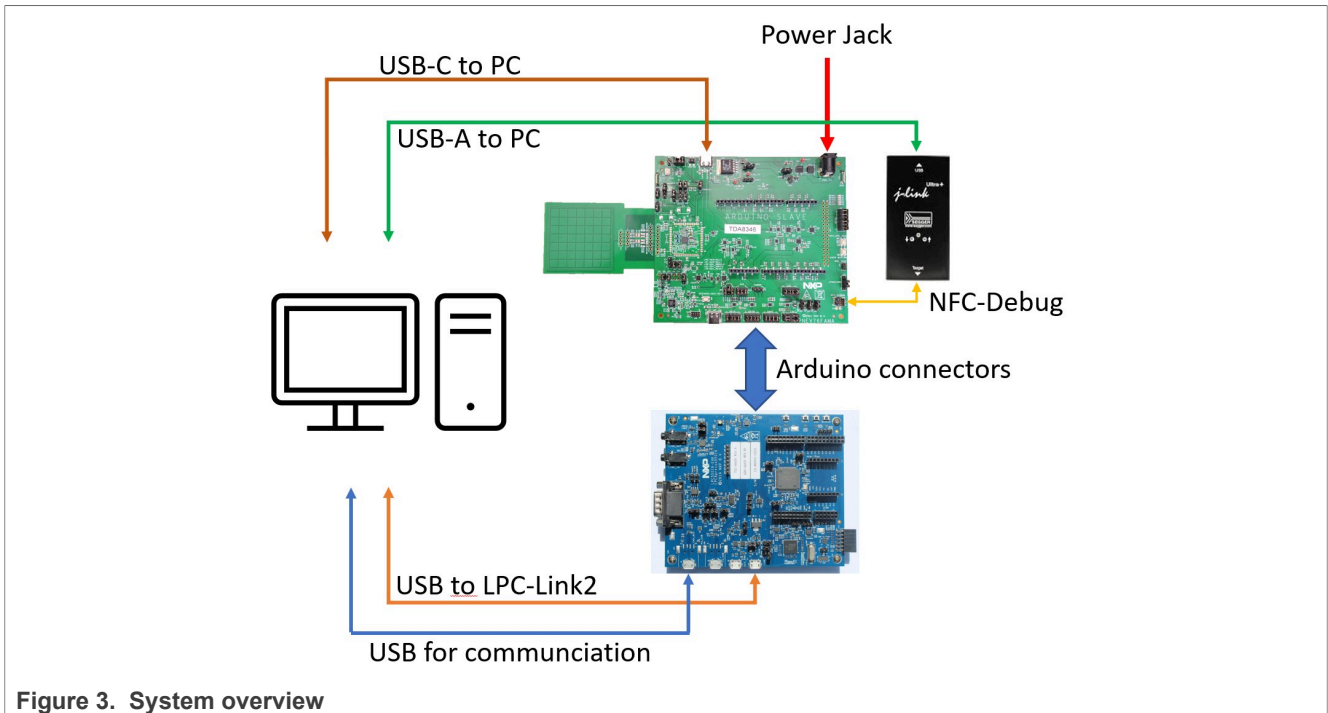
Figure 3. System overview

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.

1.2 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 will overwrite the settings while the other one, with "F", will not touch the settings.

SDK version

An SDK is bound to a certain firmware version. Make sure to use the right SDK version for the firmware version. This means that an SDK upgrade is necessary if a firmware upgrade is made.

This guarantees that the compiled application is running on the target firmware version.

NFC Cockpit version

The NFC Cockpit application within the installation folder of NFC Cockpit is compiled for a certain firmware version. That is why the PN7642 NFC Cockpit application of the newest NFC Cockpit will not run on an older firmware version (like v01.00).

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

Firmware update

In general, a firmware update shall be made with caution. As it brings implications and dependencies to update other tools as well. Before doing a firmware update, it is highly recommended to read the release notes.

Firmware backward compatibility

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

Before updating a firmware check the release notes.

Compatibility table

Table 4. Firmware compatibility

FW version	SDK version	NFC Cockpit	Comment
01.00	2.12.1	7.1.x.x	Standard firmware on C100 chips. DWL_REQ pin necessary to enter into the bootloader.
01.F0	2.12.1	7.1.x.x	
02.00	2.12.3	7.2.x.x	Standard firmware on C101 chips. DWL_REQ pin not necessary. Pin-less download active as default.
02.F0	2.12.3	7.2.x.x	
02.01	2.12.4	7.3.x.x	Firmware officially withdrawn due to backward compatibility issues.
02.F1	2.12.4	7.3.x.x	
02.02	2.12.5	7.4.x.x	Minor bug fixes. Backward compatible to FW v02.00.
02.F2	2.12.5	7.4.x.x	

AN13134
Application note

All information provided in this document is subject to legal disclaimers.

Rev. 2.7 — 6 December 2023

© 2023 NXP B.V. All rights reserved.

6 / 67

1.2.1 Firmware release notes

For the latest release notes, always check the FW API Guide within the SDK documents folder. Below is a table of the release notes FW 02.02.

Table 5. Firmware release notes

Release	Date	Notes
02.02.00	9 October 2023	Secure FW compatibility with v02.00.00 features same as that of v02.01.00.
02.01.00	1 September 2023	Secure FW bug fix and feature add release.
		- Fixed issues in card mode of operation.
		- Fixed an issue where in call to API *PN76_Sys_UpdateRfConfiguration()* and *PN76_Sys_RetrieveRfConfiguration()* when used for TX index *E_PN76_LOADRF_TX_ISO180003M3_TARI_9_44_ASK* and *E_PN76_LOADRF_TX_ISO180003M3_TARI_18_88_ASK*.
		- USB download mode is disabled when the pin-less download mode feature is enabled, even if USB_VBUS is present.
		- New APIs *PN76_Sys_Set_HIF_Timeout()* and *PN76_Sys_Get_HIF_Timeout()* added to update/get the timeout values for waiting for HIF commands in case of pinless download feature is enabled.
		- Implemented CT as a wake-up source during standby wake-up.
		- New API *PN76_Sys_Configure_TestBus_MultipleDigital()* for configuring multiple available digital test bus signals on selected pad configurations.
02.00.00	28 April 2023	Secure FW feature adds release.
		- Updated support for FreeRTOS kernel V10.5.0 from FreeRTOS kernel LTS Patch 2.
		- Added support for USB-CCID class suspend and resume operation for USB certification (chapter9 and electrical compliance).
		- New API *PN76_Sys_GetPartId()* provide information about the PN7642 IC part number.
		- New API *PN76_Sys_Hal_USB_Suspend()* provides functionality for a USB suspend operation and resume after suspend.
		- New API *PN76_Sys_OTPConfigs_EnableDwnldReqLessBoot()* provides for entering into download mode without DWL_REQ pin.
01.00.00	21 November 2022	FW production release.

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

7 / 67

1.3 Documentation overview

The PN7642 is a complex chip with the possibilities it offers. A Cortex-M33, contactless front end, hardware secure key store and hardware crypto support.

Below is an overview of documents for the PN7642:

- PN7642 data sheet
  - 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 as well as 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 is holding 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.

1.4 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 settings are optimized for the 45 mm x 45 mm antenna of the board RF performance board and can be changed by the user in case a customized antenna and matching network is used. The load RF configuration method allows to initialize multiple registers with an efficient single API call and allows to distinguish between transmit and receive configuration. 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 PN76_Sys_LoadRfConfiguration(…) considers this.

Some of these settings can or even **must** be adapted toward a new antenna design (e.g. the dynamic power control). All those design-specific settings should 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 might also be adapted for optimum performance of the chip.

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

2 Quick start with USB as supply

The following steps shall quickly enable you to work with the PN7642 chip and its development board. For further details about the development board, the development environment, the product support package, and other related topics, read through the whole document.

1. Unpack your PNEV7642A Rev-B development board.

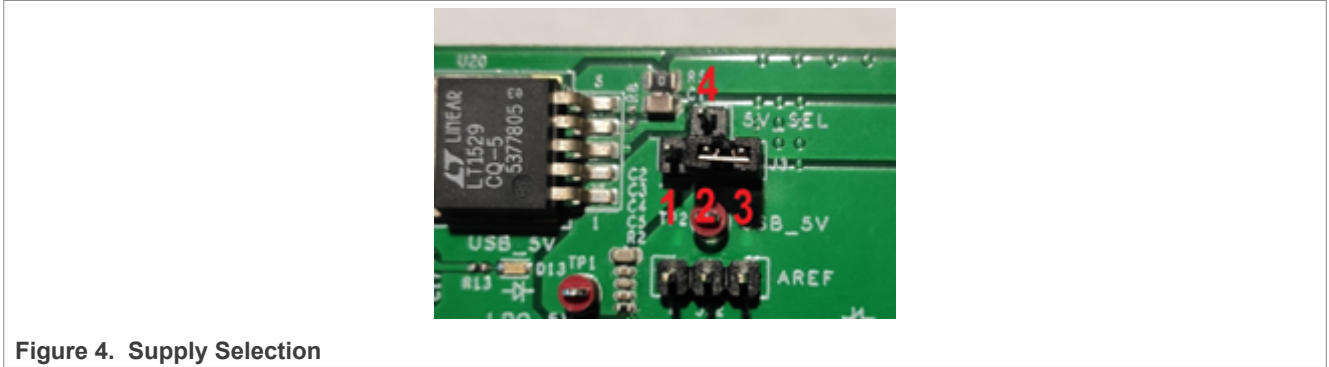2. Set jumper "J3 - 5V_SEL" to 3-2 (USB 5 V):

Figure 4. Supply Selection

See Section 3.1.1.1 "Power supply selection".

3. Plug in a USB-C cable to J5 (top USB-C connector) and to your PC (active USB hub recommended due to the high in-rush current).
See Section 1.1.1 "PN76 and external debugger".

4. Download, install, and open the MCUXpresso IDE.
   a. Create a new workspace for your PN7642 examples.
   See Section 5.1.1 "MCUXpresso IDE".

5. Download the PN7642 MCUXpresso SDK v02.12.01[1] from the PN7642 website, category Software.
   a. Install the SDK by dragging and dropping the .zip file in the "Installed SDKs" tab, found at the bottom of your MCUXpresso:
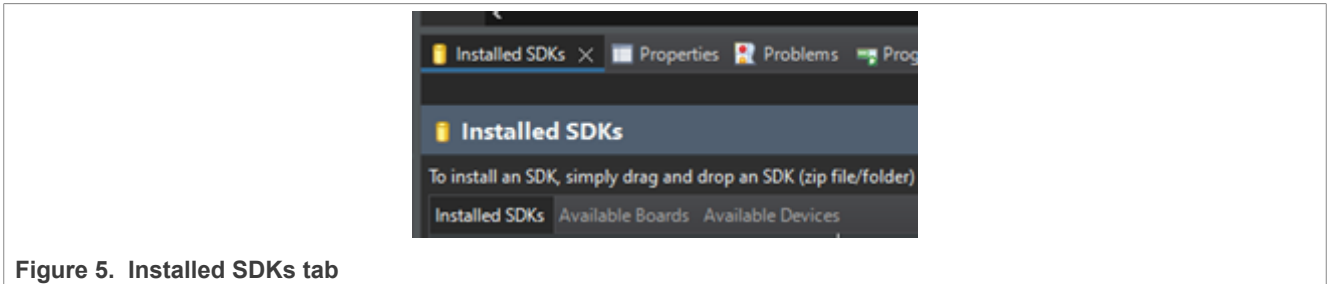
Figure 5. Installed SDKs tab

See Section 5.1.1.2 "Install SDK".

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

---

1 This version of the SDK has been specifically picked as it is compatible with FW v01.00, which is per default on our development boards. See Section 1.2 "Firmware overview".

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

9 / 67
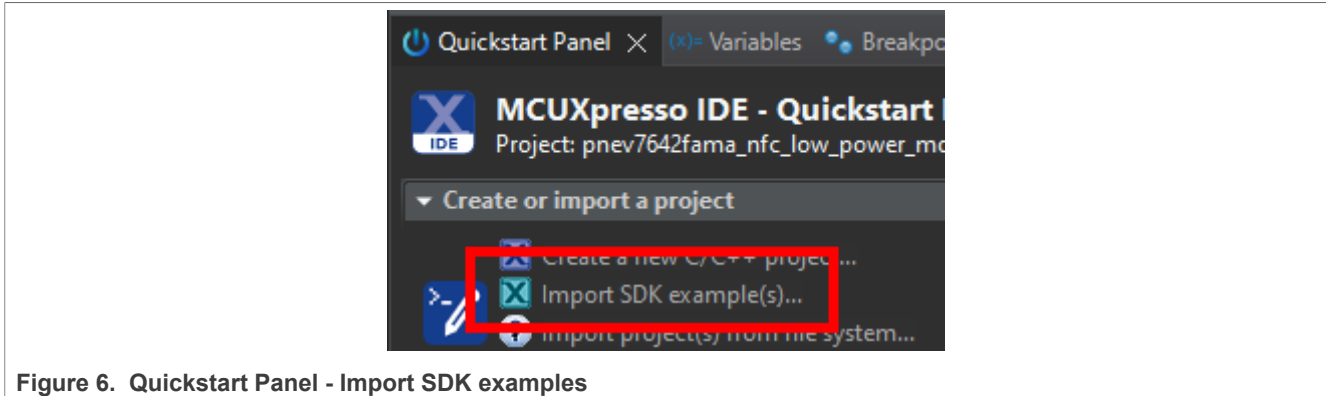
Figure 6. Quickstart Panel - Import SDK examples

    a. Import "nfc_examples" → "NfcrdlibEx1_DiscoveryLoop_Freertos" and check "Semihost" at the "Project Options":
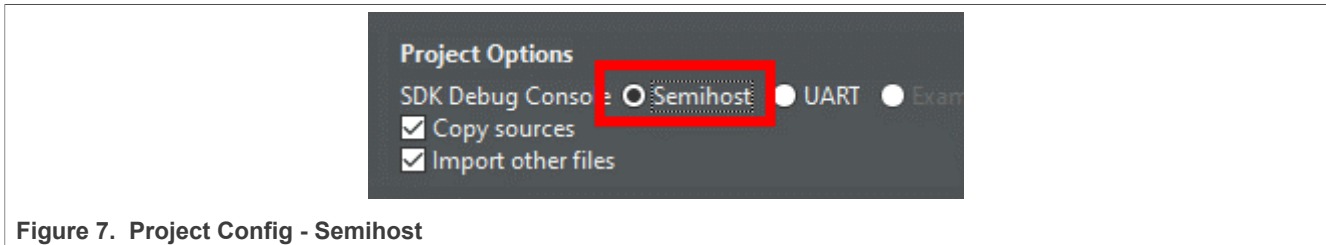
Figure 7. Project Config - Semihost

    b. Click "Finish" to start importing the example into your project workspace.
    See Section 5.1.2 "Import SDK example".

7. 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 3.1.2.1 "JTAG/SWD interface".

8. Click on the blue 'Bug' in MCUXpresso to compile and start debugging the example:
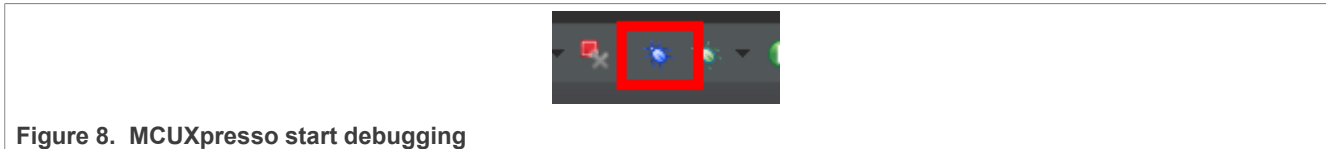
Figure 8. MCUXpresso start debugging

    a. A pop-up window should appear where you have to choose your debugger. Click on the debugger of your choice, which is connected to the PN7642, and press "Ok".
    See Section 5.1.3.2 "Debug example".

9. The program should be automatically halted at the void main(). Click "Resume" or press "F8" to let the example run:
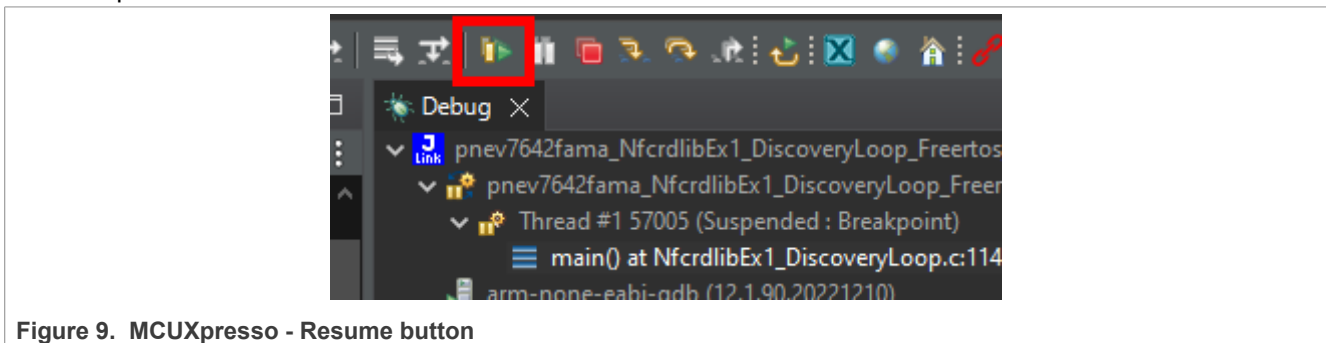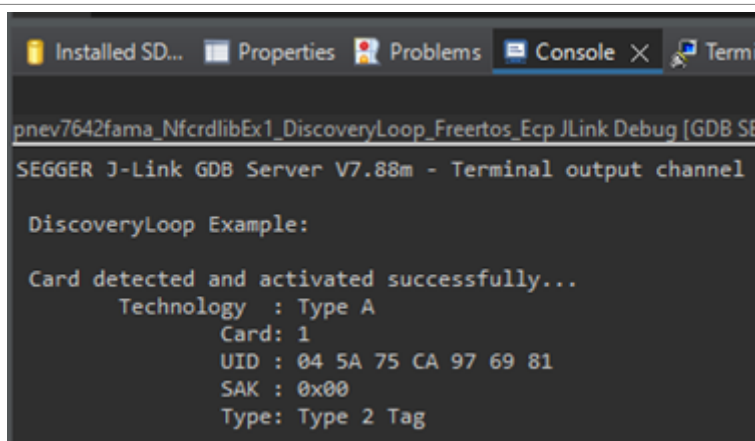
Figure 9. MCUXpresso - Resume button

10. Place a NFC-Tag on the antenna of the PNEV7642A development board. The example prints out information of the detected tag:

Figure 10. MCUXpresso - Console print out of ex1

3 Hardware

The PN76 PSP 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 6. PN76 family boards

Board name	Description
PNEV7642A Rev.B	Section 3.1
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.

3.1 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 [9].

The development board assembled of two major components:

- The based board which hosts all the periphery and connectors.
  - Schematic: [2]
- The module board which hosts the PN7642 chipset and its minimum required components (See Section 3.2 "PNEV7642MA").
  - Schematic: [3]
  - The module board can be used for rapid prototyping on a custom PCB.
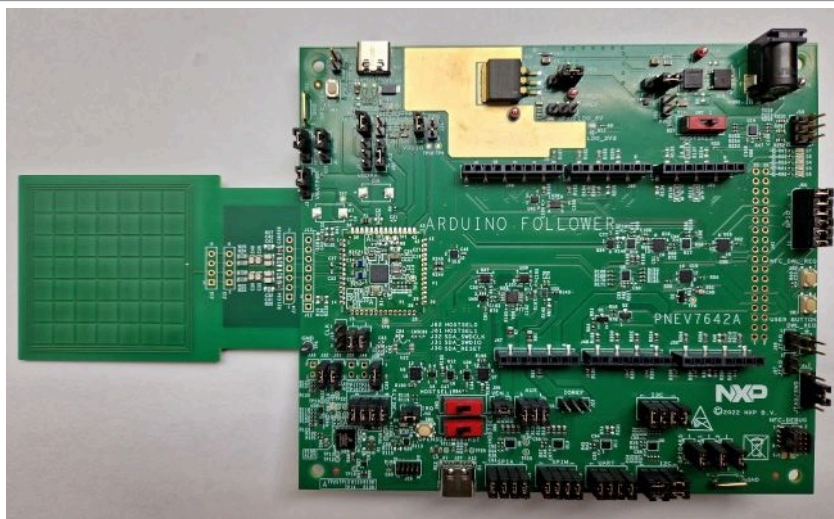
Figure 11. PNEV7642A Rev.B

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

12 / 67

3.1.1 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. 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 800 mA. The polarity does not matter, since there is a rectifier onboard. The inrush current of the DC-DC can be up to 1.6 A, when enabling the RF-Field.

Figure 12. Power section

AN13134

Application note

All information provided in this document is subject to legal disclaimers.

Rev. 2.7 — 6 December 2023

© 2023 NXP B.V. All rights reserved.

13 / 67
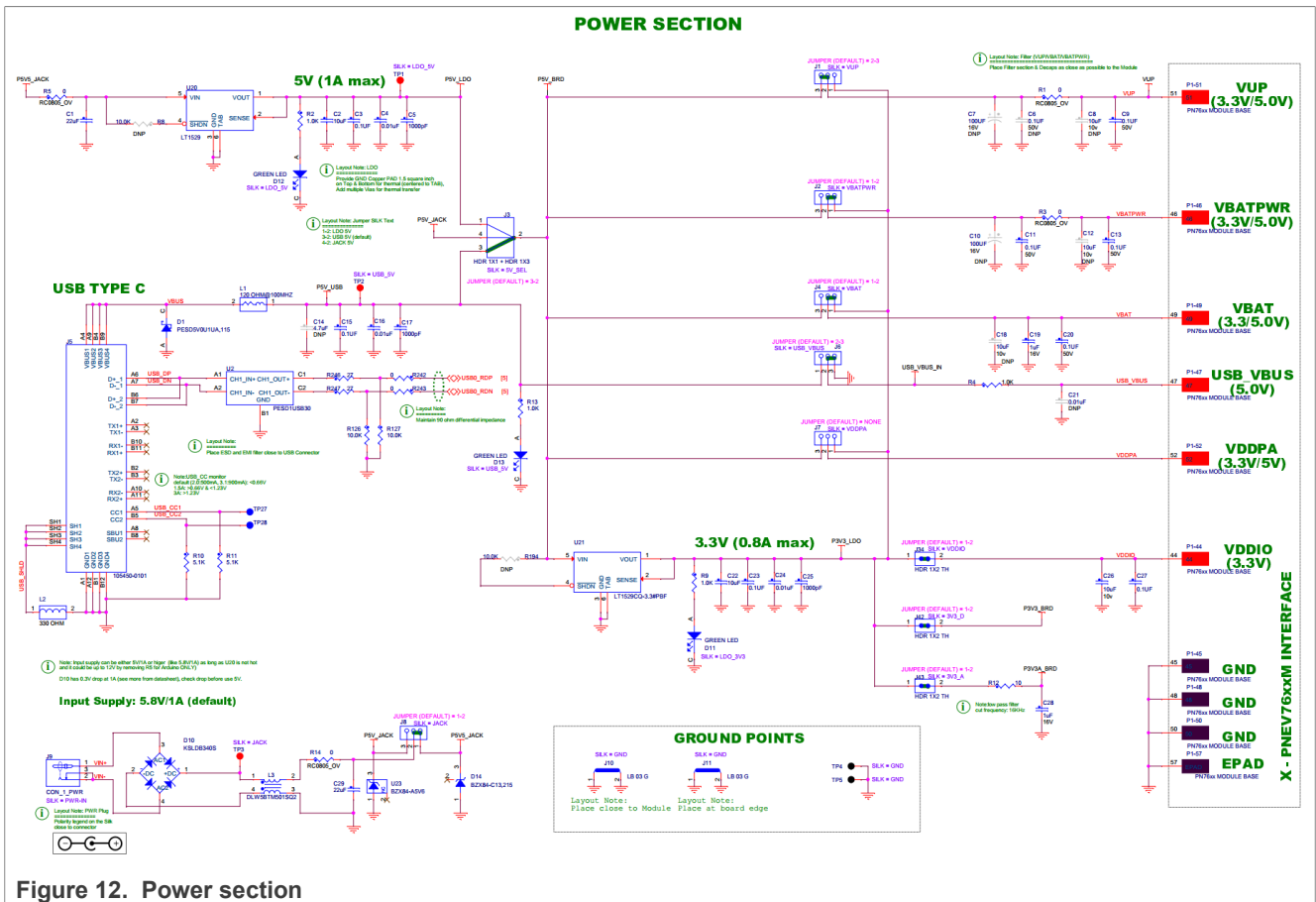
3.1.1.1 Power supply selection

The board can be either supplied by USB (not recommended) or a power jack (recommended), which requires an external power supply with 5 V to 7.5 V. It is not recommended to use the USB as power supply as the in-rush current is higher as the USB specification allows. Some USB ports (especially active USB-Hubs) might support this.

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

Table 7. Power selection

Pins	Description
1-2	LDO 5 V
3-2	USB 5 V (default)
4-2	JACK 5 V - 7.5 V
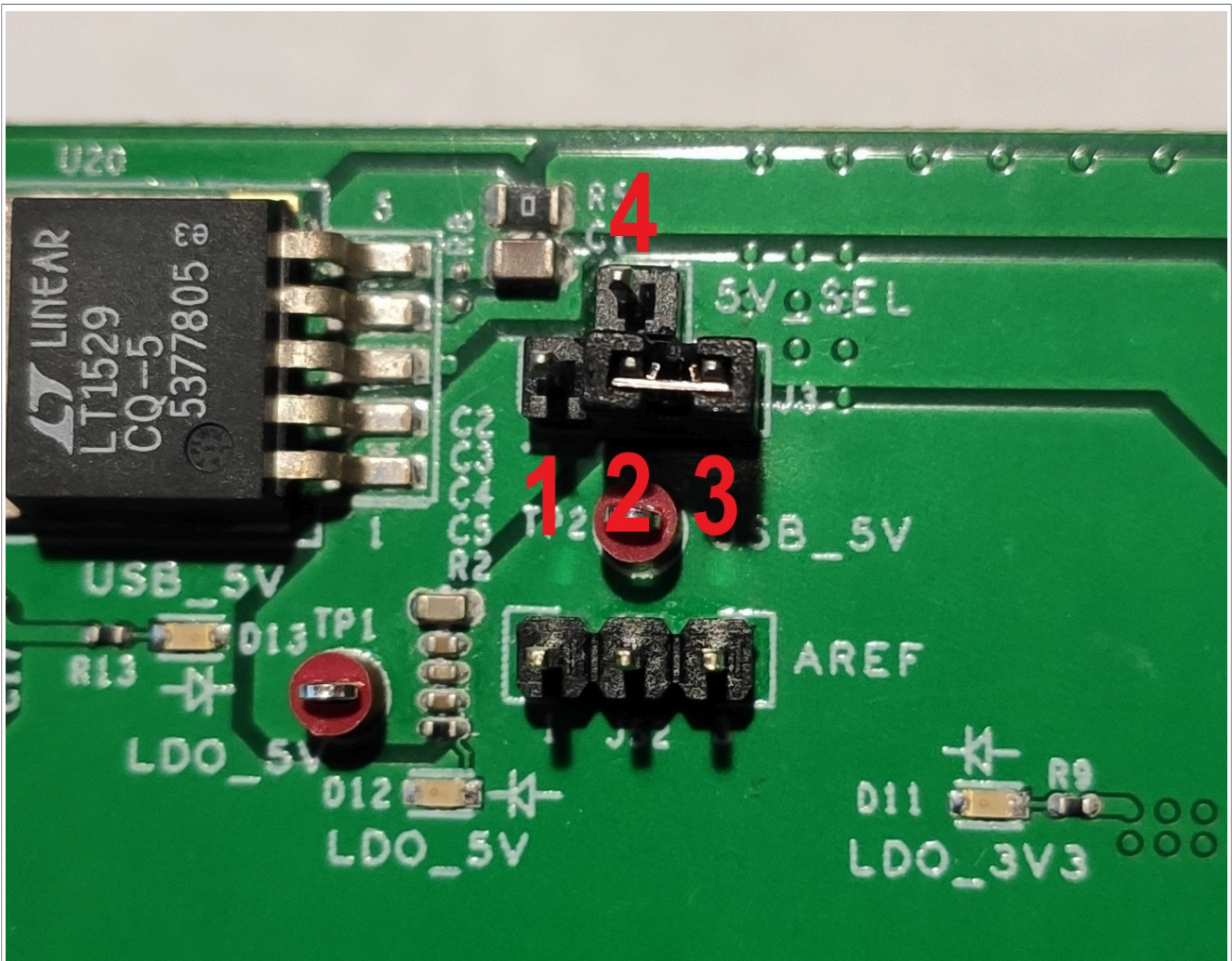
Figure 13. Power selection jumper

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

14 / 67

3.1.1.2 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 8. 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!
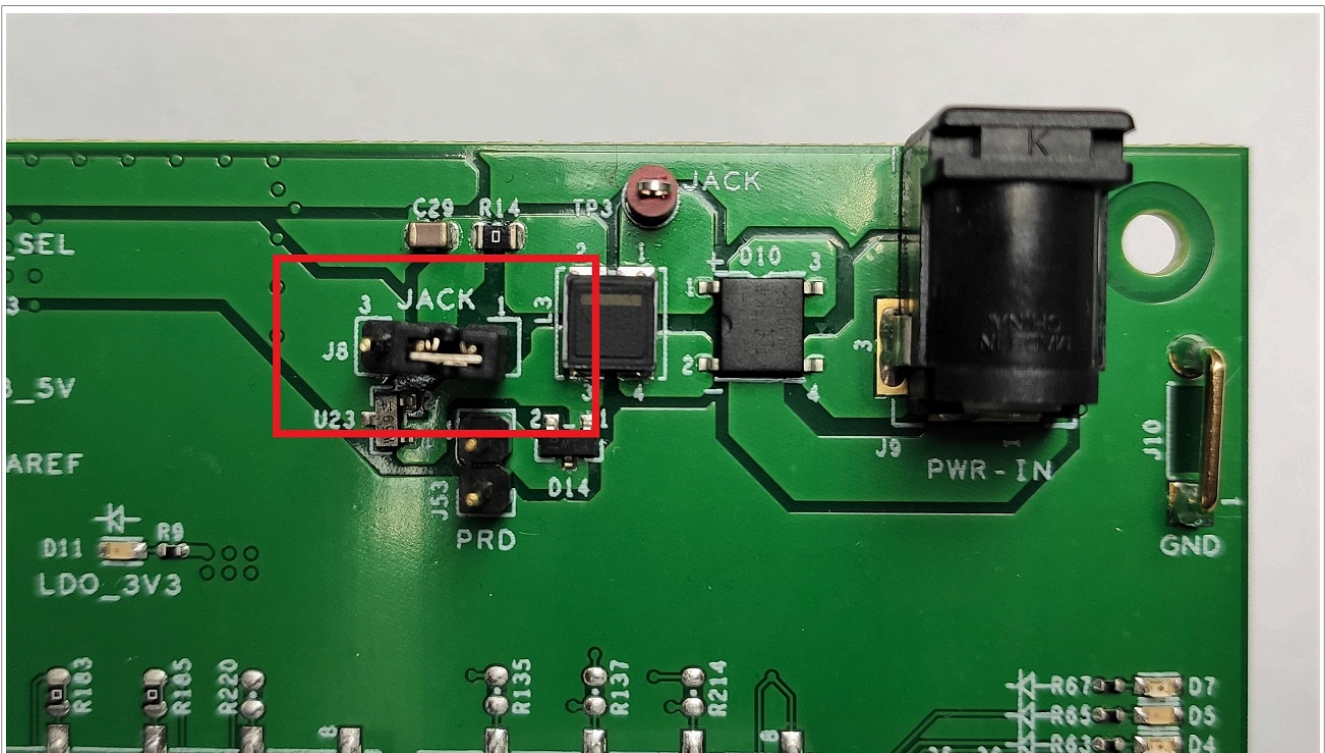
Figure 14. PNEV7642A development board power jack jumper

Figure 15. PNEV7642A development power jack schematic
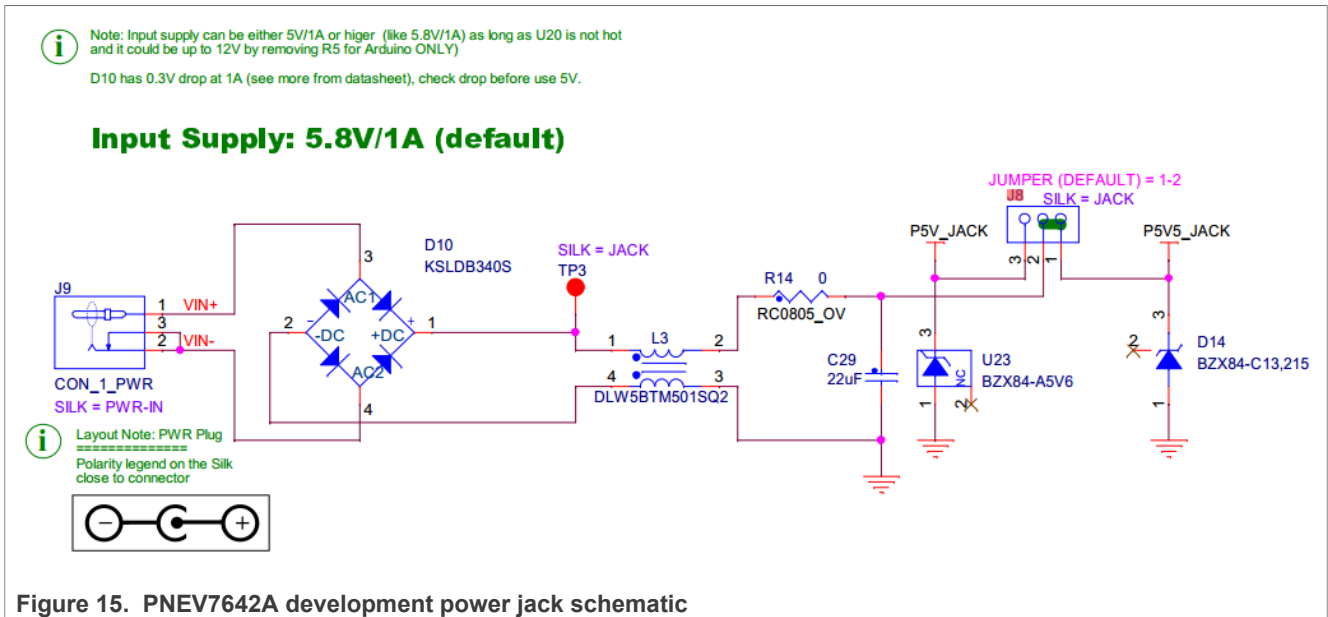
3.1.2 Debug interfaces

The board offers two ways of debugging. They are mutual exclusive. Hardware changes have to be made to switch from one to the other.

Per default, the JTAG/SWD interface (Section 3.1.2.1) for an external debugger is active.

3.1.2.1 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).

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 9. J17 Interface jumper setting

Pins connected	Description
1-2	JTAG Interface
2-3	SWD Interface (recommended)
3-4	JTAG Interface
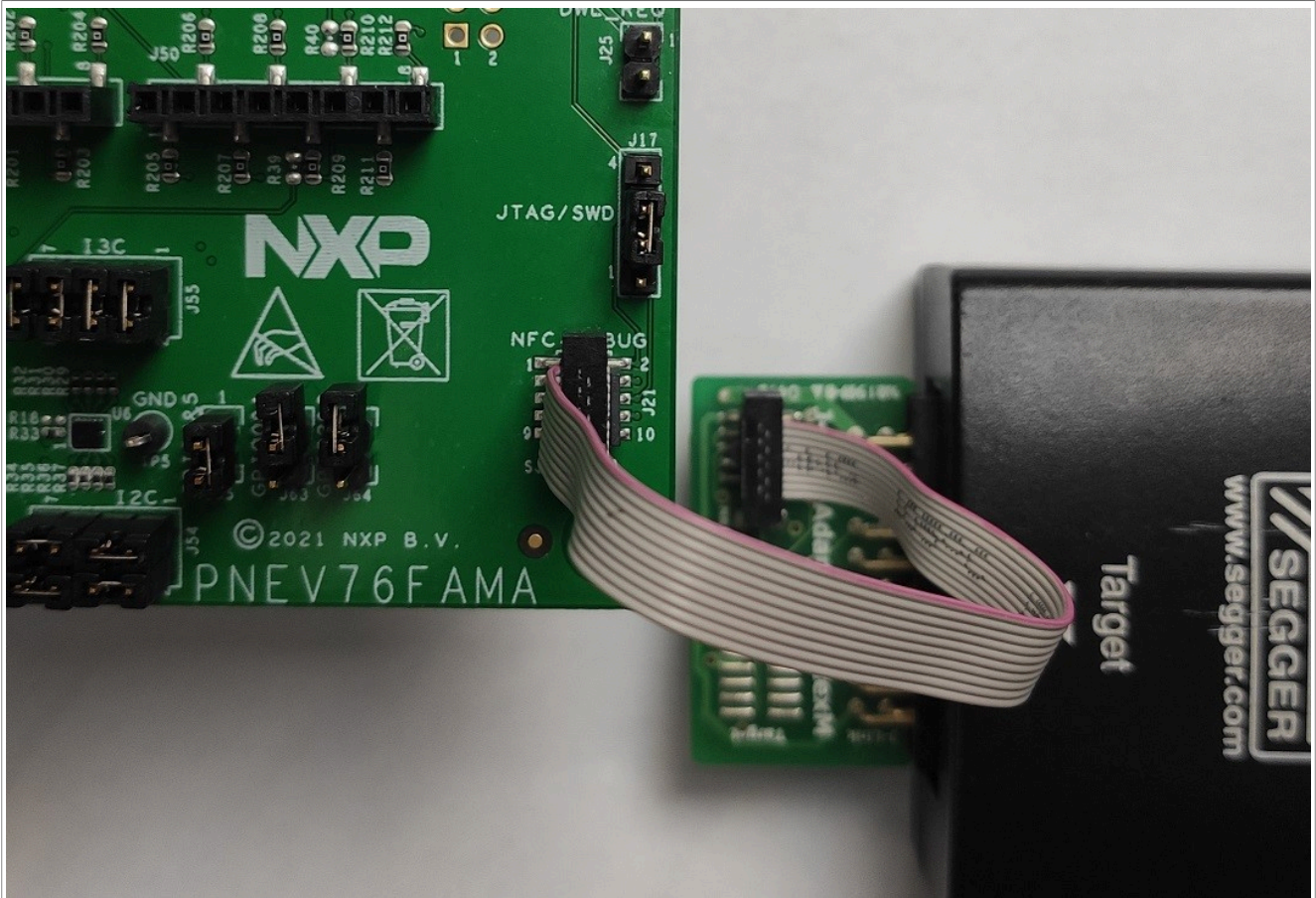
Figure 16. Debug interface PNEV76FAMA and PNEV6742A
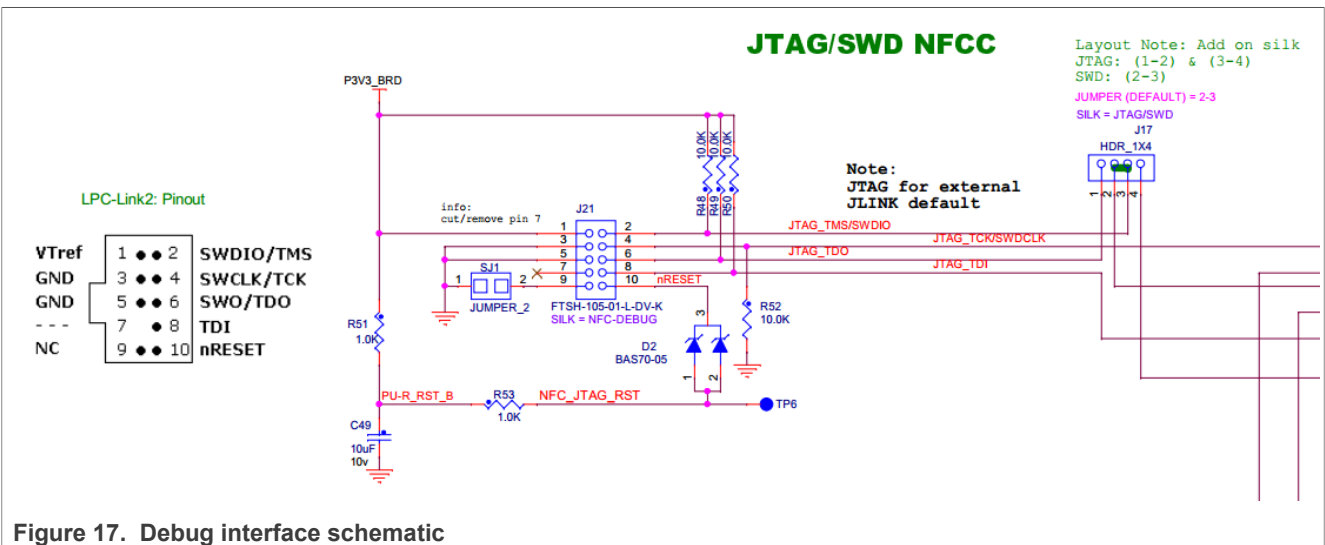
Figure 17. Debug interface schematic

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

17 / 67

3.1.2.2 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 the application note [1] how to enable OpenSDA.
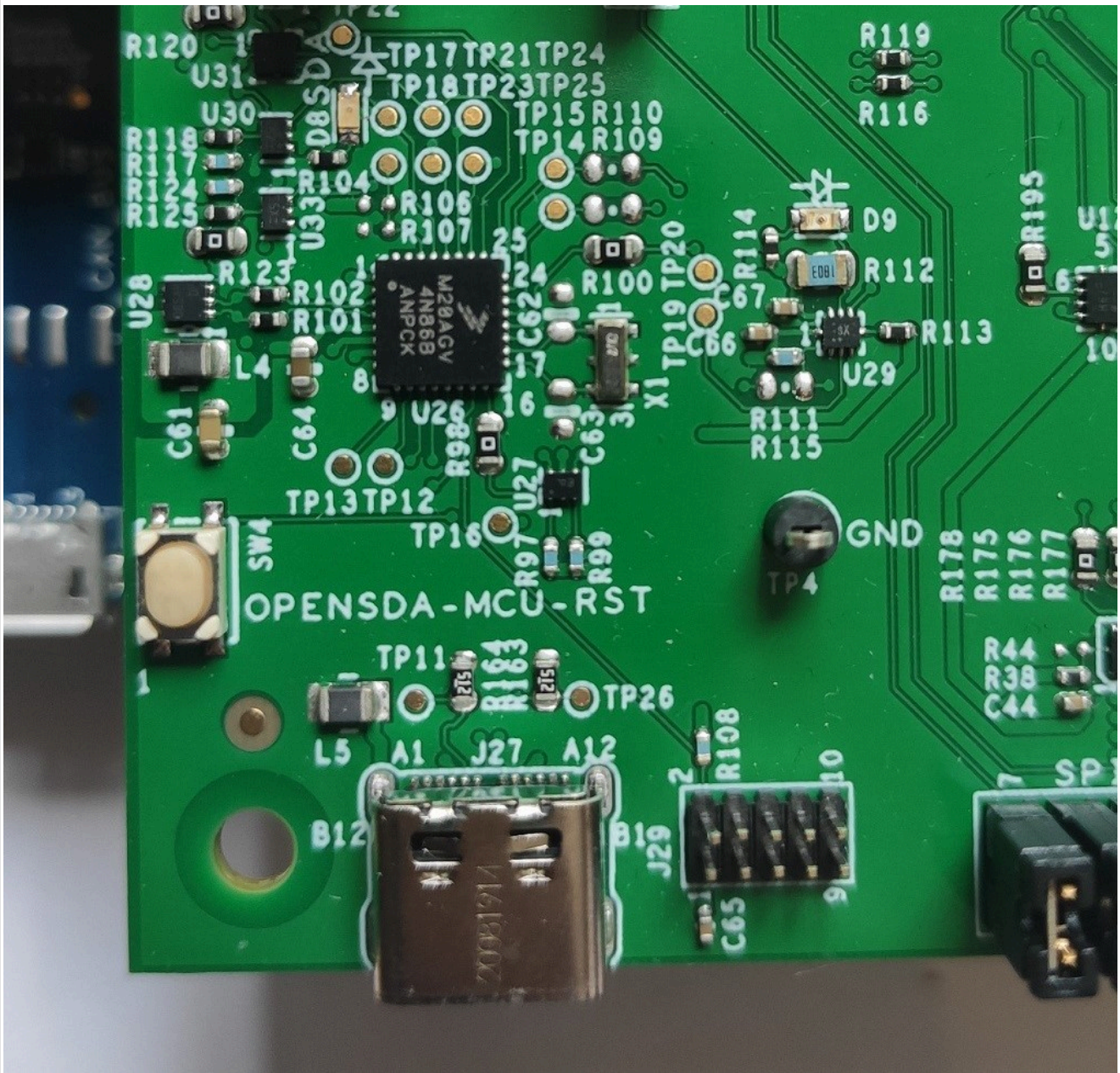
Figure 18. OpenSDA connectors

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

18 / 67

3.1.3 Antenna

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.
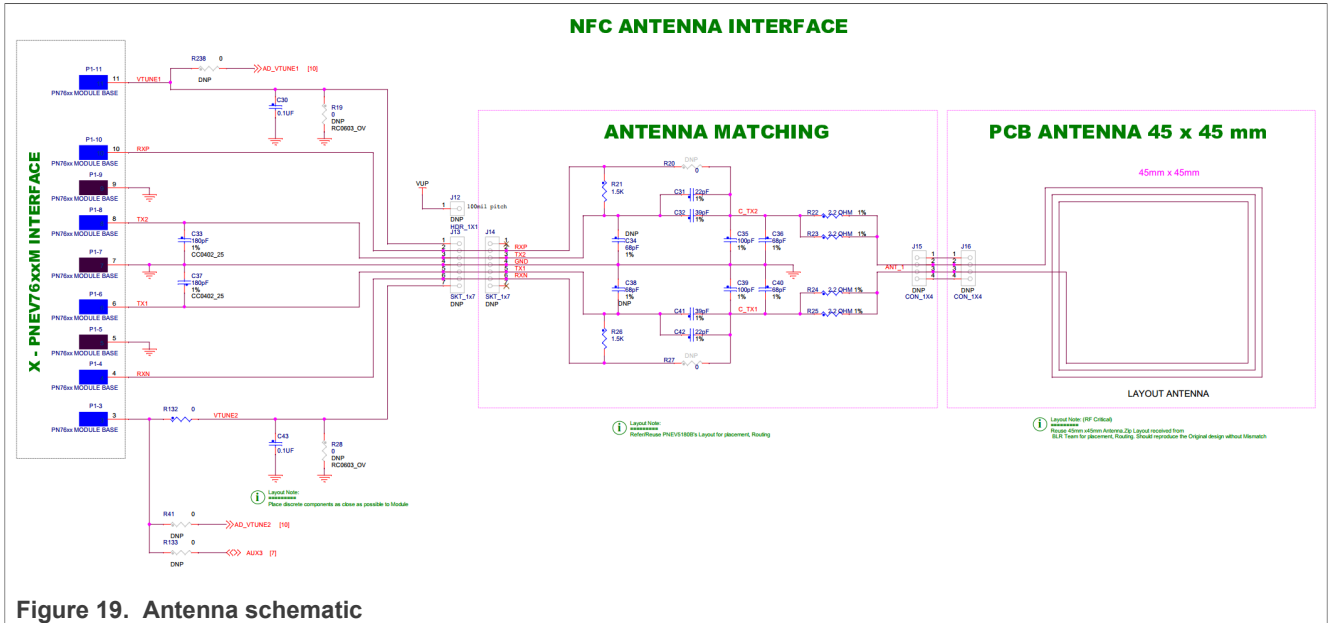
Figure 19. Antenna schematic

3.1.4 Jumper settings

The development board is hosting many jumpers and switches. This is to enable the configuration of every signal and easier debugging. With the downside of looking, and being, more complex.

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 [2].

3.1.4.1 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).

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

19 / 67
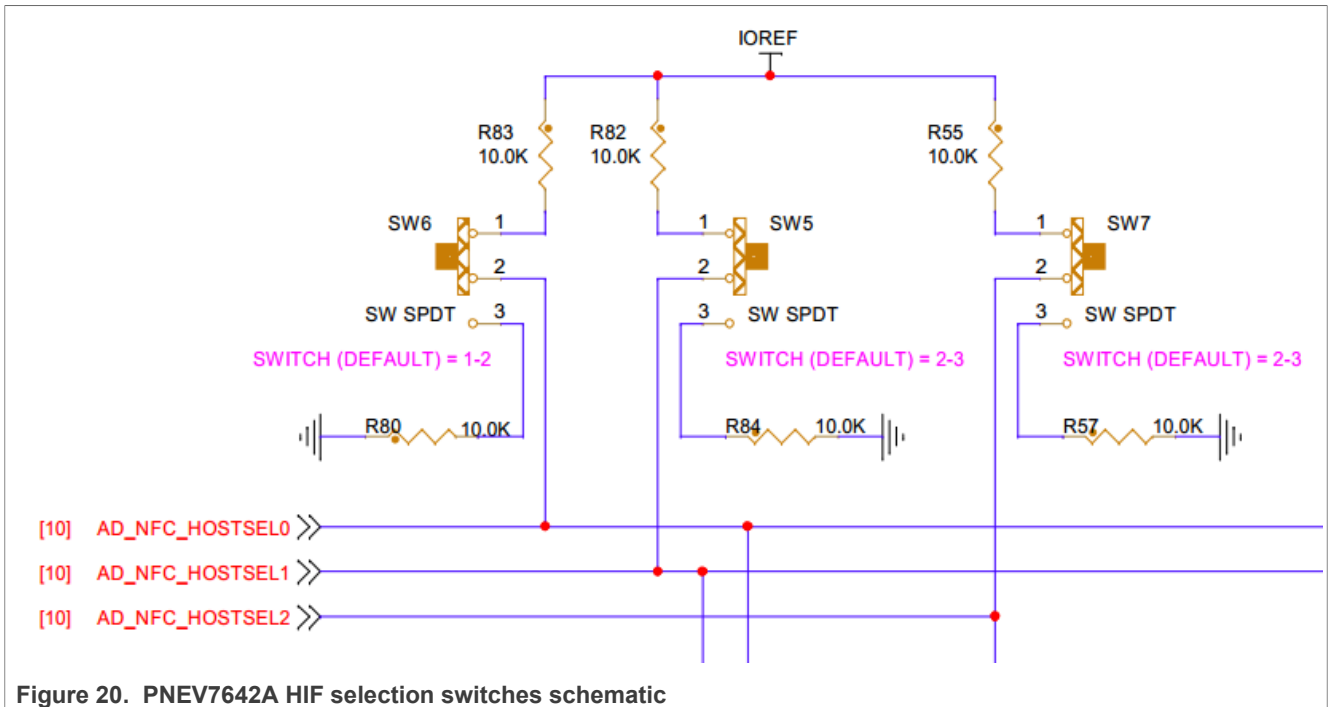
Figure 20. PNEV7642A HIF selection switches schematic

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

Table 10. 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.

3.1.4.2 IOREF

Per default, IOREF is not connected and this represents the "standalone mode" of the board (no Arduino controller installed - like our 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.

3.1.4.3 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.
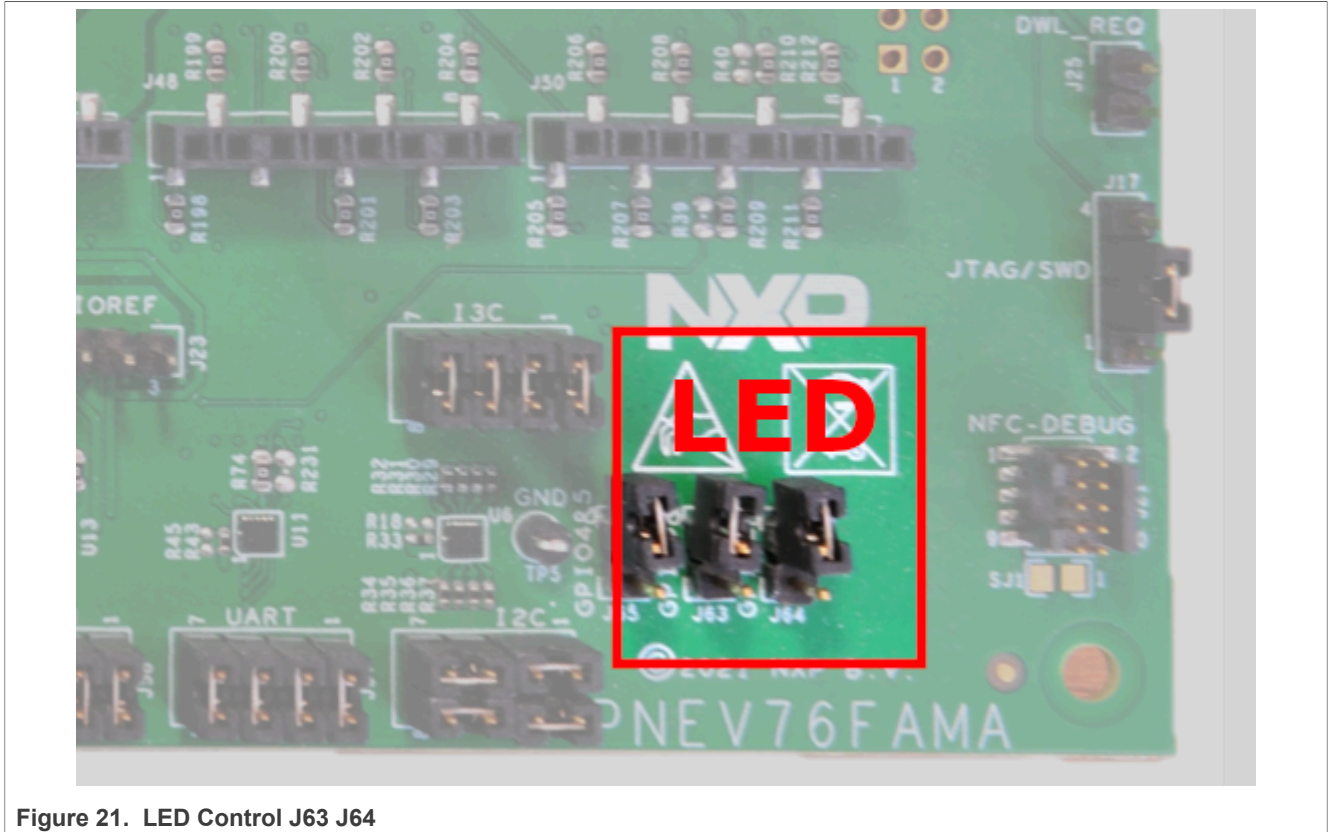
Figure 21. LED Control J63 J64

3.2 PNEV7642MA

On the development board is 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 PCB's, 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.

Rev. 2.7 — 6 December 2023
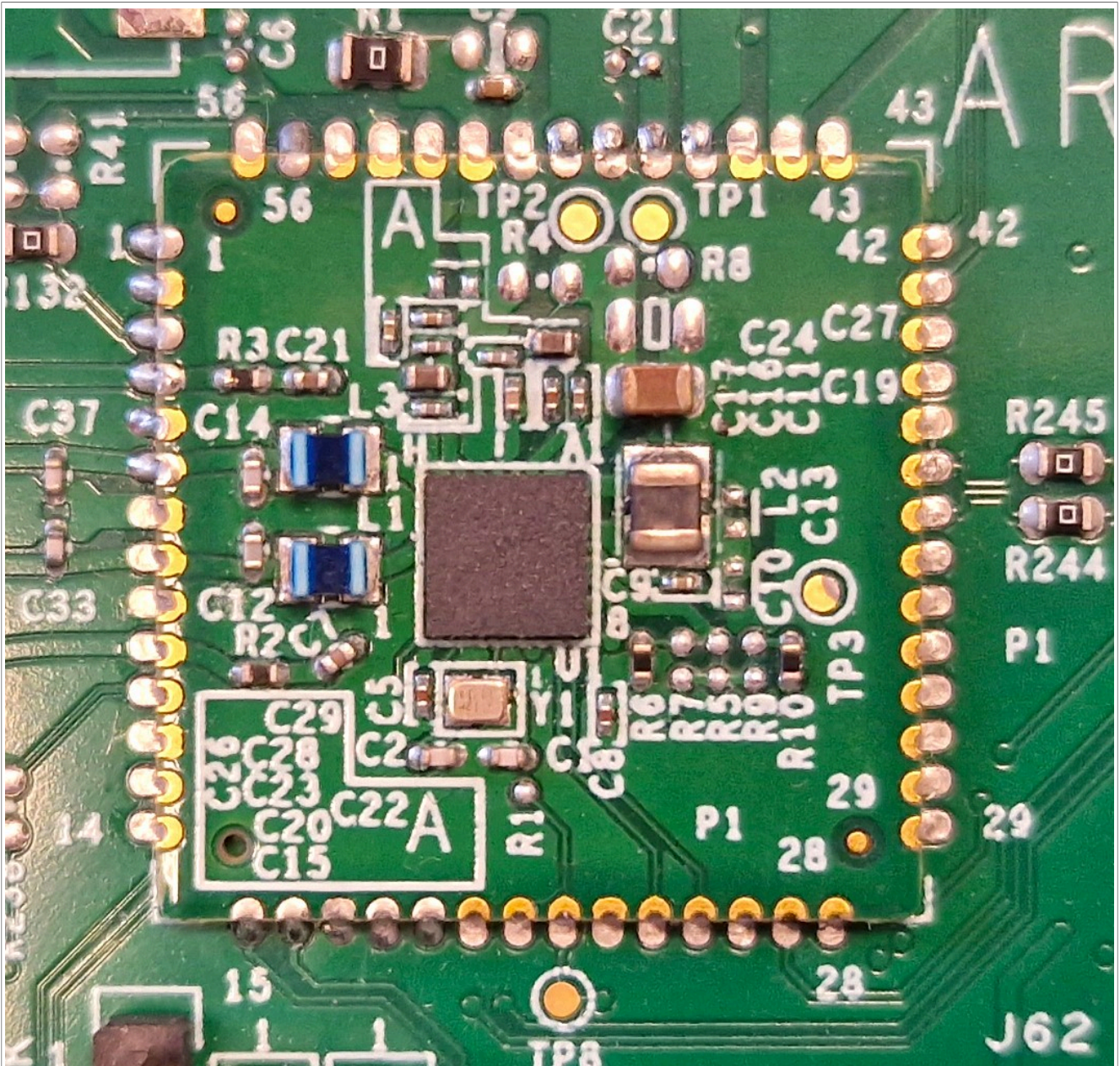
Figure 22. PNEV7642MA

The module board shown in [Figure 22](#) is according to schematic file "spf-54814_a1".

DCDC disabled: On this module board the DCDC is disabled. R4, C29 and C13 are not populated (see [)](#). VUP is connected to P5V_BRD through "J1" of the development board.
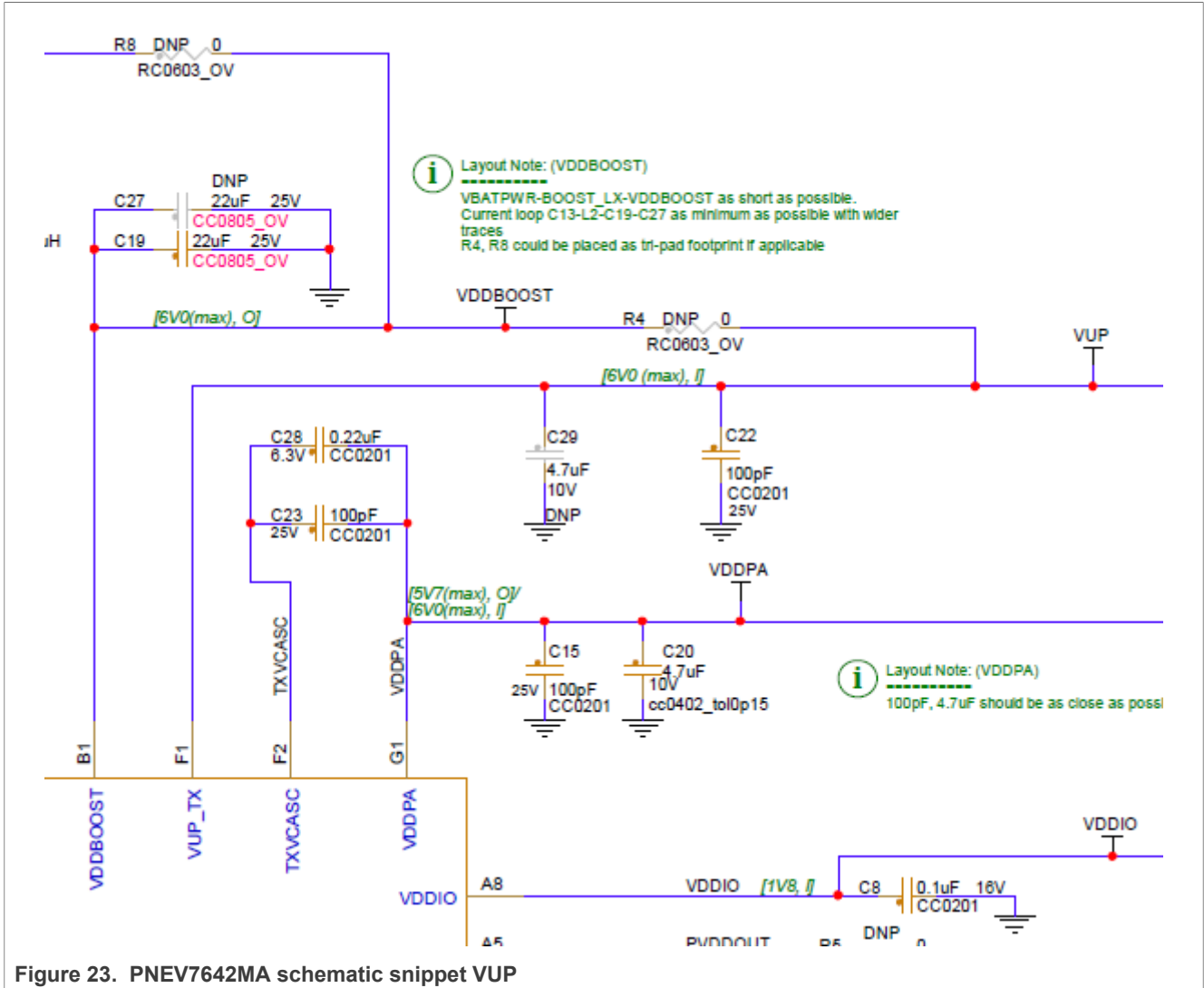
Figure 23. PNEV7642MA schematic snippet VUP

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

23 / 67

3.3 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: [4]
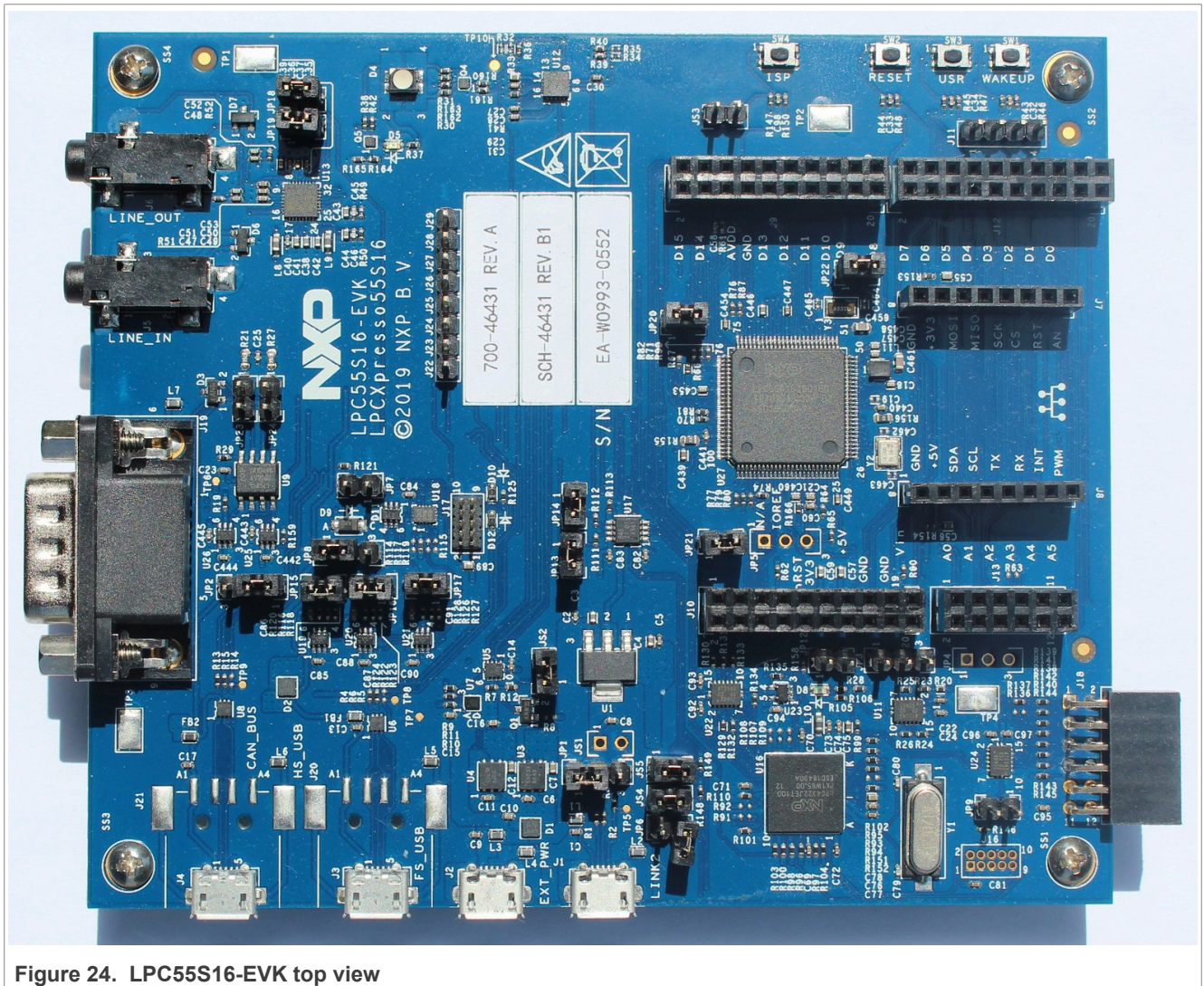
Figure 24. LPC55S16-EVK top view

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

24 / 67

3.3.1 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).
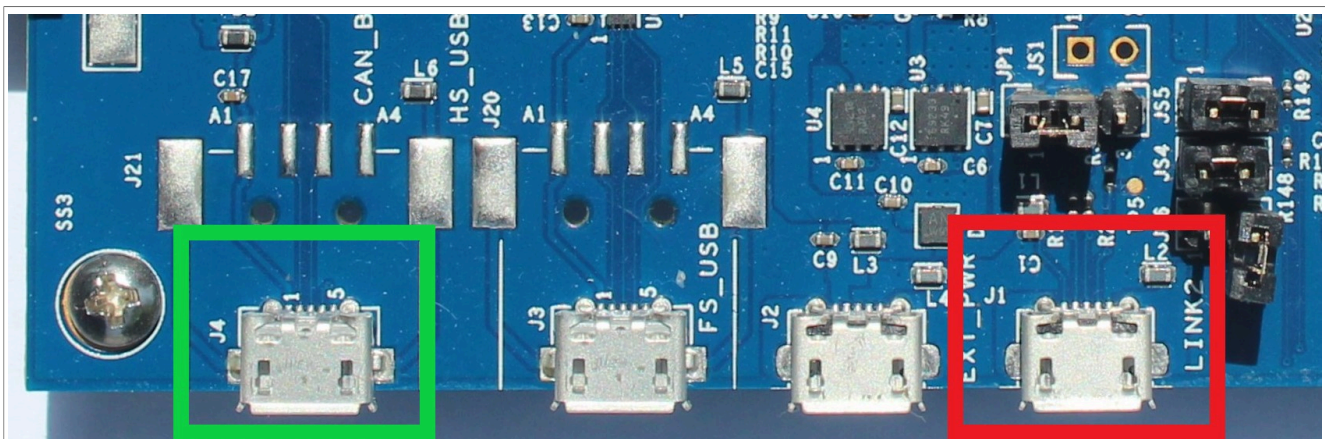
Figure 25. USB connectors

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.

3.4 Connecting the LPC with the PN76

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 26 is to be connected with the PN76 board while the red area stays unconnected.

AN13134

Application note **Rev. 2.7 — 6 December 2023**

26 / 67
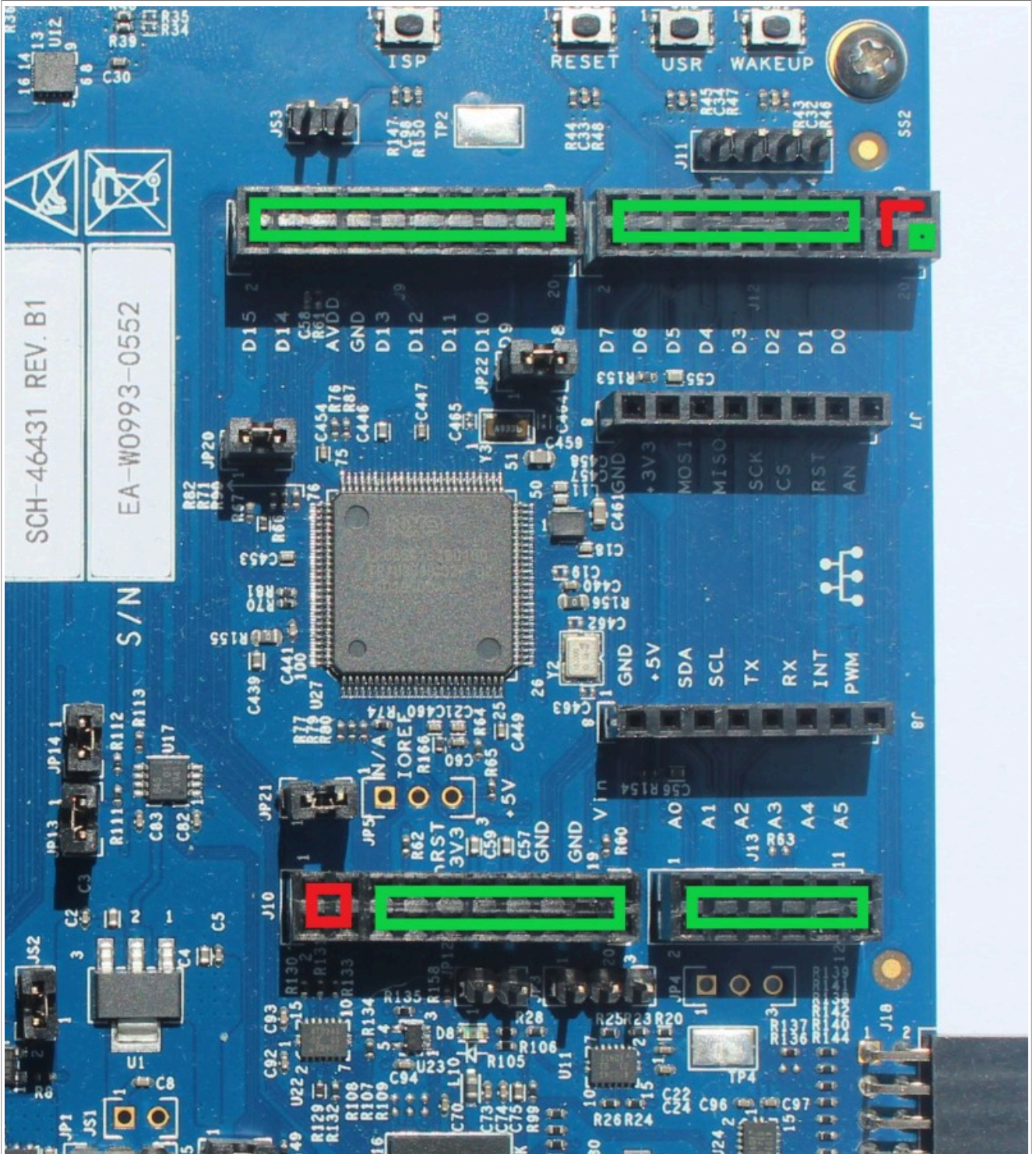
Figure 26. LPC55 arduino alignment

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

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

27 / 67
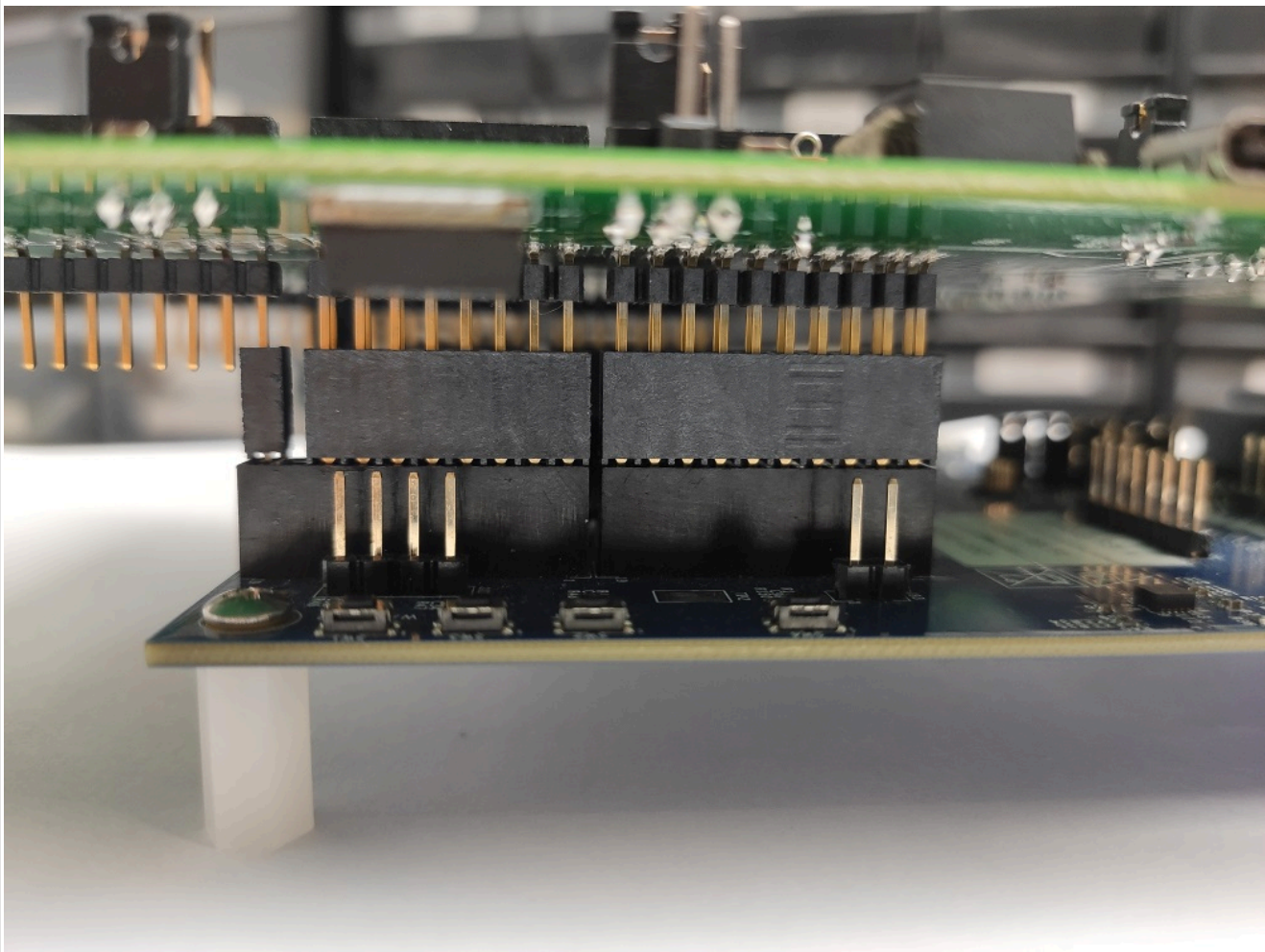
Figure 27. Alignment J9
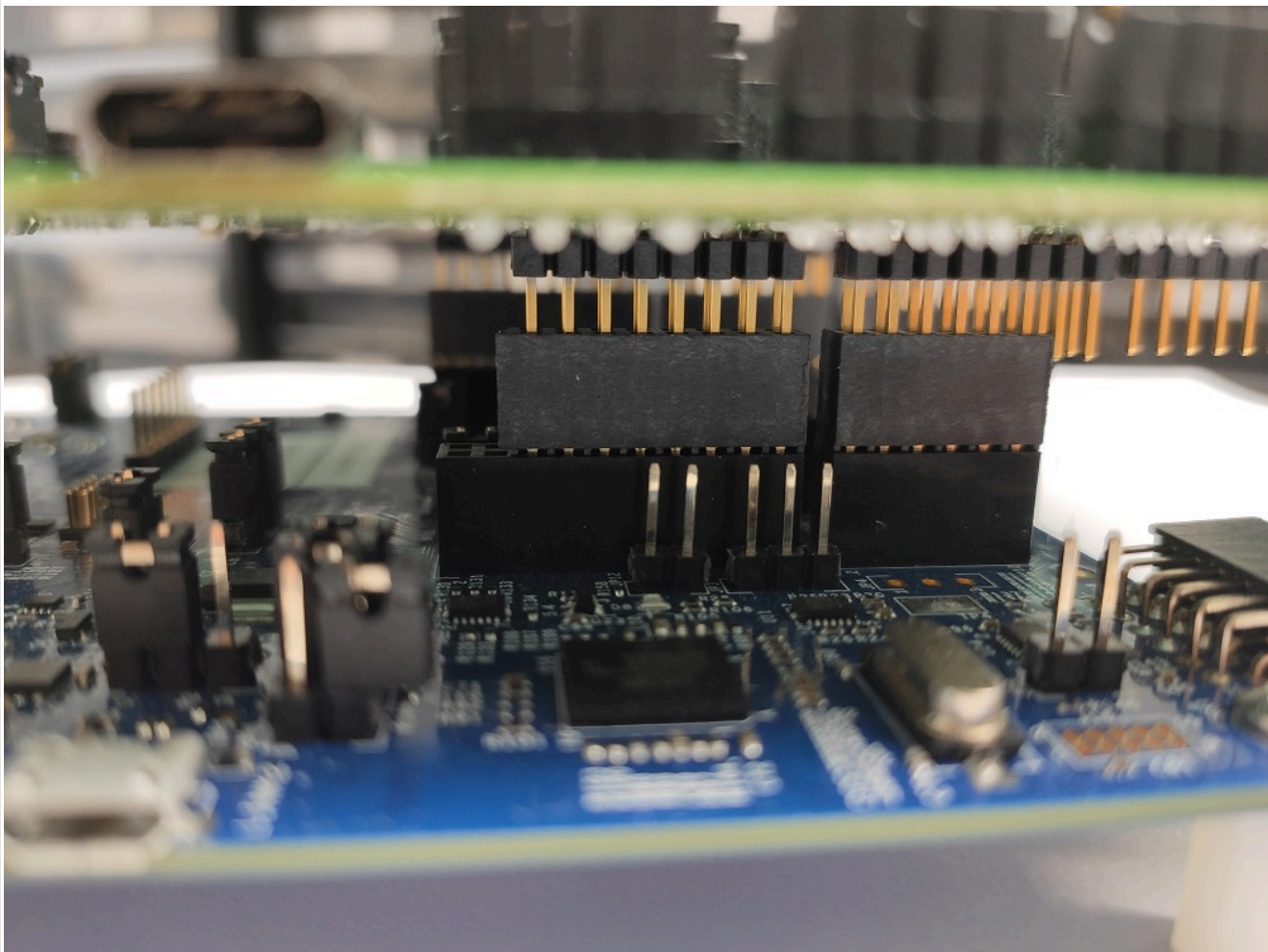
Figure 28. Alignment J10
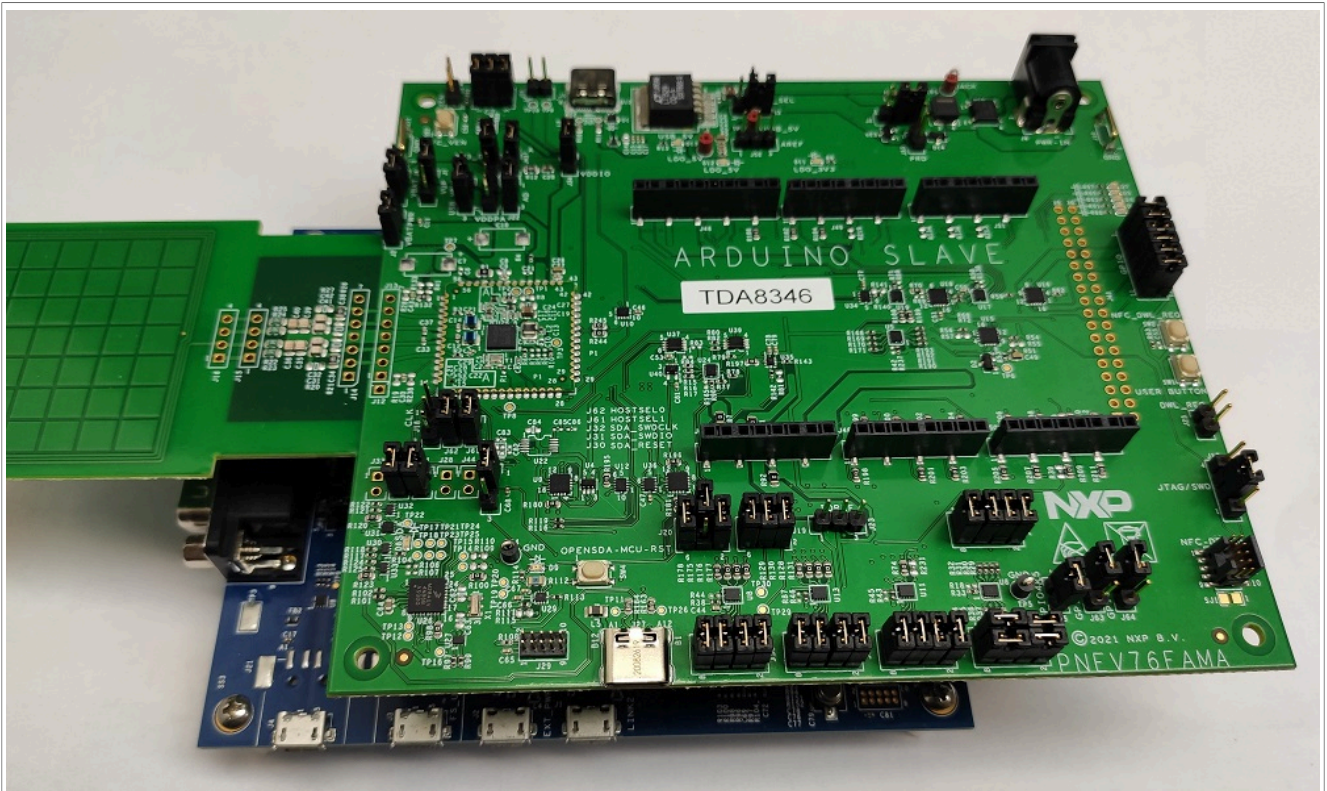
Figure 29. Stacked PNEV76FAMA on LPC55S16-EVK

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

30 / 67

3.5 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.
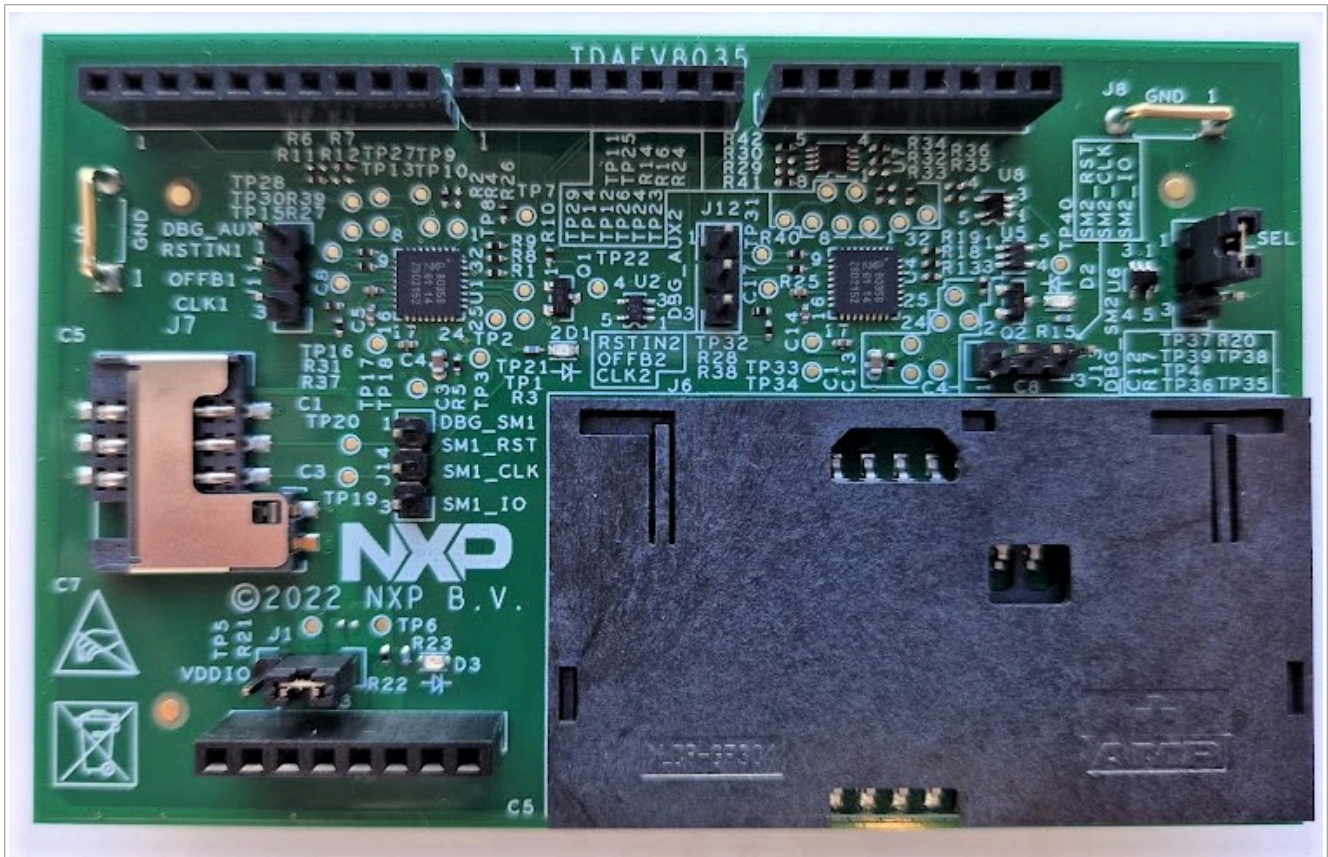
Figure 30. TDAEV8035

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

3.5.1 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 11. TDAEV8035 to PNEV7642A alignment

TDAEV8035		PNEV7642A
J3	aligned with	J46
J4	aligned with	J49
J5	aligned with	J51
J2	aligned with	J47

Table 11. TDAEV8035 to PNEV7642A alignment...*continued*

TDAEV8035		PNEV7642A
	not connected	J48
	not connected	J50
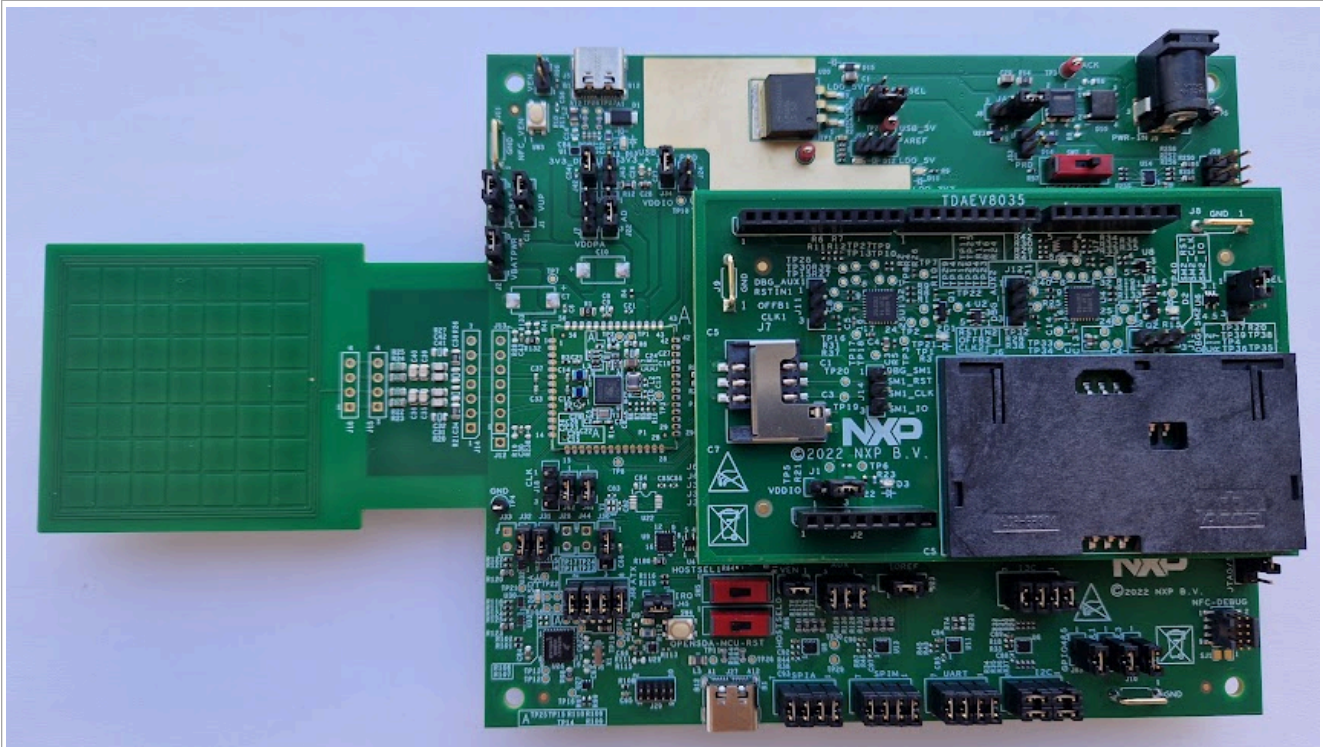
Figure 31. TDAEV8035 on PNEV7642A

3.5.2 Jumper settings

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

Table 12. 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).

3.6 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 minimum 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.

3.6.1 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 [10].

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.

3.6.2 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.

Rev. 2.7 — 6 December 2023

4 Software

All PN7642 relevant software can be found on [Secure Files 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 1.2](#) for the version compatibility.

Table 13. 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 14. 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 15. 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

4.1 MCUXpresso SDK

MCUXpresso homepage: [5]

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 5.1.1.2 "Install SDK"](#)

The SDK offers many examples designed for the PNEV7642A development board:
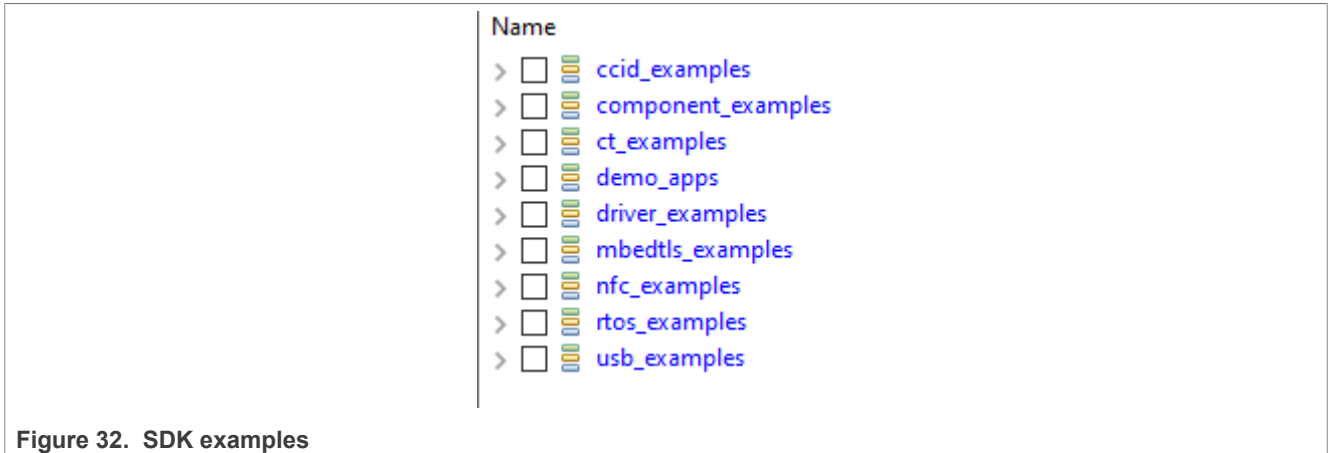
Figure 32. SDK examples

How to import and build examples is described in: Section 5.1.2 "Import SDK example"

4.2 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 3.4.

The host software can be downloaded on nxp.com at PN7642 software: [6]

4.2.1 Import projects

Download and extract the software package. Create a separate folder for the MCUXpresso workspace and open MCUXpresso using this folder as 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".
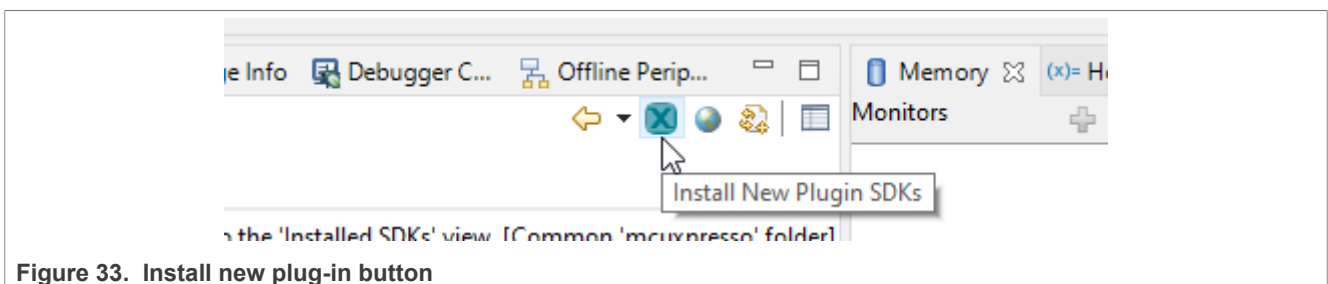
Figure 33. Install new plug-in button

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.
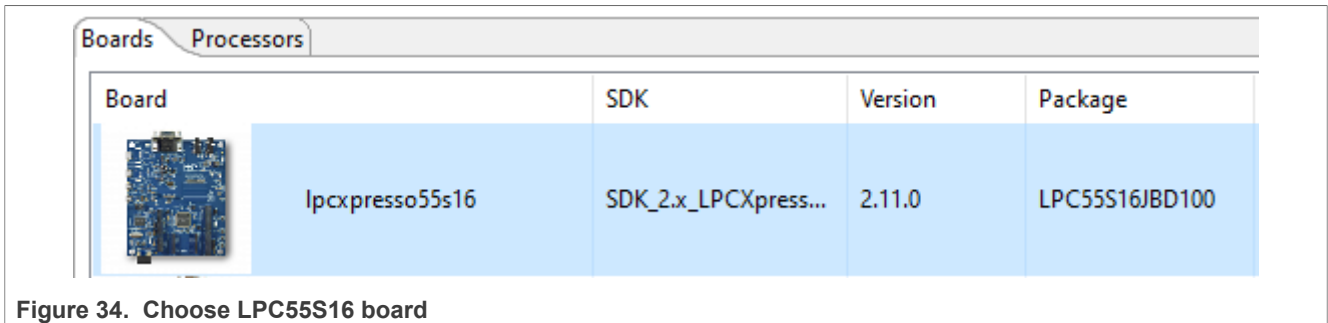
Figure 34. Choose LPC55S16 board

Having the projects imported and the SDK installed, the workspace is ready to run the host utils examples.
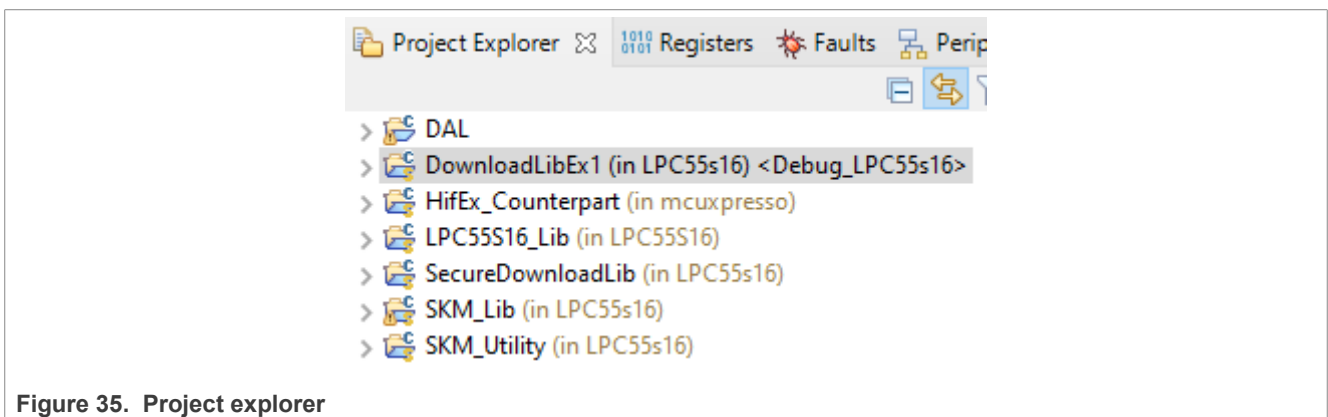
Figure 35. Project explorer

4.2.2 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.

4.2.2.1 Edit firmware location

At the top, around like 60, of the main file of DownloadEx1 the path to the firmware files is defined. This has to be changed to meet your 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.

You can either rename the firmware file to match the define in the example or you can edit the define in the example to match the firmware files name. Later has the benefit of not losing important information in the name, such as the version. Which often is handy if you have to run the update process later and might have forgotten what firmware file it was.

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

```
57
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 36. Firmware file path editing

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

Take special care to use a double backslash, as a single backslash is marking a special character.

4.2.2.2 Run firmware update

By running the example, there are several options printed in the console to be 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 37. Firmware update options

Choosing the first 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 38. Read firmware version

The connected PN7642 is running firmware version 01.00 which is deprecated and must be updated. 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 39. 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.

5 Development environment

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

Table 16. 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

5.1 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 5.1.1.2 "Install SDK".

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

5.1.1 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.

5.1.1.1 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.

5.1.1.2 Install SDK

Make sure that you have either downloaded the PN7642 SDK from the nxp website or received it from an NXP official.

Close the "Welcome page" of MCUXpresso. Go to the "Installed SDKs" tab on the bottom. In the following picture, multiple SDKs are already installed. This variously depending on previously installed SDKs.

If any SDKs are installed or not will not change further procedure, do not get confused by it.
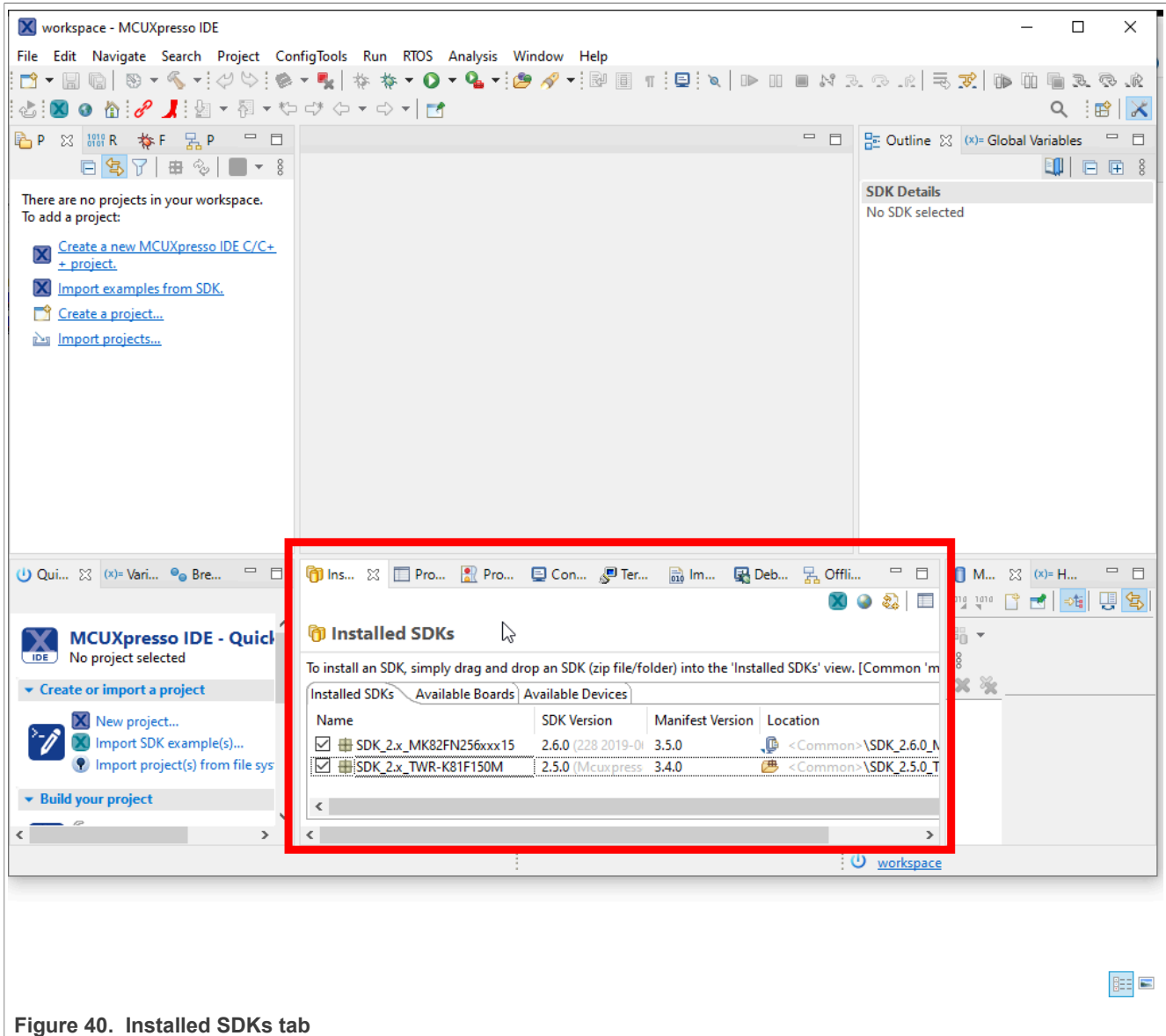
Figure 40. Installed SDKs tab

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.

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 41. Installed PN76 SDK
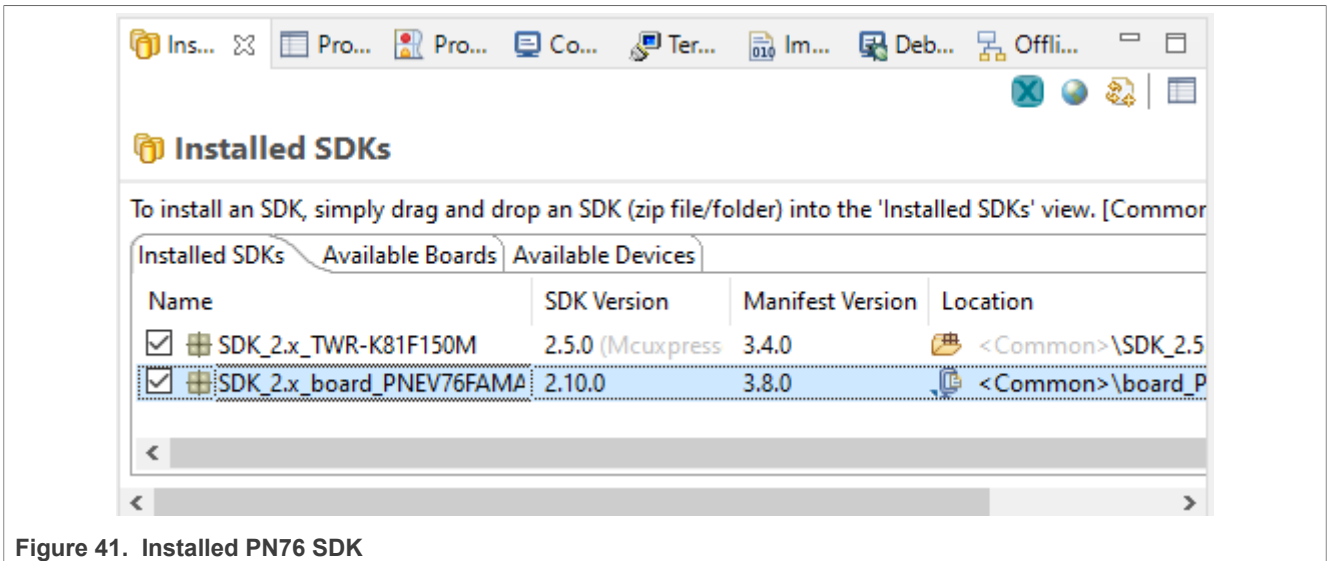
5.1.2 Import SDK example

After the successful SDK installation, you can import examples from it. This is done by clicking "Import SDK example(s)..." in the Quickstart Panel or by clicking "Import examples from SDK." in the Project Explorer.
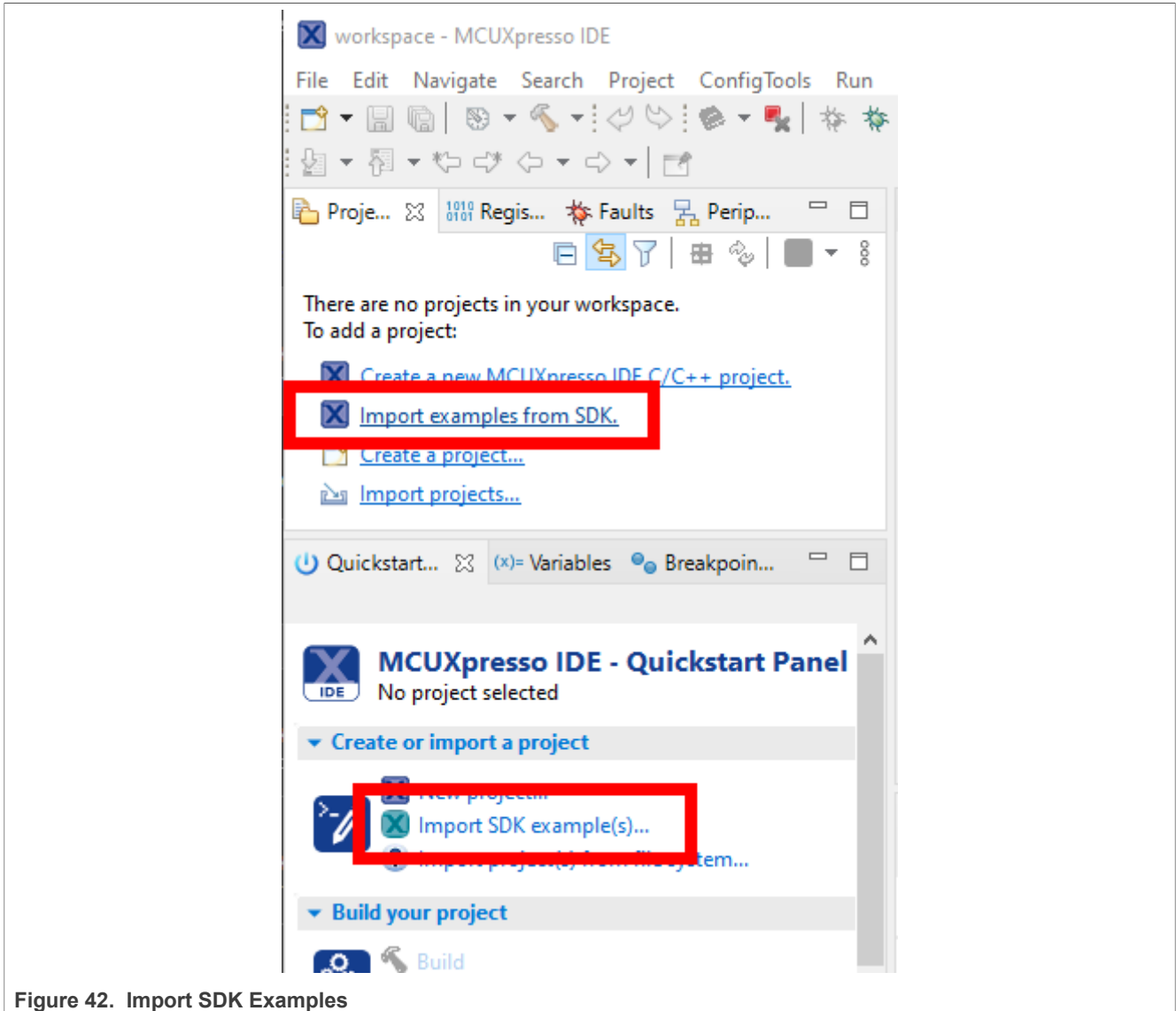
Figure 42. Import SDK Examples

In the following window, your 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.
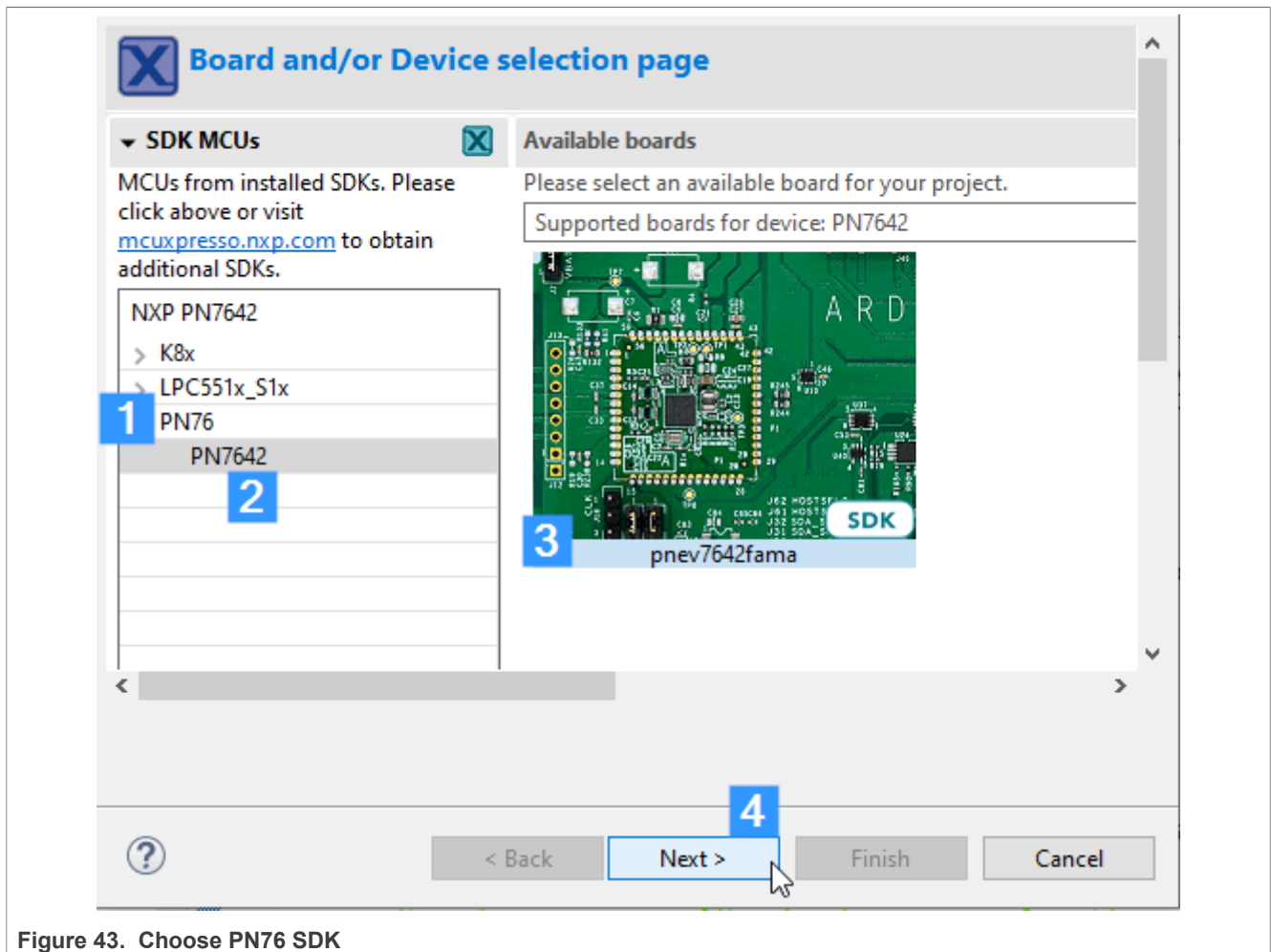
Figure 43. Choose PN76 SDK

At the list of examples chose whatever fits your needs. For the purpose of showing how it is done, we choose the 'led_blinky' example.
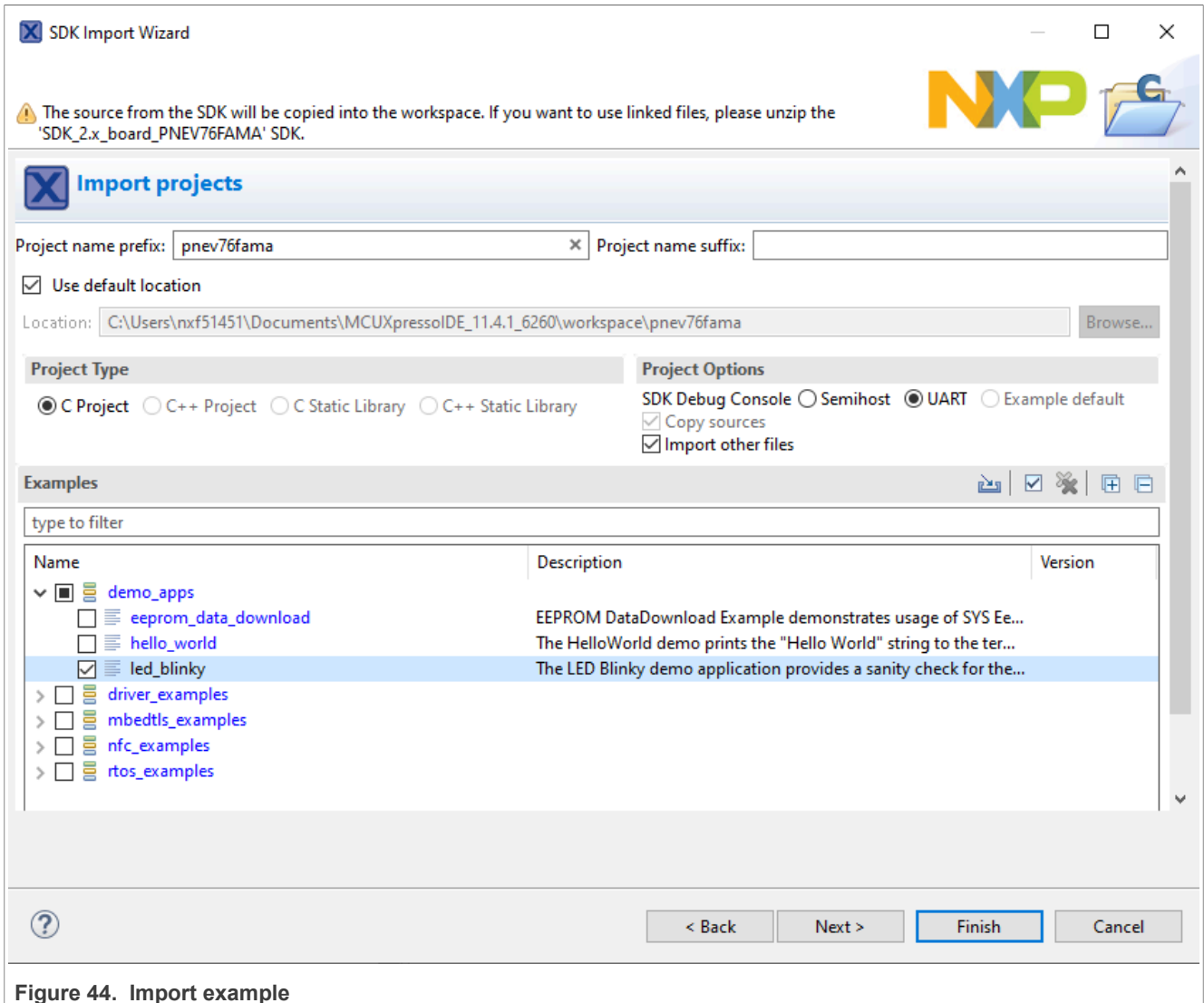
Figure 44. Import example

NOTE: Per 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.

5.1.3 Build example

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 is optimized in size the option, "Debug" will disable optimization and enable (via Preprocessor defines) also 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 pre-configured option, choose either or at the build button at top:

Figure 45. Build example
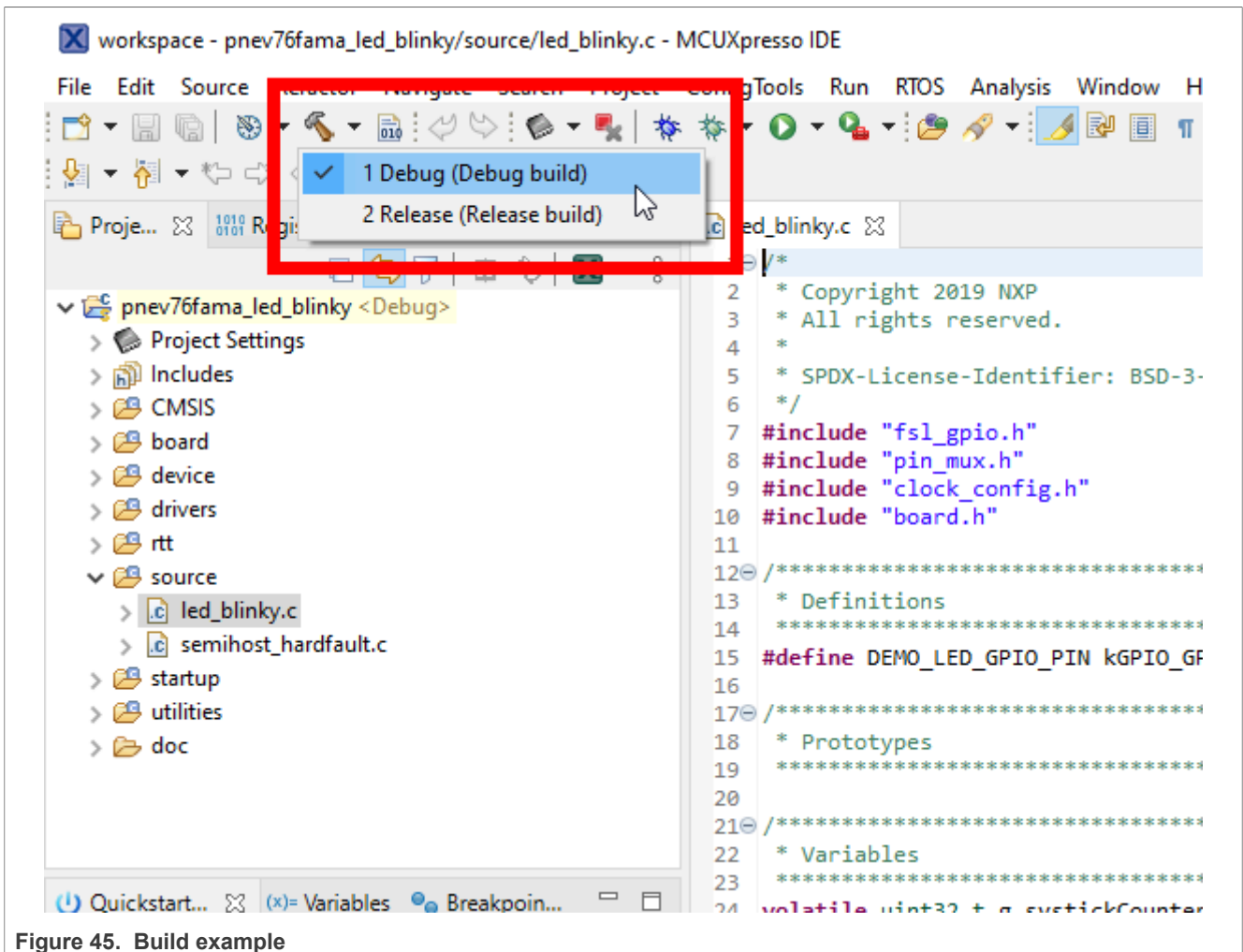
You can also choose the active build configuration by right-click on the project → Build Configurations → Set Active → Debug/Release.

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

Figure 46. Build finished successful
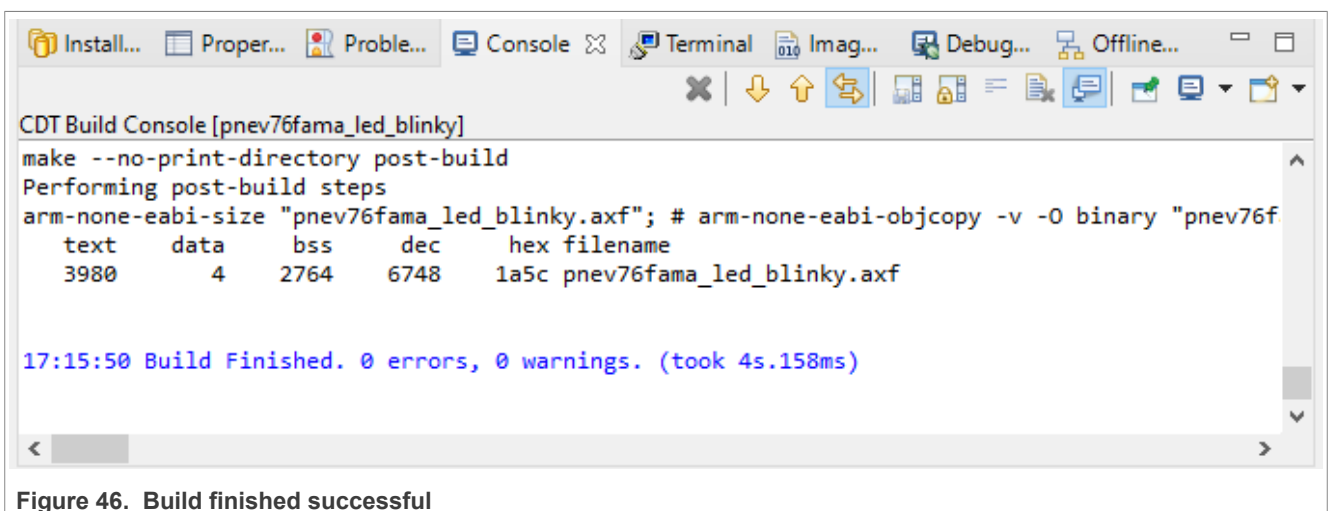
Now you can either flash or debug the build project.

AN13134

Application note

All information provided in this document is subject to legal disclaimers.

Rev. 2.7 — 6 December 2023

© 2023 NXP B.V. All rights reserved.

45 / 67

5.1.3.1 Flash example

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

Figure 47. GUI Flash Tool button
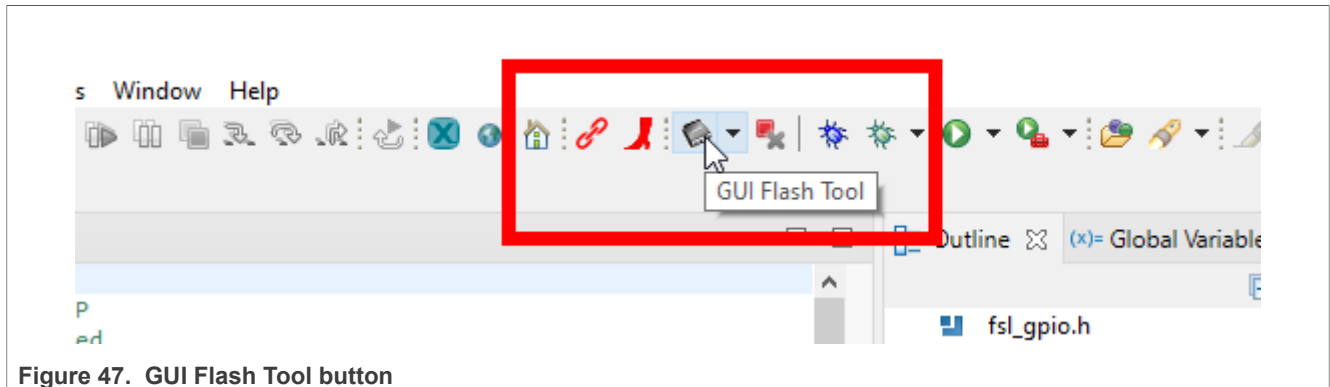
MCUXpresso searches available programmers like a LPC-Link2 or SEGGER J-Link now. In case multiple probes are detected, chose the one connected to the NFC-Debug port on the PNEV7642A Rev-B development board and click "OK".

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

46 / 67
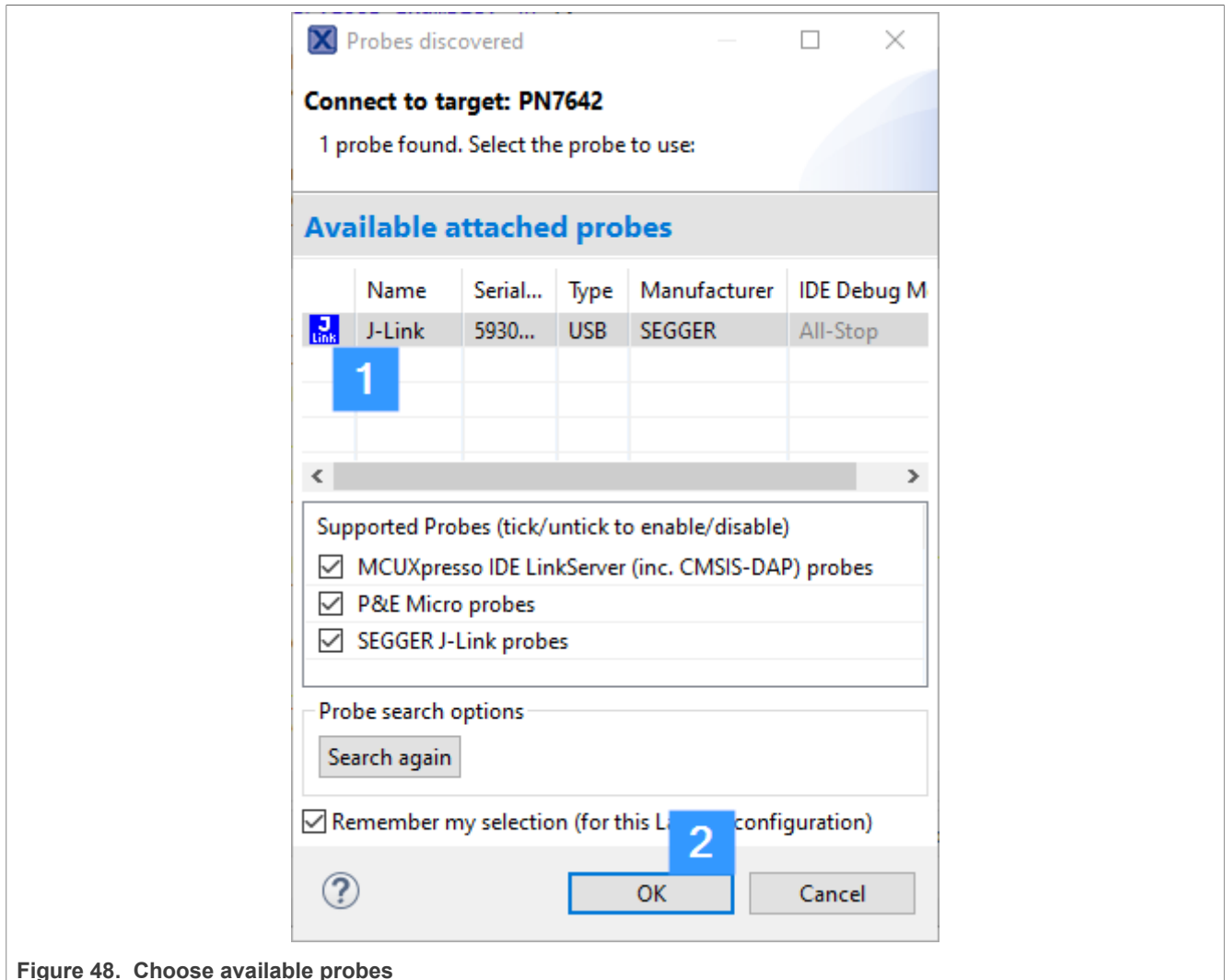
Figure 48. Choose available probes

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.
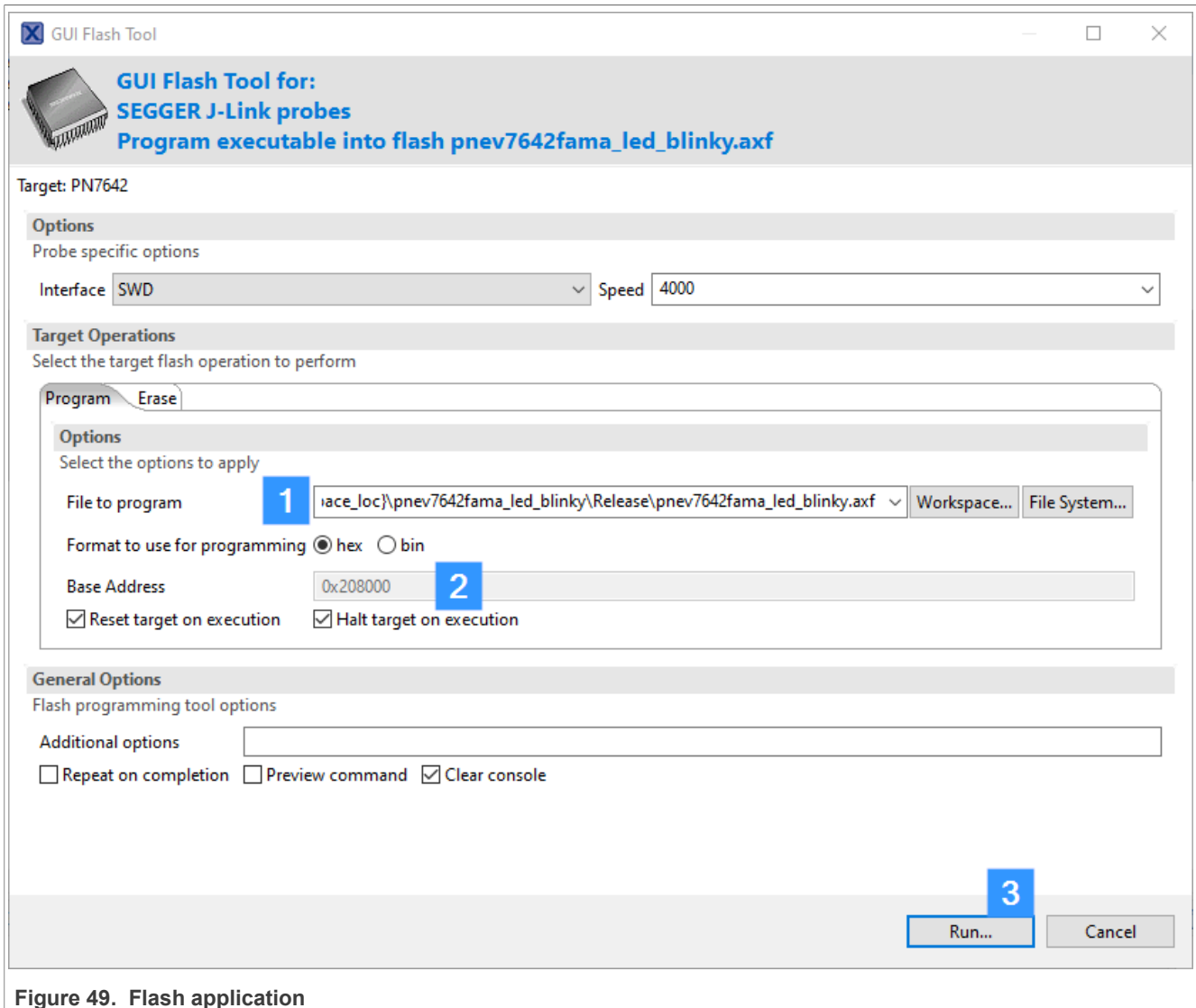
Figure 49. Flash application

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.
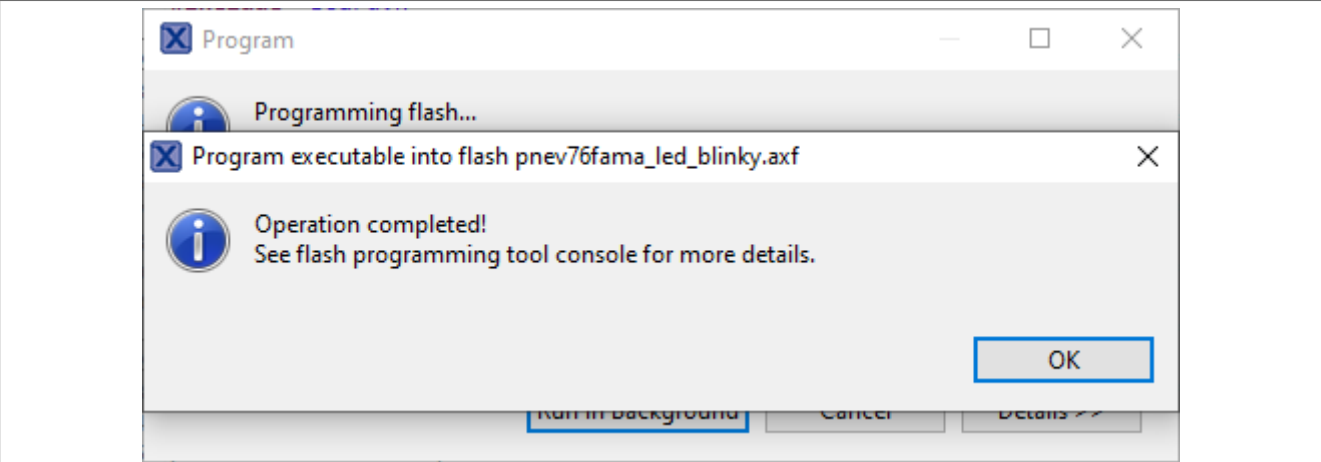
Figure 50. Flash successful

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

49 / 67

5.1.3.2 Debug example

If you build your example in "Debug" configuration, start debugging by clicking "Start debugging..." in the top toolbar.
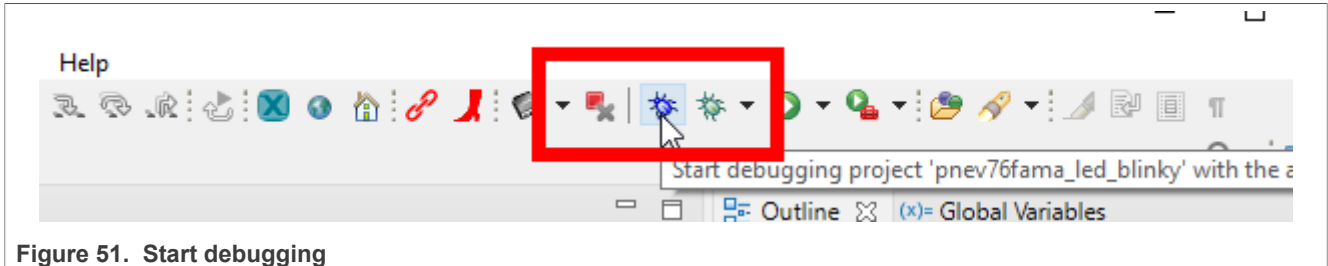
Figure 51. Start debugging

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 52. Debug toolbar

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

50 / 67
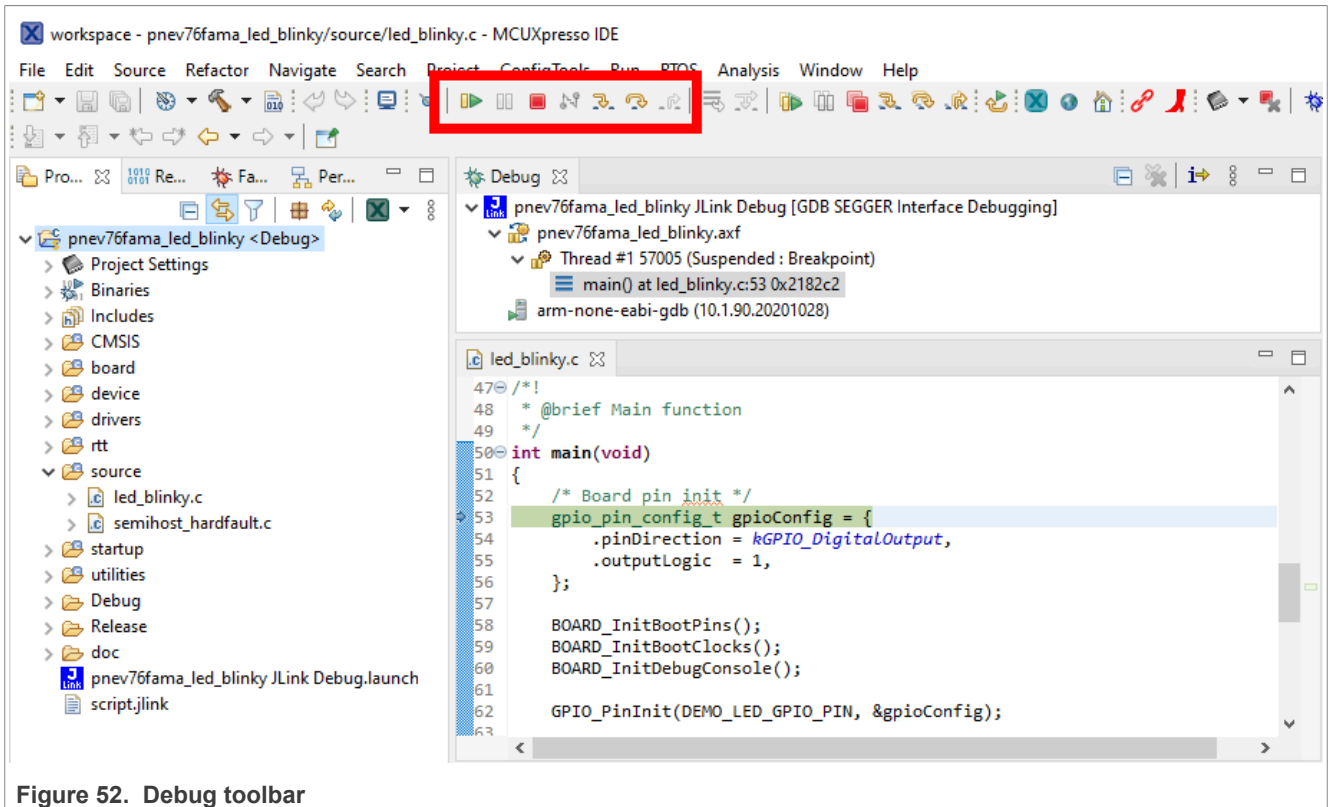
5.1.4 Bin file generation

To run an application standalone on the chip, it must be compiled in 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
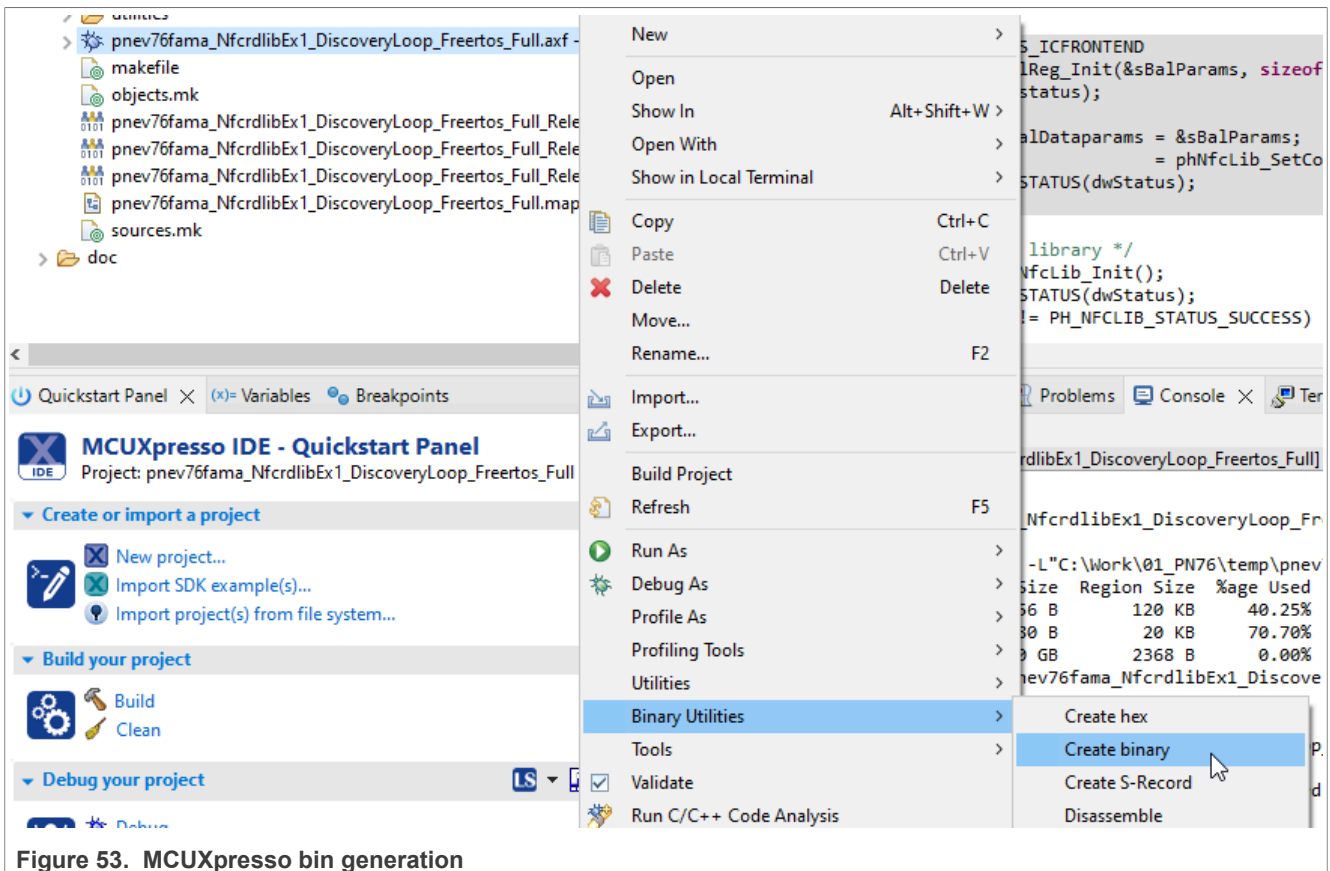
Figure 53. MCUXpresso bin generation

AN13134

Application note

All information provided in this document is subject to legal disclaimers.

Rev. 2.7 — 6 December 2023

© 2023 NXP B.V. All rights reserved.

51 / 67

6 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.

Link: [7]

This chapter describes how to configure, set-up, and use 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 there the Table 4 regarding valid combinations.

While the NFC Cockpit GUI supports all NFC Cockpit applications on the PN7642, the NFC Cockpit application put onto the PN7642 has to match the PN7642s firmware.

NFC Cockpit is using VCOM and for this you have to set the host-interface to use USB. The NFC Cockpit counterpart integration on the PN7642 is also available as source code: [8]

This VCOM library can be integrated in a users' application to provide the full NFC cockpit experience for configuration of the users' 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 adviced to do. Read Section 1.2 "Firmware overview" careful before doing a firmware update.

6.1 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 54. NFC cockpit folders
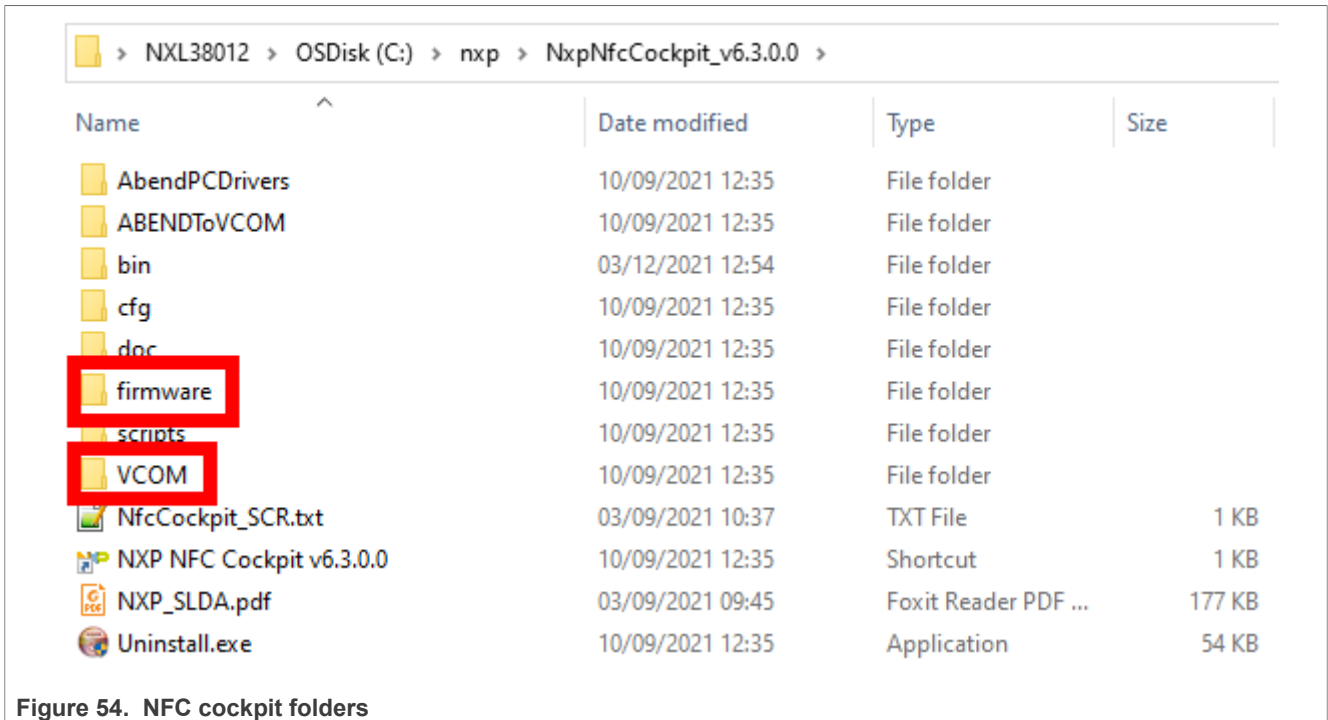
Table 17. NFC Cockpit folder content

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 1.2.
VCOM	'install_PN76xx_vcom.bat', which installs the proprietary PN7642 VCOM driver.

6.2 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).

6.3 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 3.1.4.1 "Host interface selection").

6.3.1 Bring to mass storage mode

To bring the PN76 development board into mass storage mode, we have to bring the DWL_REQ pin HIGH while resetting the board.

This is done by:

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 shall appear in your explorer, which holds two files "CRP_00.BIN" and "CRPSTA_3.BIN".
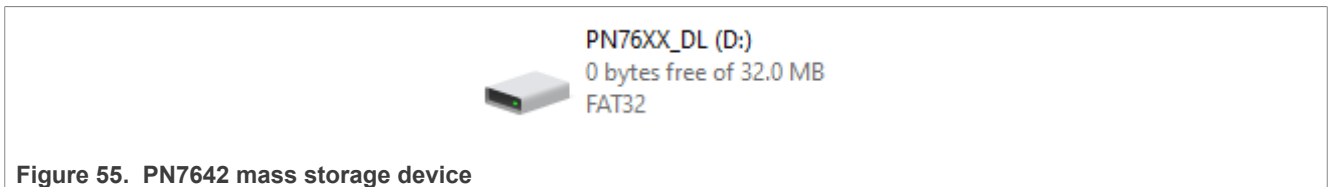
Figure 55. PN7642 mass storage device
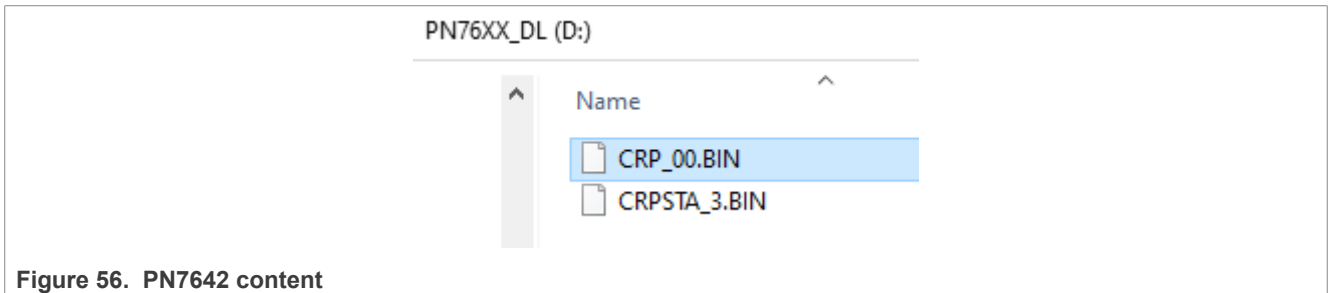
Figure 56. PN7642 content

6.3.2 Flash NFC Cockpit firmware

1. Open the PN76xx_DL mass storage device and delete the "CRP_00.BIN" file.
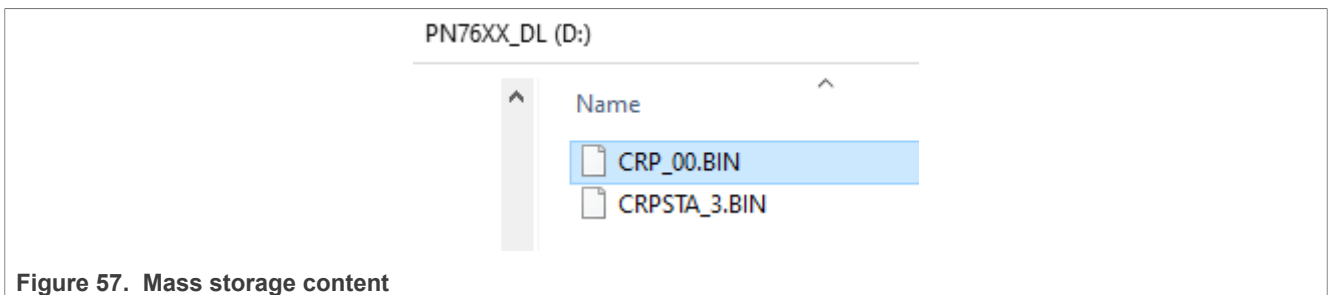
Figure 57. Mass storage content

2. Go to your NFC Cockpit installation folder → firmware → PN7642 and copy the "NxpNfcCockpit_05_03_00.bin"[2] onto the mass storage device.

---

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

Note: Check Section 1.2 "Firmware overview" to find the matching NFC Cockpit application version to your PN7642s firmware version.

For the PN7642, use the PN7642 folder and firmware.
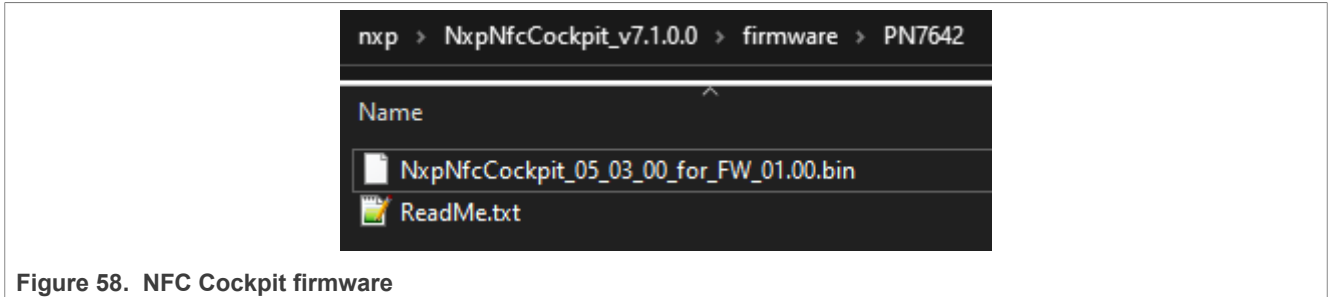
Figure 58. NFC Cockpit firmware

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.
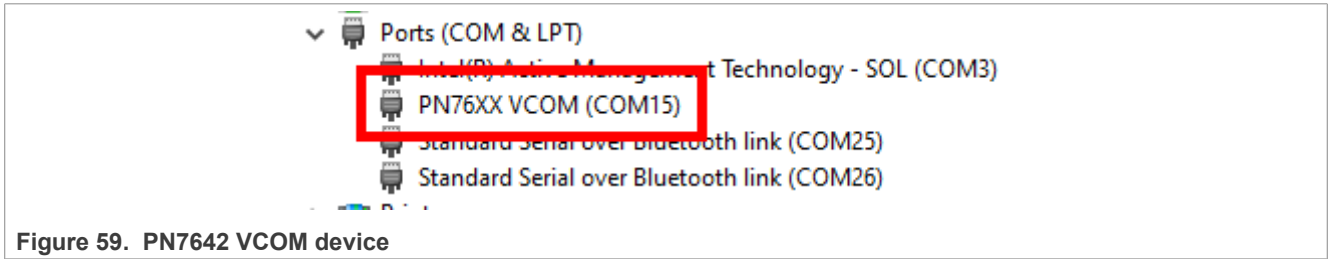
Figure 59. PN7642 VCOM device

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.

6.4 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.
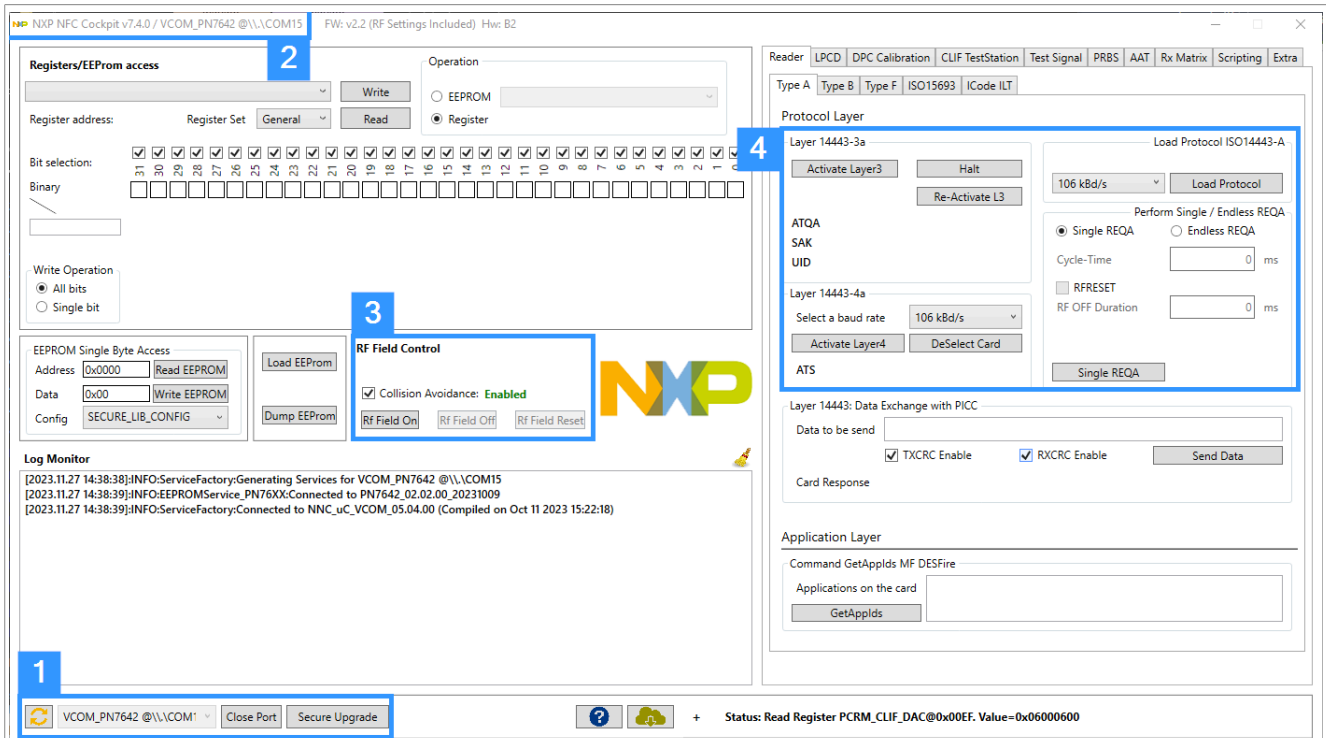
Figure 60. NFC Cockpit overview

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.

7 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 per external power supply or USB.
   a. See Section 3.1.1 "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 3.1.2 "Debug interfaces"
3. MCUXpresso installed
   a. See Section 5.1.1 "MCUXpresso IDE"
4. PN7642 SDK installed
   a. See Section 5.1.1.2 "Install SDK"
5. PNEV7642 jumper settings according to your 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 3.1.4 "Jumper settings"
6. Firmware version of your 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 with the correct SDK.
   b. See Section 1.2 "Firmware overview"

8 Troubleshooting

A short list of the most common issues and its solutions.

Table 18. Troubleshooting

Cause	Comment
IOREF	Depending on the usage of the board (standalone, follower or leader) the jumper IOREF has to be set correctly. Section 3.1.4.2 "IOREF"
HIF selection	Make sure to set the HIF switches according to the host-interface of your choice (USB, I2C, SPI, etc.). Section 3.1.4.1 "Host interface selection"
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 3.1.1.2 "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. Section 3.1.1.1 "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 1.2 "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.

9 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 19. 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.

10 References

[1]   AN13898 OpenSDA on PN76 development boards: nxp.com → PN7642 → Documentation

[2]   Base Board Schematics: nxp.com → PN7642 Development Kit → Design Resources

[3]   Module Board Schematics: nxp.com → PN7642 Development Kit → Design Resources

[4]   LPC55S16 Development Board: nxp.com → LPC55S16

[5]   MCUXpresso IDE: nxp.com → MCUXpresso IDE

[6]   LPC55S16 Host Software: nxp.com → PN7642 → Software

[7]   NFC Cockpit: nxp.com → NFC Cockpit

[8]   PN76 VCOM LibNncFw: nxp.com → PN7642 → Software

[9]   PN7642 development board: nxp.com → PN7642 development kit

[10]   PN7642 data sheet: nxp.com → PN7642 Datasheet

[11]   AN13925 PN76 EEPROM configuration handling: nxp.com → PN7642 → Documentation

[12]   AN13467 PN76 family antenna design guide: nxp.com → PN7642 → Documentation

[13]   AN13996 How to use the low-power features of the PN76 family NFC controller: nxp.com → PN7642 → Documentation

[14]   AN14060 How to use the PN76 family cryptographic features: nxp.com → PN7642 → Documentation

[15]   AN13720 PN7642 Secure Key Mode demo application: nxp.com → PN7642 → Documentation

11 Revision history

Revision history

Rev	Date	Description
v2.7	20231206	• Added Section 2 "Quick start with USB as supply" • Added Section 3.2 "PNEV7642MA" • Editorial changes: – Consistent naming of PN7642 – Restructuring of chapters to have a better reading flow.
v2.6	20231018	• Added section Setup without DC-DC • Added section Firmware overview – Added multiple references to this chapter in applicable chapters. • Added section Documention overview • Updated section NFC Cockpit
v2.5	20230502	• Deleted old boards and environments not relevant for the official released version. • Added section Checklist • Added section Troubleshooting.
v2.4	20230313	• Added section TDAEV8035 • Added section MCUXpresso SDK • Updated section Radio Equipment Directive (RED) • Security status changed into public
v2.3	20221201	• 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.
v2.2	20220818	• Section IOREF added
v2.1	20220809	• 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
v2.0	20220215	• 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
v1.6	20220118	Editorial changes
v1.5	20211209	• Add NFC Cockpit chapter • Enhance system overview
v1.4	20211201	• Add system overview • Change to new revision D PNEV76FAMA board • Add important jumper settings • Add MCUXpresso chapter
v1.3	20210721	Change from PN5190 RF Performance board to PNEV76FAMA board.
v1.2	20210422	• PN76 wording changed • PN76 family software package contents added

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

61 / 67

Revision history*...continued*

Rev	Date	Description
v1.1	20210319	Change to GCC version
v1.0	20210211	Initial version

AN13134

Application note

Rev. 2.7 — 6 December 2023

62 / 67

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.

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

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

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

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

Licenses

Trademarks

AN13134

All information provided in this document is subject to legal disclaimers.

© 2023 NXP B.V. All rights reserved.

Application note

Rev. 2.7 — 6 December 2023

64 / 67

Tables

Figures

Contents