AN11802

NFC Reader Library for Linux installation guidelines

Rev. 1.8 — 21 December 2022 Application note

Document information

Information	Content
Keywords	NFC, Reader Library, Linux, PN5180, CLRC663, PN5190, Raspberry Pi
Abstract	This note describes how to install NXP’s NFC Reader Library on a GNU/Linux system.

Revision history

Revision history

Rev	Date	Description
1.8	20221221	• Added kernel-irq (PN5190) resource location in Section 4.5 • Corrected build steps for kernel-irq driver in Section 4.5
1.7	20220321	• Alignment on latest SW delivery v07.03.00 • Support for PN5190 added • Corrected wrong connection between PN5180 and Raspberry Pi • Detailed explanation on adding kernel-bal (PN5180) and kernel-irq (PN5190)
1.6	20180419	Editorial updates
1.5	20170907	Correction of missing reference
1.4	20170516	Alignment on latest SW delivery v05.02.00 Updated reference platform
1.3	20170104	Alignment on latest SW delivery v4.050.03.001651
1.2	20161115	Alignment on latest SW delivery v4.040.05.011646 Updated examples descriptions
1.1	20160418	Corrected wrong connection between CLEV663B and Raspberry Pi
1.0	20160404	First release

1 Introduction

The NFC Reader Library is a feature-complete software support library for NXP's NFC frontend ICs. It is designed to give developers a faster and simpler way to deliver NFC-enabled products. This multi-layer library, written in C, makes it easy to create NFC-based applications. See [1] for more details.

The purpose of the present document is to give instructions on how to install the NFC Reader Library on a generic GNU/Linux platform.

It takes as reference the support of CLEV6630B board on Raspberry Pi platform. The reference environment is further described in Section 2.

In below chapters, information highlighted thanks to surrounded borders relates to examples on this reference platform.

Finally, it gives details about modifications to be done to make the porting of the NFC Reader Library for Linux to another Linux platform than the one used as reference, or for others NXP NFC frontend ICs support.

2 Reference environment

2.1 Overview

The selected reference environment is the CLEV6630B v2.0 board (see [6]), including CLRC663 NXP's NFC Frontend (see [4]), connected to Raspberry Pi platform (refer to [2] for more details) running Raspbian Bullseye Linux distribution.

Selecting RC663 (refer to Section 3 for more information about RC663) is done defining adding **`OPTION(FRONTEND_RC663 "IC Frontend is RC663" ON)`** parameter to the cmake command, additionally parameter **`OPTION(FRONTEND_PN5190 "IC Frontend is PN5190" OFF) and OPTION(FRONTEND_PN5180 "IC Frontend is PN5180" OFF)`** must also be defined to unselect PN5180 and PN5190 use (only one frontend can be selected at a time).

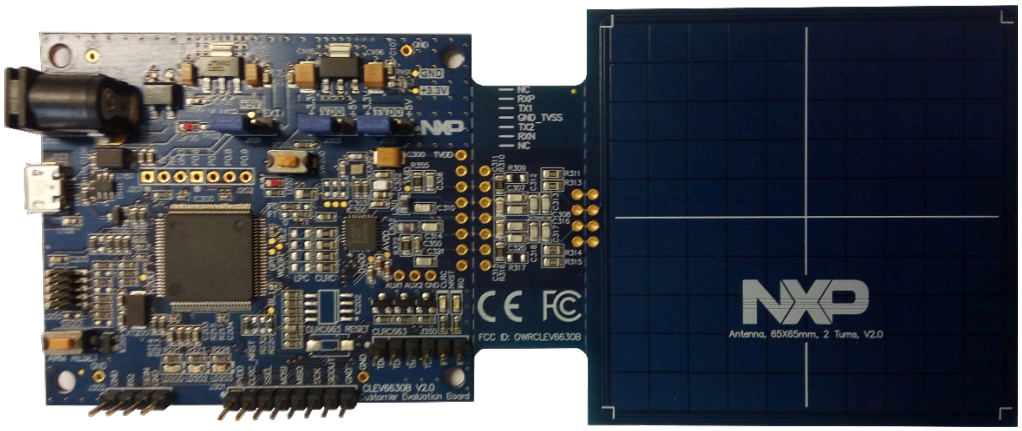

Figure 1. CLEV6630B v2.0 board

2.2 Boards connections

Using CLEV6630B board connection to the Raspberry Pi must be done following definition in Table 1 and Figure 2.

The CLEV6630B board must have been previously set into the proper configuration as indicated in the related document Section 5.

Table 1. CLEV663B connection to Raspberry Pi

Wiring to be done to connect CLEV6630B demo board to Raspberry Pi

CLEV6630B pin	Raspberry Pi	
	Pin number	Pin function
MOSI	19	SPI-MOSI (GPIO 10)
MISO	21	SPI-MISO (GPIO 9)
SCK	23	SPI-CLK (GPIO 11)
SSEL	24	SPI-CE0 (GPIO 8)
CLRC_NRST	18	GPIO 24
IRQ	16	GPIO 23

Table 1. CLEV663B connection to Raspberry Pi*...continued*

Wiring to be done to connect CLEV6630B demo board to Raspberry Pi

CLEV6630B pin	Raspberry Pi	
	Pin number	Pin function
IFSEL0	13	GPIO 27
IFSEL1	15	GPIO 22
GND	6, 9, 14, 20, 25, 30, 34 or 39	GND

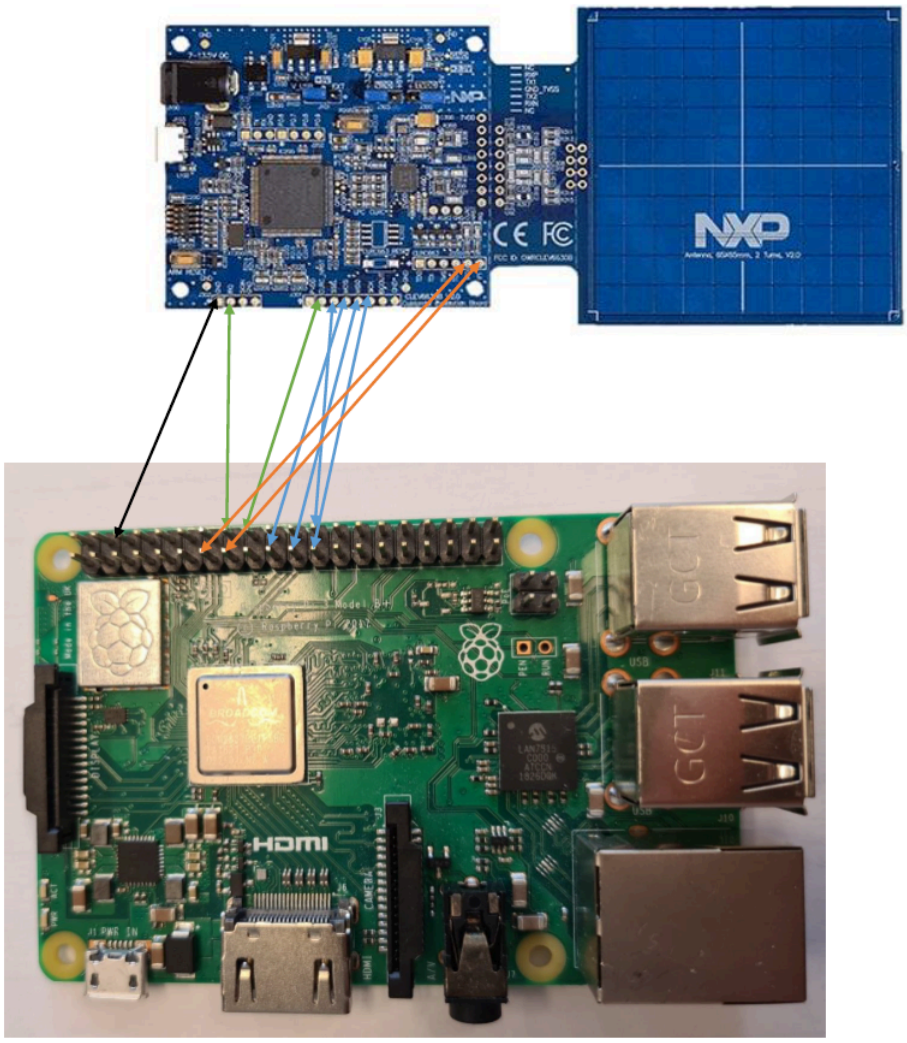

Figure 2. CLEV6630B connection to Raspberry Pi

3 NFC Reader Library for Linux delivery

3.1 Step 1: Installing NFC Reader Library for Linux delivery

The NFC Reader Library delivery consists of a zip file to be uncompressed on the Linux target (or on host machine in case of cross compilation):

```
$ unzip sw3693.zip (name of zip file is different for each version)
```

This creates the following directory structure in the current path:

```
├── docs
│   └── NxpNfcRdLib
│   └── DAL
│   └── OSAL
├── Examples
│   └── NfcrdlibEx1_DiscoveryLoop
│   └── NfcrdlibEx3_NFCForum
│   └── NfcrdlibEx4_MIFAREClassic
│   └── NfcrdlibEx5_ISO15693
│   └── NfcrdlibEx6_LPCD
│   └── NfcrdlibEx8_HCE_T4T
│   └── NfcrdlibEx9_NTagI2C
│   └── NfcrdlibTst12_Rc663Lpcd
│   └── Nfcrdlib_SimplifiedAPI_ISO
├── ExamplesPN5190
│   └── Nfcrdlib_PRBSApplication
├── ComplianceApp
│   └── Nfcrdlib_EMVCo_AnalogComplApp
│   └── Nfcrdlib_EMVCo_InteropComplApp
│   └── Nfcrdlib_EMVCo_LoopBackComplApp
├── NxpNfcRdLib
├── Platform
├── RTOS
├── CMakeLists.txt
└── NXP_SLDA.pdf
└── NXP_SCR.pdf
```

- {docs} contains the NxpNfcRdLib as well as DAL and OSAL layers Doxygen documentation, entry points being the html files
- {Examples} contains code examples of the NFC Reader Library use
- {ExamplesPN5190} contains code examples of the NFC Reader Library specific for PN5190.
- {ComplainceApp} contains code of applications for compliancy testing)
- {NxpNfcRdLib} is the NFC Reader Library source code
- {Platform} is the DAL layer source code (relates to RaspberryPi)
- {RTOS} is the OSAL layer source code (relates to Linux)
- {CMakeLists.txt} allows building makefiles structure
- {NXP_SLDA.pdf} describes the NXP Software License and Distribution Agreement of the current package.

3.2 Step 2: Generating makefiles structure

Generating the makefiles structure requires the following modules been installed on the target (or on host machine in case of cross compilation):

- gcc
- make
- cmake (version 2.8.11 minimum)

On reference platform (Raspberry Pi running Raspbian Bullseye), cmake installation is done using command:

- $ sudo apt-get install cmake

Makefiles generation is then done running cmake command, from the build sub-directory:

- $ mkdir _build
- $ cd _build
- $ cmake ..

3.3 Step 3: Enabling the SPI physical interface

The physical link used to interface the NFC Frontend must be enabled on the platform: node */dev/spidev* must be present and accessible.

On the Raspberry Pi, enabling SPI is done through raspi-config tool:

```
$ sudo raspi-config
```

The option to activate SPI can be found in: Advanced Options ➔ SPI ➔ <Yes>. Then reboot is required to take into account the change:

```
$ sudo reboot
```

3.4 Step 4: Building and running the examples

Building the examples (including the NFC Reader Library source code) is then just done with the simple *make* command from the *_build* directory:

```
$ cd ../_build
$ make all
```

Examples can also be individually built from the dedicated subdirectory:

```
$ cd ../_build/Examples/NfcrdlibEx<#_ExampleName>
$ make
```

This generates **NfcrdlibEx**<*#_ExampleName*> example binary in the current directory.

Running the example is done executing the related command:

```
$ ./NfcrdlibEx<#_ExampleName>
```

Below is the output obtained running *NfcrdlibEx1_DiscoveryLoop*, tapping an NFC tag:

```
$ ./NfcrdlibEx1_DiscoveryLoop
```

```
DiscoveryLoop Example:
Card detected and activated successfully...
Technology : Type A
Card: 1
UID : 04 60 32 6A 64 34 80
SAK : 0x00
Type: Type 2 Tag
```

4 Porting of the NFC Reader Library for Linux

By default, the NFC Reader Library for Linux delivery is suitable to run the PN5190 platform. In case of different setup, some adaptations are required.

4.1 Support of CLRC663

The way CLRC663 is connected is defined in *Board_PiRc663.h* file located under *Platform/DAL/boards* subfolder of the delivery. In case of different connection than the one depicted at Section 2, this has to be reflected there.

4.2 Support of PN5180

Selection of the NFC Frontend IC is done when generating the makefiles structure (see Step 2 described at Section 3.2).

Selecting PN5180 [3] (refer to Section 3 for more information about PN5180) is done defining adding **`OPTION(FRONTEND_PN5180 "IC Frontend is PN5180" ON)`** parameter to the cmake command, additionally parameter **`OPTION(FRONTEND_PN5190 "IC Frontend is PN5190" OFF) and OPTION(FRONTEND_RC663 "IC Frontend is RC663" OFF)`** must also be defined to unselect RC663 and PN5190 use (only one frontend can be selected at a time).

Using the PNEV5180B board connection to the Raspberry Pi must be done according to Table 2 and Figure 3.

The PNEV5180B board must have been previously set into the proper configuration as indicated in the related document Section 6.

Table 2. PNEV5180 connection to Raspberry Pi

Wiring to be done to connect PNEV5180 demo board to Raspberry Pi

PNEV5180 pin	Raspberry Pi	
	Pin number	Pin function
MOSI	19	SPI-MOSI (GPIO 10)
MISO	21	SPI-MISO (GPIO 9)
SCK	23	SPI-CLK (GPIO 11)
NSS/SSEL	24	SPI-CE0 (GPIO 8)
nRESET/PN_RST	36	GPIO 16
BUSY	22	GPIO 25
IRQ	16	GPIO 23
DWL	6, 9, 14, 20, 25, 30, 34 or 39	GND
GND	6, 9, 14, 20, 25, 30, 34 or 39	GND

In case different connection is defined, this must be reflected in *Board_PiPn5180.h* file located under *Platform/DAL/boards* subfolder of the delivery.

Figure 3. PNEV5180 connection to Raspberry Pi

4.3 Support of PN5190

Selection of the NFC Frontend IC is done when generating the makefiles structure (see Step 2 described at Section 3.2).

Selecting PN5190 [5][8] (refer to Section 3 for more information about PN5190) is done defining adding **`OPTION(FRONTEND_PN5180 "IC Frontend is PN5190" ON)`** parameter to the cmake command, additionally parameter **`OPTION(FRONTEND_PN5190 "IC Frontend is PN5180" OFF) and OPTION(FRONTEND_RC663 "IC Frontend is RC663" OFF)`** must also be defined to unselect RC663 and PN5180 use (only one frontend can be selected at a time).

Using the PNEV5190 board connection to the Raspberry Pi must be done according to Table 3 and Figure 4.

The PNEV5180B board must have been previously set into the proper configuration as indicated in the related document. Section 7

Table 3. PNEV5190 connection to Raspberry Pi

Wiring to be done to connect PNEV5190 demo board to Raspberry Pi

PNEV5190 pin	Raspberry Pi	
	Pin number	Pin function
MOSI	19	SPI-MOSI (GPIO 10)
MISO	21	SPI-MISO (GPIO 9)
SCK	23	SPI-CLK (GPIO 11)
SSEL	24	SPI-SSEL (GPIO 8)
VEN	18	GPIO24
IRQ	16	GPIO 23
GND	6, 9, 14, 20, 25, 30, 34 or 39	GND

In case different connection is defined, this must be reflected in *Board_PiPn5190.h* file located under *Platform/DAL/boards* subfolder of the delivery.

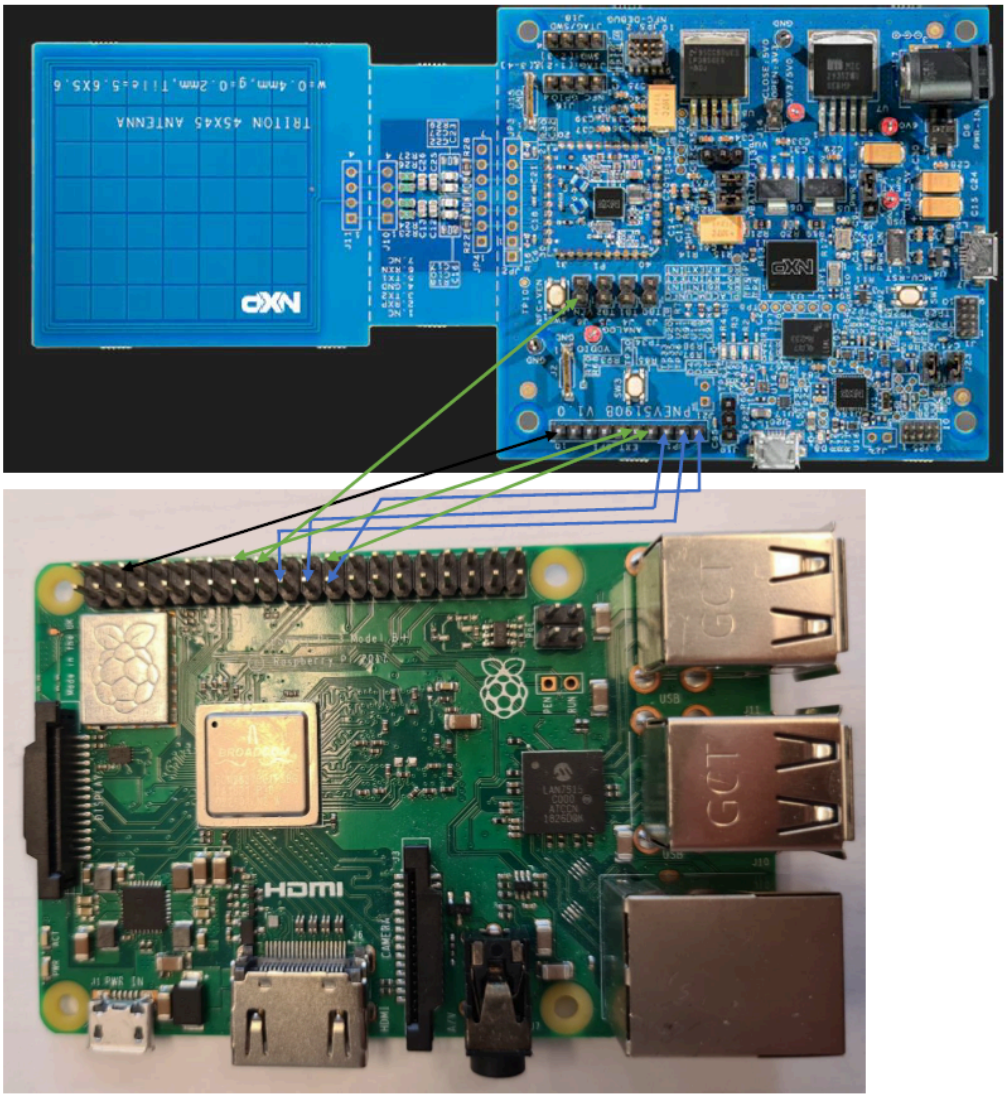

Figure 4. PNEV5190 connection to Raspberry Pi

4.4 Support of “bal” kernel module

!!! Only PN5180 is currently supported by the current “bal” kernel module!!!

To insure critical timings (for instance for EMVCo compliancy) on slow platforms, use of “bal” kernel module is recommended. It abstracts SPI access which significantly speed-up the communication time between the Linux platform and the NFC Frontend IC.

Source code of the “bal” kernel module is provided on the following GitHub repository: https://github.com/NXPNFCLinux/nxprdlib-kernel-bal. All information to build and install this kernel module on the Linux platform is described in the related *README.md* file present in this repository.

When properly loaded on the Linux platform, the module should expose */dev/bal* device node.

To make use of this kernel module related **`PHDRIVER_LINUX_KERNEL_SPI`** definition must be uncommented inside *Board_PiPn5180.h*, located under *Platform/DAL/boards* subfolder of the delivery. Other definition **`PHDRIVER_LINUX_USER_SPI`** must be commented out.

Please follow next steps:

- $ sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev libsystemd-dev flex bison
- $ sudo apt install git bc bison flex libssl-dev make libc6-dev libncurses5-dev
- $ sudo wget https://raw.githubusercontent.com/notro/rpi-source/master/rpi-source -O / usr/bin/rpi-source
- $ sudo chmod +x /usr/bin/rpi-source
- $ /usr/bin/rpi-source -q --tag-update
- $ rpi-source → This will create /home/pi/linux
- $ cd /home/pi/linux
- $ KERNEL=kernel7
- $ make bcm2709_defconfig
- $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2709_defconfig
- $ cd /home/pi
- $ git clone https://github.com/NXPNFCLinux/nxprdlib-kernel-bal.git
- $ cd /home/pi/linux
- $ make M=../nxprdlib-kernel-bal/bal
- $ cd /home/pi/nxprdlib-kernel-bal
- $ dtc -I dts -O dtb -o bal-overlay.dtbo -@ bal-overlay.dts
- $ sudo dtoverlay -d . bal-overlay
- $ sudo insmod /home/pi/nxprdlib-kernel-bal/bal/bal.ko
- After that, we check if bal existing in /dev
  - $ dmesg
  - $ ls -als /dev/bal

4.5 Support of “irq” kernel module

!!! Only PN5190 is currently supported by the current “irq” kernel module!!!

Please follow next steps for adding irq kernel module:

- $ sudo apt-get install cmake cmake-curses-gui cmake-gui libssl-dev libsystemd-dev flex bison python2 raspberrypi-kernel-headers
- $ sudo apt install git bc bison flex libssl-dev make libc6-dev libncurses5-dev
- $ sudo apt-get update
- $ sudo apt-get upgrade
- $ sudo wget https://raw.githubusercontent.com/RPi-Distro/rpi-source/master/rpi-source -O /usr/local/bin/rpi-source && sudo chmod +x /usr/local/bin/rpi-source && /usr/local/bin/rpi-source -q --tag-update
- $ rpi-source → this will create /home/pi/linux
- $ cd /home/pi/linux
- $ KERNEL=kernel7
- $ make bcm2709_defconfig
- $ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf- bcm2709_defconfig
- Download IRQ Kernel driver from https://github.com/NXPNFCLinux/nxprdlib-kernel-irq.git and put it here"/home/pi/".
- Build IRQ kernel driver (currently IRQ Pin is fixed in Kernel Driver to GPIO23):
  - $ cd /home/pi/linux
  - $ make M=../nxprdlib-kernel-irq/irq
- $ sudo insmod /home/pi/nxprdlib-kernel-irq/irq/irq_poll.ko
- $ sudo mknod /dev/irq c 101 0
- After that, we check if irq existing in /dev
  - $ dmesg
  - $ ls -als /dev/irq

5 Annex 1: CLEV6630B rework for direct access

In order to assure direct access to the CLRC663 frontend IC (bypassing LPC1769) on the CLEV6630B board, the following rework of the CLEV6630B v2.0 has to be done:

In total, four resistors (R362/R364/R366/R368) need to be removed to obtain proper decoupling of the LPC1769 MCU from the CLEV6630B board. See on Figure 6 resistors to be removed marked red.

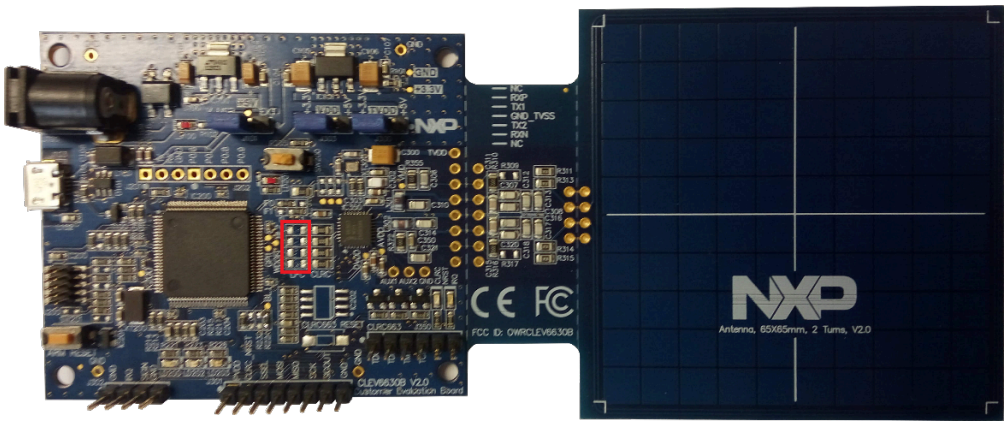

Figure 5. CLEV663 board rework

To check for the correct CLEV663B board configuration, refer to [6].

The Power of CLEV663B board still come from either the USB or the 5 V Power connectors.

6 Annex 2: PNEV5180B rework for direct access

In order to assure direct access to the PN5180 frontend IC (bypassing LPC1769) on the PNEV5180B board, the following rework of the PNEV5180B v2.0 has to be performed:

In total, six resistors in two groups need to be removed to obtain proper decoupling of the LPC1769 MCU from the PNEV5180 board. See on Figure 6 resistors to be removed marked red.

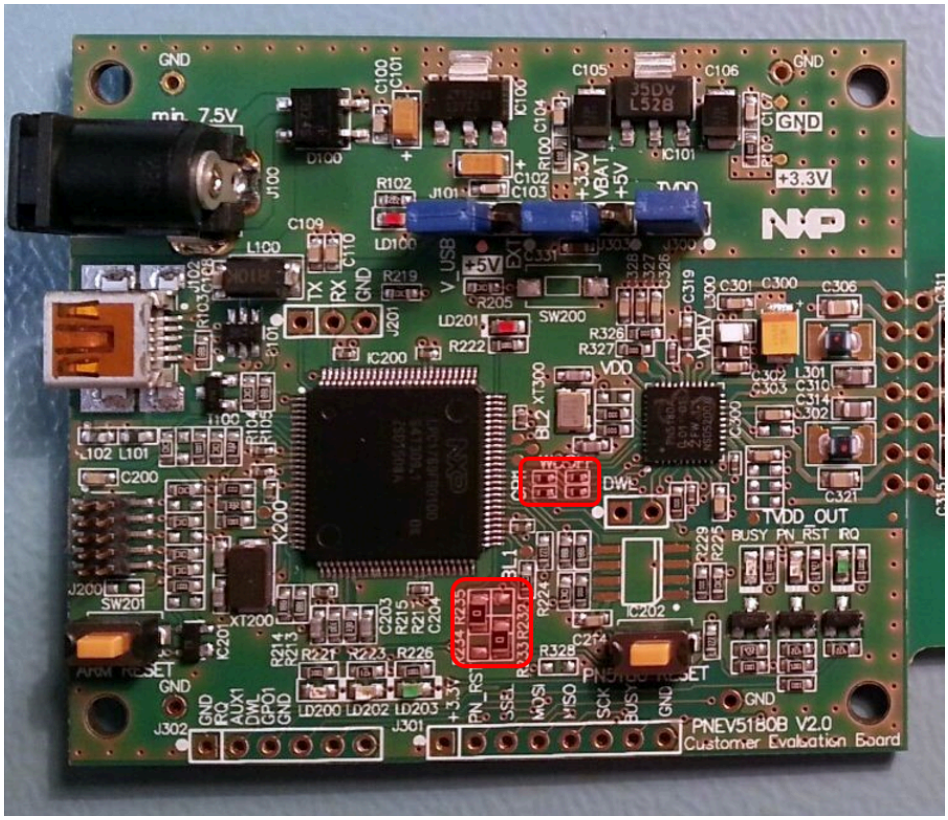

Figure 6. PNEV5180B board rework

To check for the correct PNEV5180B board configuration, refer to [7].

The Power of PNEV5180 board still come from either the USB or the 5 V Power connectors.

7 Annex 3: PNEV5190 rework for direct access

In order to assure direct access to the PN5190 frontend IC (bypassing K82) on the PNEV5190 board, the following rework of the PNEV5190 has to be performed:

In total, two resistors should be removed and two resistors should be added. See Figure 7.

R5 resistor should be removed (red square) and R7 should be added (green square).

R19 resistor should be removed (red square) and R20 should be added (green square).

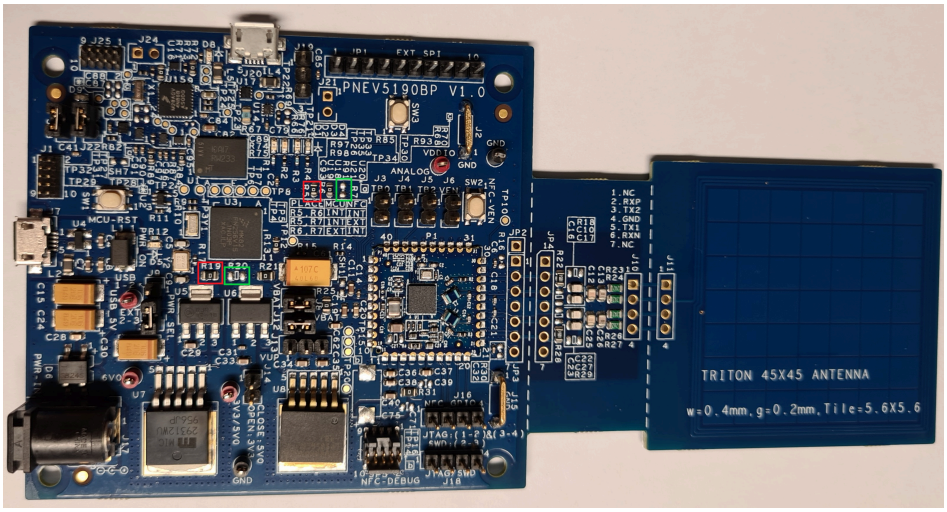

Figure 7. PNEV5190B board rework

The Power of PNEV5180 board still come from either the USB or the 5 V Power connectors.

8 References

[1] The NFC Reader Library gives Software support for NFC Frontend solutions. For more information about it, visit NFC Reader Library.

[2] The Raspberry Pi is a credit card sized computer. To get started quickly, the Raspberry Pi Foundation provides several preconfigured Linux distributions. For more information about it, visit http://www.raspberrypi.org/

[3] PN5180 is a highly integrated high performance full NFC Forum-compliant frontend IC for contactless communication at 13.56 MHz. For more information about it, visit PN5180.

[4] CLRC663 *plus*, the high performance multiprotocol NFC Frontend. For more information about it, visit CLRC663 *plus* Family

[5] PN5190 is a robust solution for payment terminals and all readers that need to generate a strong RF field in a difficult environment. For more information about it, visit PNEV5190.

[6] AN11022 CLRC663 evaluation board quick start guide

[7] UM10954 PN5180 SW quick start guide

[8] AN12550 PNEV5190B evaluation board quick start guide

9 Legal information

9.1 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.

9.2 Disclaimers

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

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.

9.3 Licenses

Purchase of NXP ICs with NFC technology — Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

9.4 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.
EdgeVerse — is a trademark of NXP B.V.
FeliCa — is a trademark of Sony Corporation.
MIFARE — is a trademark of NXP B.V.
MIFARE Classic — is a trademark of NXP B.V.

Tables

Figures

Contents

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

Date of release: 21 December 2022
Document identifier: AN11802