Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

UM10954

PN5180 SW Quick start guide

Rev. 1.6 — 7 May 2018
345816

Document information

Info	Content
Keywords	PN5180, PN5180 SW design, PNEV5180B, NFC NXP Cockpit
Abstract	This user manual is related to the installation procedures of the PN5180 Evaluation board, which are related to the installation of the SW sample projects as well as the re-installation of the original LPC firmware to run the NFC Cockpit. It describes the steps to be done to become acquainted with the demo reader especially for SW development.

Revision history

Rev	Date	Description
1.6	20180507	Editorial updates
1.5	20170511	MCUXpresso IDE description added
1.4	20170117	Updated description how to flash FW for the NFC Cockpit tool.
1.3	20170105	Updated examples descriptions, reworked firmware update
1.2	20161124	Updated examples descriptions
1.1	20160803	Note in section 5 regarding the LPCXpresso version added HCE, NFC Forum and MIFARE DESFire added to the Associated projects Cockpit version changed from 2.2 to 2.3 Required LPCXpresso version changed from 7.9 to 8.1.4 RTOS options added
1.0	20151126	Initial version

Contact information

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

UM10954

User manual **1.6 — 7 May 2018** **2 of 32**
COMPANY PUBLIC **345816**

1. Introduction

This document is the continuation of the "AN11744 - PN5180 Quick start guide" and describes the installation procedures of the SW development environment and handling SW example projects using the NFC Reader Library prepared for the PN5180 evaluation board.

It also describes how to re-install the original LPC firmware binary to use the NFC Cockpit again.

In this document the term „MIFARE Classic card" refers to a MIFARE Classic IC-based contactless card, the term "MIFARE DESFire card" refers to a MIFARE DESFire IC-based contactless card, the term "MIFARE Ultralight card" refers to a MIFARE Ultralight IC-based contactless card.

Projects used and explained in this documentation are:

Table 1.   Example projects
Example projects delivered with the NFC Reader Library

Example	Description
NfcrdlilbEx1_Basic DiscoveryLoop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. Default configuration parameters are used.
NfcrdlilbEx2_AdvancedDiscoveryLoop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. All configuration parameters are used and explained.
NfcrdlilbEx3_ NFCForum	Explains how to configure the NFC Reader Library for different P2P modes such as Active Mode, Target Mode, Initiator Mode and SNEP Client/Server.
NfcrdlilbEx4_MIFARE Classic	Explains the usage of MIFARE Classic card commands.
NfcrdlilbEx5_ ISO15693	Explains the usage of this technology and provides an overview about the most common commands.
NfcrdlilbEx7_ EMVCo_Polling	Explains polling for EMVCo payment cards.
NfcrdlilbEx8_ HCE_T4T	Explains how to emulate a NFC Forum Type 4 Tag supporting read and write operations.
NfcrdlilbEx9_ NTagI2C	Explains NTAG-I2C specific commands.
NfcrdlilbEx10_ MIFAREDESFire	Explains the usage of MIFARE DESFire cards. (This example is delivered with the NFC Reader Library version available via NXP DocStore)
NfcrdlibEx11_ISO10373_ PCD	Example is used to perform ISO 10373-6 PCD compliance validation.
Nfcrdlib_SimplifiedAPI EMVCo	EMVCo loopback application with simplified API, which can be used for EMVCo level 1 digital certification.
Nfcrdlib_SimplifiedAPI EMVCo_Analog	Application is used to perform EMVCo2.6(L1) Analog compliance validation.
Nfcrdlib_SimplifiedAPI ISO	Explains how to use simplified API with different types of cards.

2. Managing the PN5180 SW projects with MCUXpresso IDE

The PN5180 SW projects are delivered in a zip package and can be extracted, edited, compiled and linked with MCUXpresso IDE.

The MCUXpresso IDE is a low-cost highly integrated software development environment for NXP's LPC microcontrollers and includes all the tools necessary to develop high-quality software solutions in a timely and cost-effective fashion. MCUXpresso IDE is based on Eclipse and has many enhancements to simplify development with NXP LPC microcontrollers. It also features the industry-standard GNU tool chain, with a choice of a proprietary optimized C library or the standard "Newlib" library. The MCUXpresso IDE can build an executable of any size with full code optimization.

Designed for simplicity and ease of use, the MCUXpresso IDE provides software engineers a quick and easy way to develop their applications.

This tool can freely be downloaded from the MCUXpresso website [1]. Before one can download the software, it is necessary to create an account. Creating an account is free.

2.1 Development environment

To use PN5180 prepared software package all components listed in the Table 2 are required.

Table 2.   Development Environment

Item	Version	Description
PN5180EV5180B	1.0 or higher	PN5180 Customer Evaluation board (hardware)
LPC-Link 2	1.0	Standalone debug adaptor (hardware)
MCUXpresso IDE	10.0.0 or higher	Development IDE (PC software)

2.2 Installation procedure of the MCUXpresso IDE

The MCUXpresso IDE is installed into a single directory, of your choice. Unlike many software packages, the MCUXpresso IDE does not install or use any keys in the Windows Registry, or use or modify any environment variables (including PATH), resulting in a very clean installation that does not interfere with anything else on your PC. Should you wish to use the command-line tools, a command file is provided to set up the path for the local command window.

Multiple versions can be installed simultaneously without any issues.

The installation starts after double-clicking the installer file.
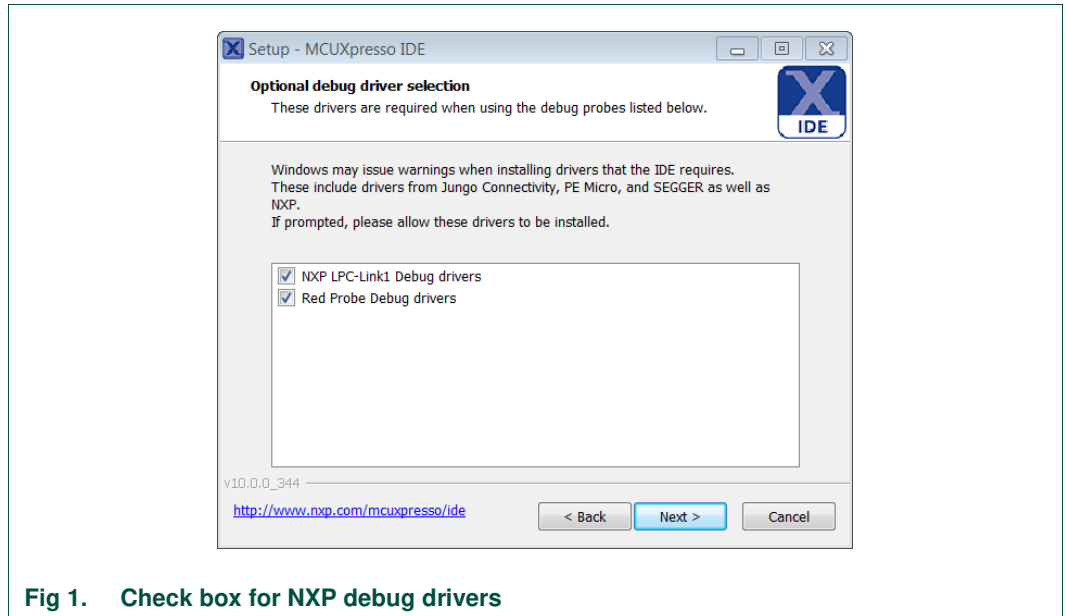
Fig 1.   Check box for NXP debug drivers

Make sure, the checkbox for installing the NXP debug drivers is activated.

During the installation, the user will be asked to install some required drivers. The installation of these drivers shall be accepted.

Fig 2.   Windows security dialog

After the setup wizard, has finished, the newly installed IDE can be launched.

Fig 3.   MCUXpresso IDE

2.3 Importing provided SW example projects

The use of quick start panel provides rapid access to the most commonly used features of the MCUXpresso IDE. Quickstart panel allows easy import projects, create new projects, build and debug projects.

The sequence of installing the software projects is indicated:

- Start the MCUXpresso IDE.
- Open new or dedicated workspace
- Select the option "Import project(s)" (see picture below).
- Browse the zip archive.
- MCUXpresso IDE unzips the software package.
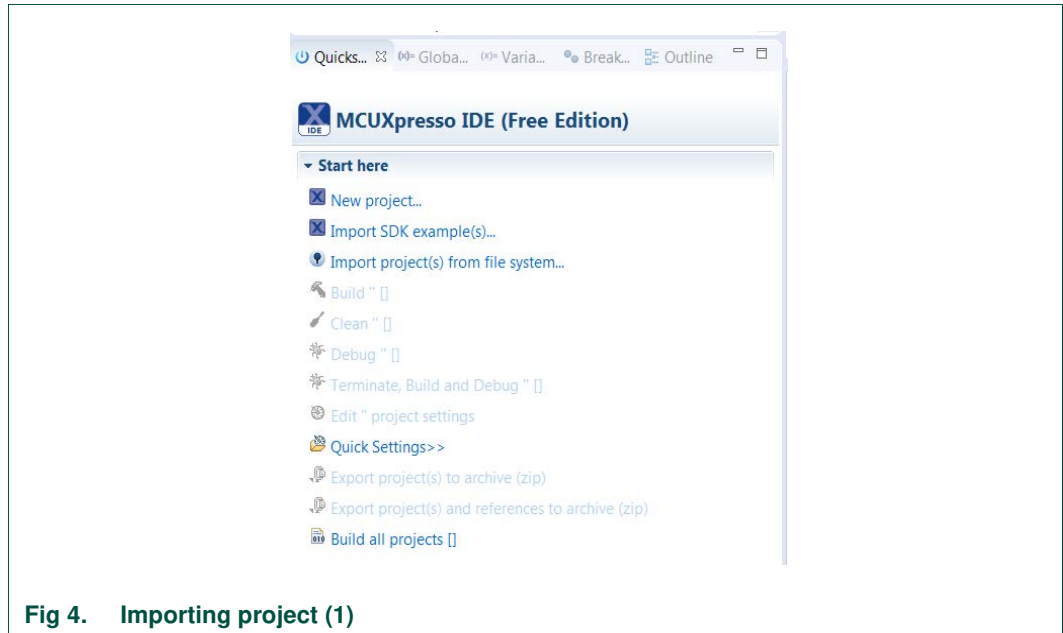- The software package is ready for use.

Fig 4.   Importing project (1)

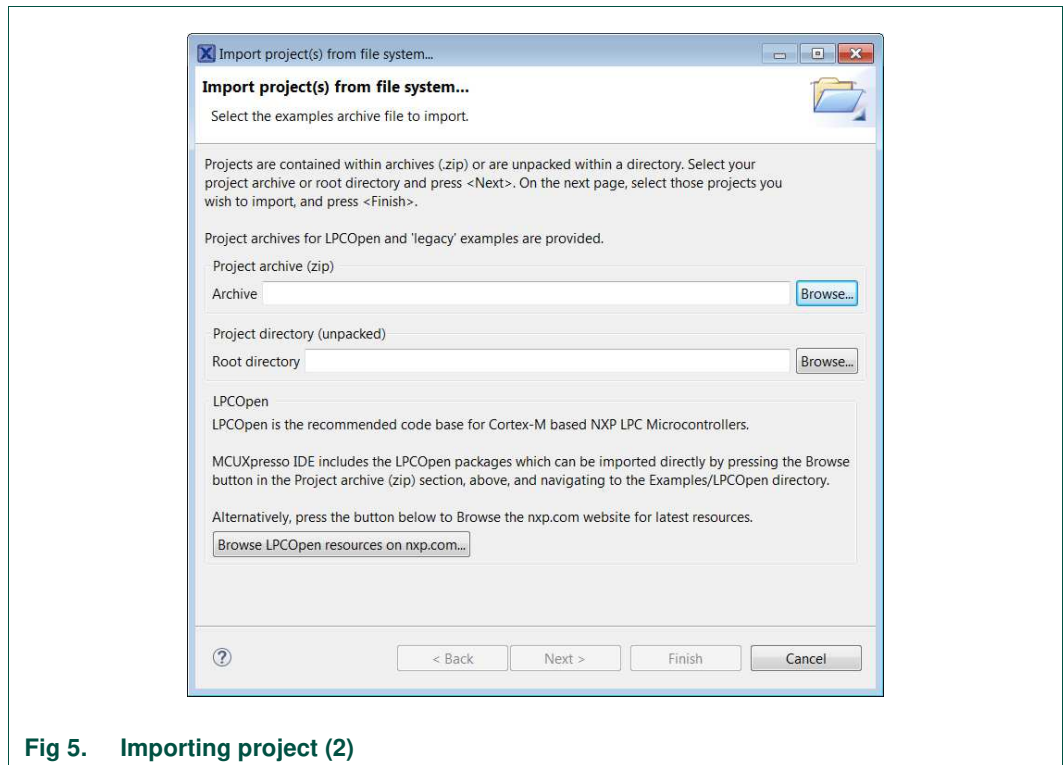In the Quickstart panel on the left-hand side, choose "Import projects(s)".
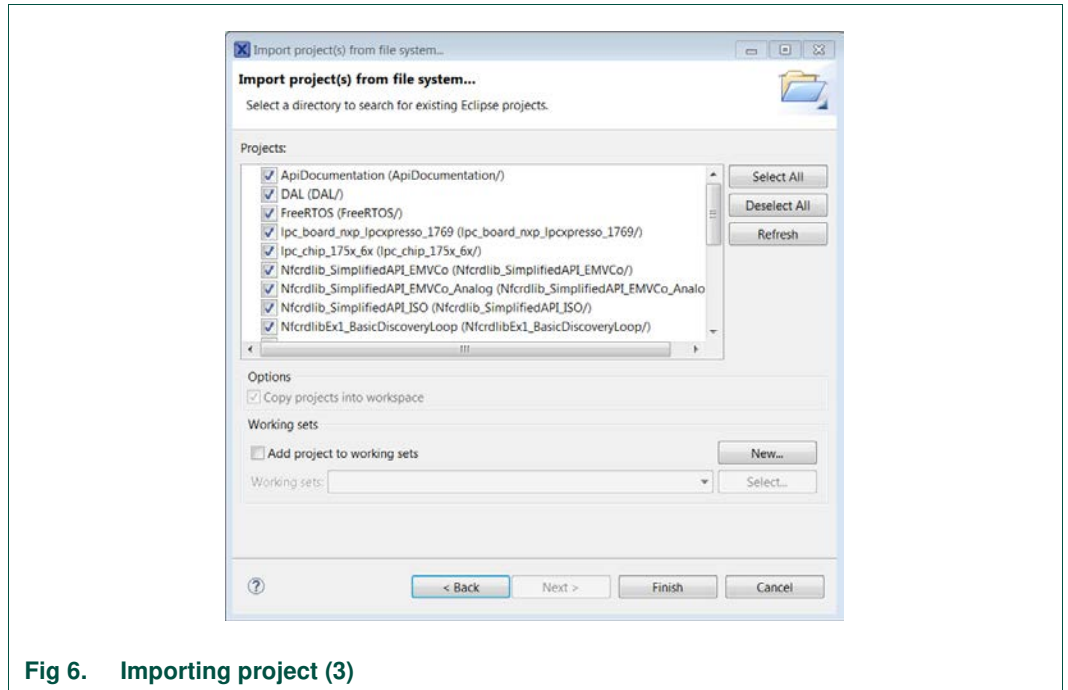

Fig 5.   Importing project (2)

Browse the desired package and click "Next".

Fig 6. Importing project (3)

For a working demo project, you need to import at least four sub projects. One example project, the NFC Reader Library, FreeRTOS, one chip library and one board library.

When the import process has finished one can start browsing the code.

2.4 Building projects

Building projects in a workspace is a simple case of using the Quickstart Panel - 'Build all projects'. Alternatively, a single project can be selected in the "Project Explorer View" and built separately. Note that building a single project may also trigger a build of any associated library projects.
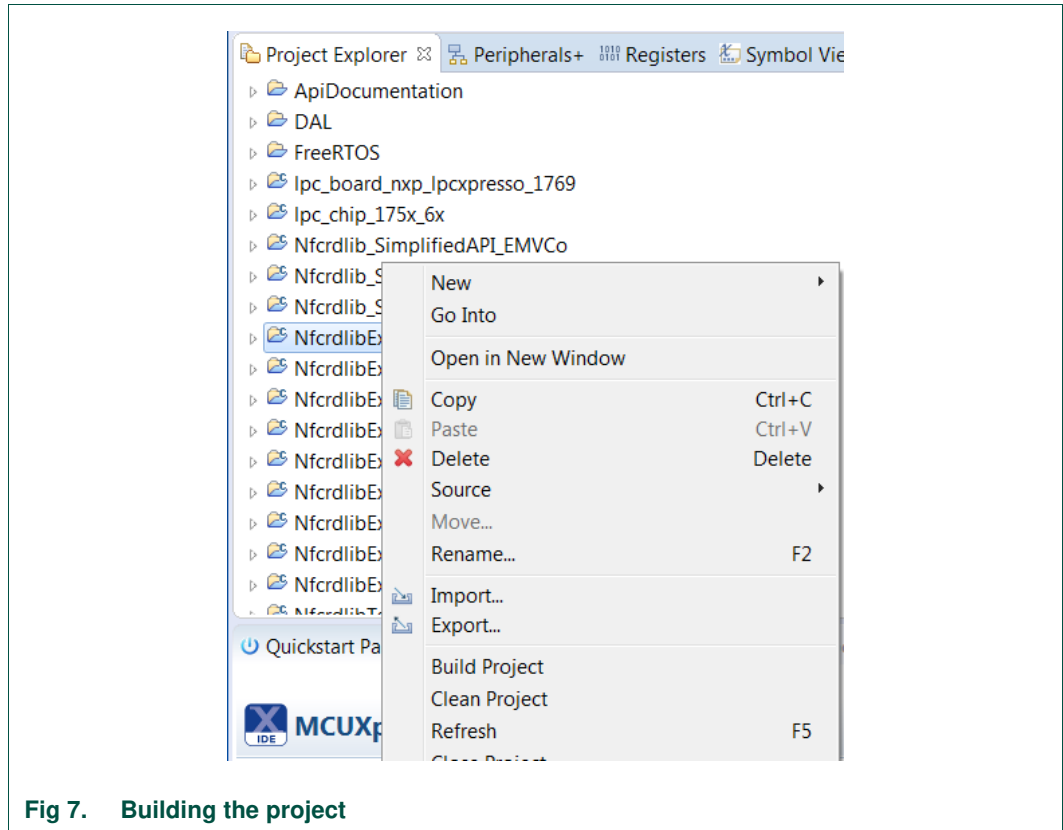
The project can be built as shown in the Fig 7.

UM10954

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

8 of 32

Fig 7.   Building the project

As a part of the build output, the binary for the "User Flash" file is created. This binary file can be later also used to update LPC1769 Flash memory via USB mass storage interface.
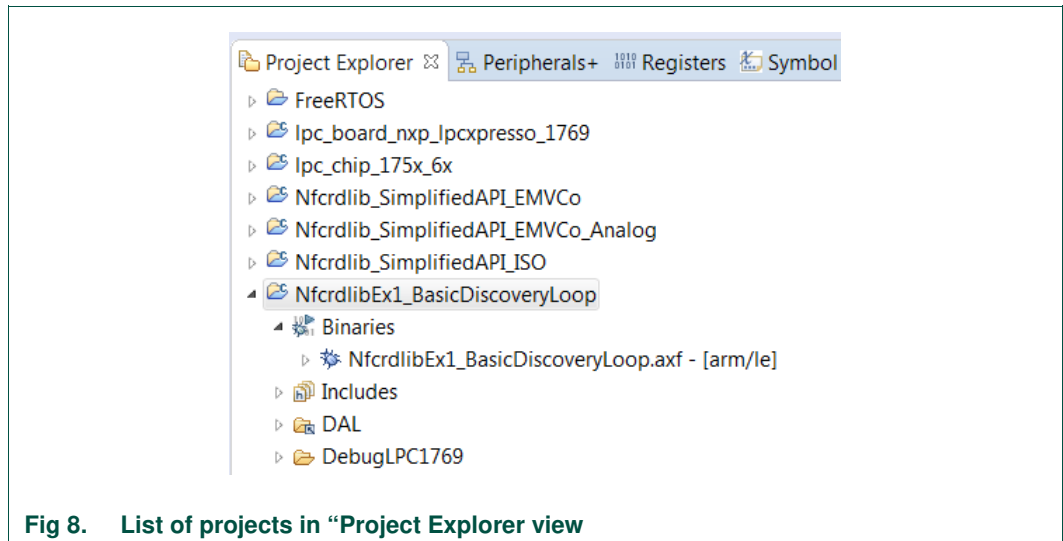

Fig 8.   List of projects in "Project Explorer view

The project settings, compiler and link flags can be changed in the project properties dialog. To open the project properties dialog, select appropriate project in the "Project Explorer View" and click "Edit 'selected-project' project settings".

2.5 Running and debugging a project

This description shows how to run the "*NfcrdlibEx1_basicDiscoveryLoop*" example application for the PN5180 evaluation development board. The same basic principles will apply for all other examples. In cases where example will need additional configuration this will be detailed described in the example description.

PN5180 evaluation board should be connected to the computer via LPC-Link 2, as shown in Fig 9.

Fig 9.   PNEV5180B with LPC-Link 2
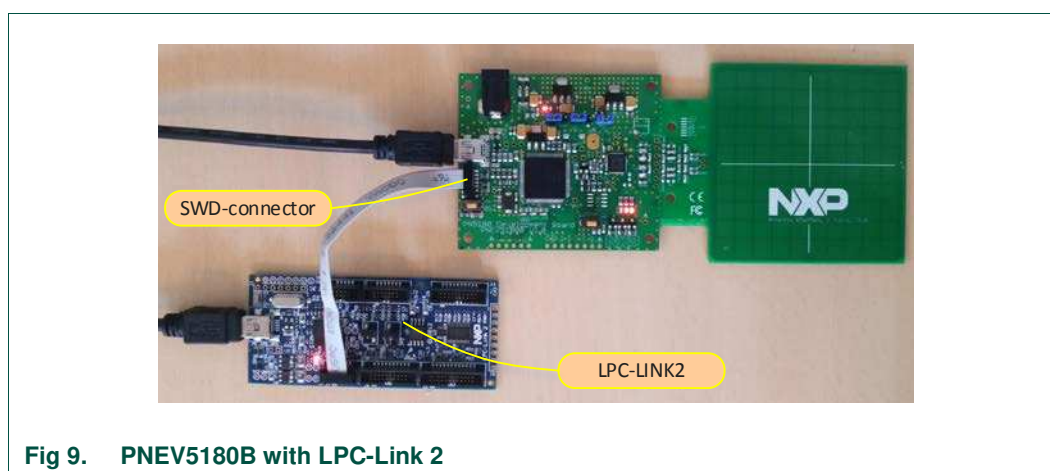
When debug is started, the program is automatically downloaded to the target and it´s programmed to the LPC1769 flash memory; a default breakpoint is set on the first instruction in main (), the application is started (by simulating a processor reset), and code is executed until the default breakpoint is hit.

To start debugging your application on the PN5180, simply highlight the project in the Project Explorer and then in the Quickstart Panel click Debug, as shown in Fig 10. The MCUXpresso IDE will first build application, flash application binary and then will start with debugging.

Before running the project, please ensure that the correct microcontroller and the correct build configurations are chosen. Information about how to do this can be found in the Fig 23 and chapter 6.3.
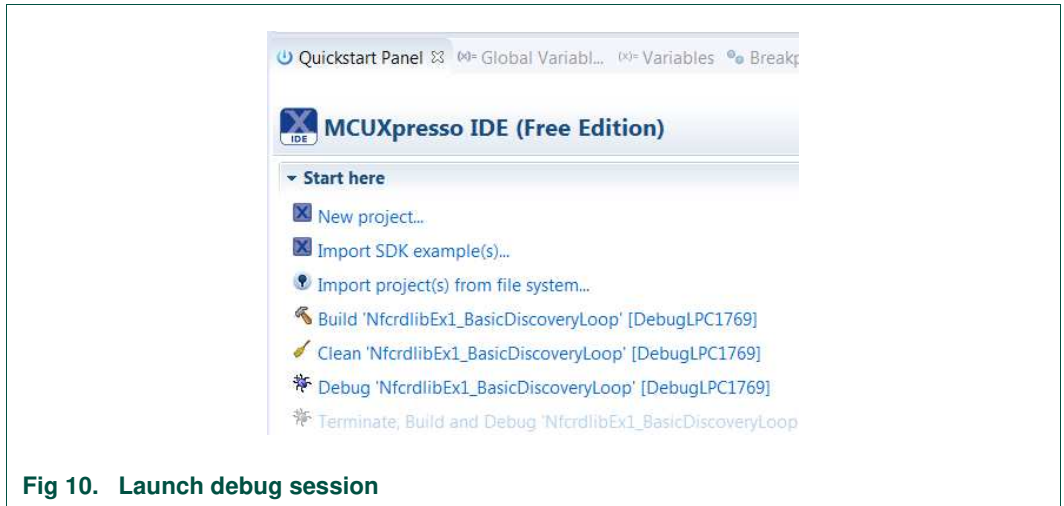
UM10954

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

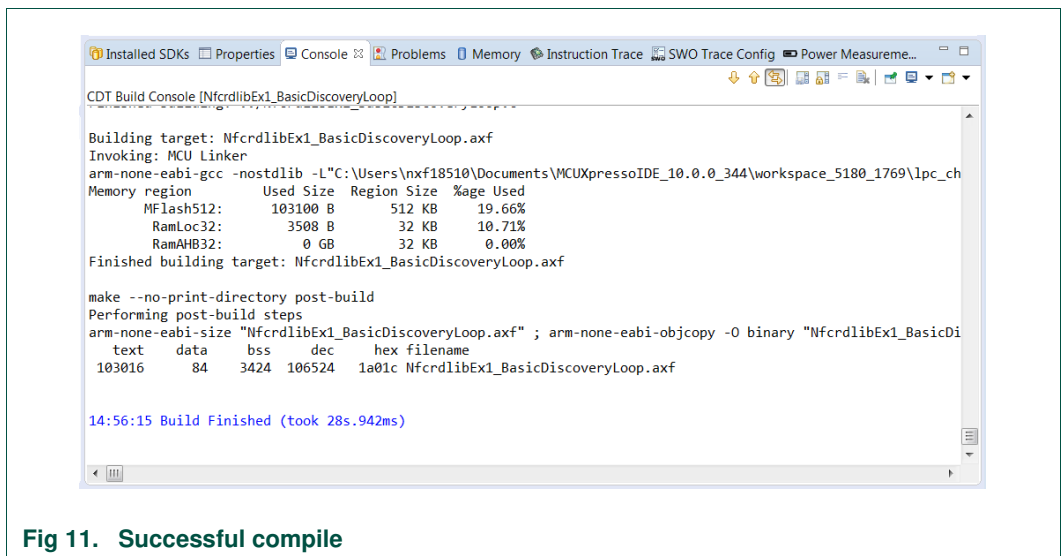**10 of 32**

Fig 10. Launch debug session

Fig 11. Successful compile

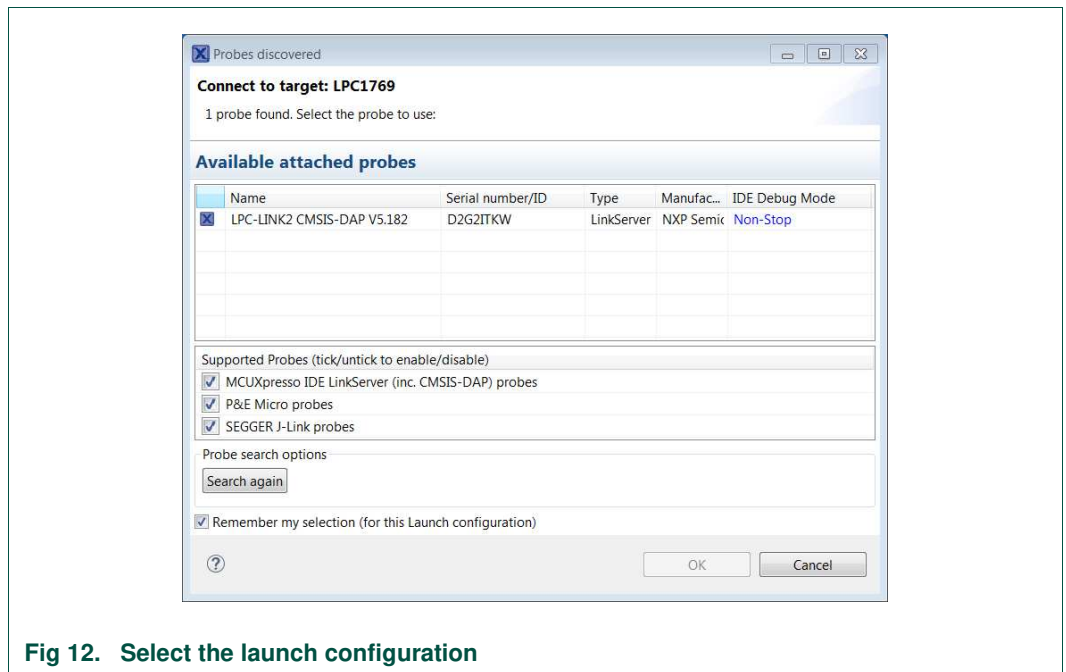Select "LPC-Link 2" as a debug probe.

Fig 12.   Select the launch configuration

After successful software upload, the execution of the project starts immediately, but might halt at the initial breakpoint. To resume execution, click the resume button.
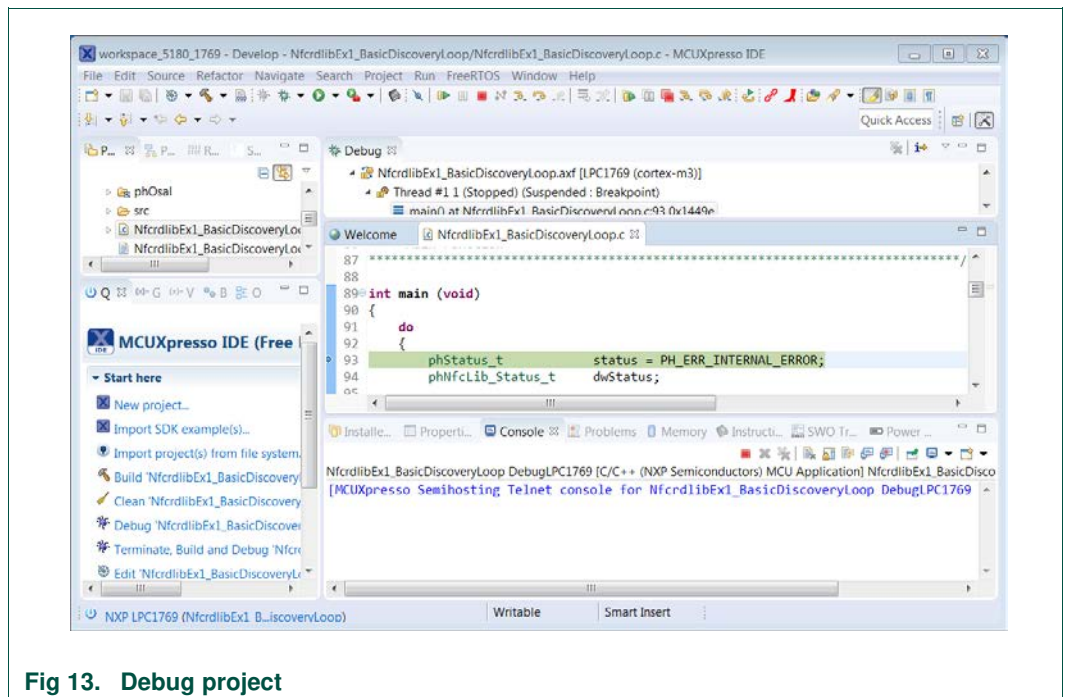

Fig 13.   Debug project

In the console window application debug outputs of the execution can be seen.


```
🔘 Installed SDKs  ⬜ Properties  🖥 Console ⊠  📋 Problems  🗐 Memory  ◈ Instruction Trace  📟 SWO Trace Config  ⬚ Power Measurement Too
NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 [C/C++ (NXP Semiconductors) MCU Application] NfcrdlibEx1_BasicDiscoveryLoop.axf
[MCUXpresso Semihosting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 started on port 3331]


BasicDiscoveryLoop Example:


Card detected and activated successfully...
Technology  : Type A
Card: 1
UID : 04 63 56 6A F1 3B 80
SAK : 0x08
Type: Type 2 Tag
```
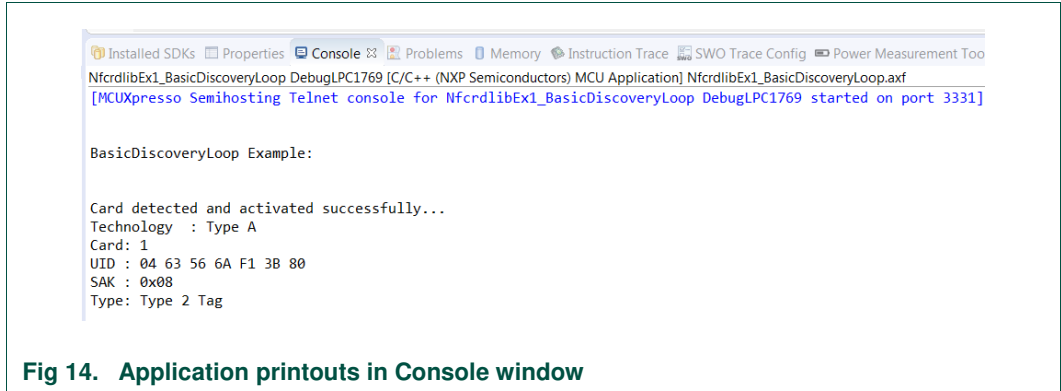
Fig 14.   Application printouts in Console window

After the execution has reached the end of the main function please click the Terminate button to stop the execution. Otherwise rerun of the project will be possible.

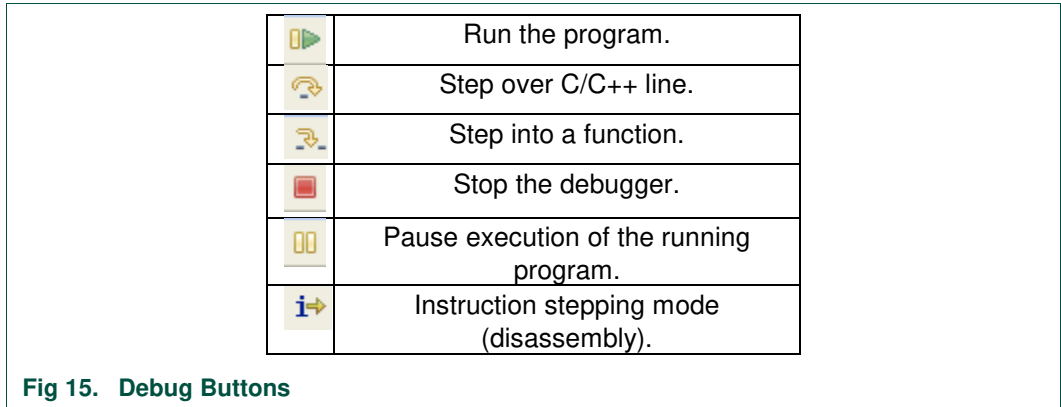Buttons in the debug toolbar provide next functionalities:

▷	Run the program.
↷	Step over C/C++ line.
⤵	Step into a function.
◼	Stop the debugger.
⏸	Pause execution of the running program.
i→	Instruction stepping mode (disassembly).

Fig 15.   Debug Buttons

3.   Managing the PN5180 SW projects with Linux and Kinetis platform

Detailed description and guideline, how to import and manage NFC NXP Reader Library projects for Linux and Kinetis platform, check:

-   AN11802 - NFC Reader Library for Linux Installation Guidelines

-   AN11908 - NFC Reader Library for FRDM-K82F Installation Guidelines

UM10954

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

© NXP B.V. 2018. All rights reserved.

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

13 of 32

4. Associated projects

All example projects are available for download at the PN5180 product page in the documents section and are being distributed in one single file.

All projects are packaged into a single installer file. After downloading the zip file, extract it and run the installer. The installer makes a copy of all documents and SW on the hard disk.

By default, the projects are preconfigured to be run on the PNEV5180B development board. This is defined by preprocessor directive PHDRIVER_LPC1769PN5180_BOARD (properties-> settings->preprocessor) and by macro in "*../intfs/ph_NxpBuild_App.h*".

```
//#define NXPBUILD__PHHAL_HW_RC663
#define NXPBUILD__PHHAL_HW_PN5180
```

Running the projects with, or without FreeRTOS

All projects described in the following sub chapters can be configured to run with or without FreeRTOS operating system. To enable/disable FreeRTOS support, define preprocessor directive (properties-> settings->preprocessor) PH_OSAL_FREERTOS or PH_OSAL_NULLOS.

4.1 Example 1 – Basic Discovery Loop

The Discovery Loop can be seen as the entry point when starting to communicate with an NFC tag or device. It scans the close environment for tags and devices of different technologies.

Example is implemented to work in POLL and LISTEN mode of the discovery loop. Information (like UID, SAK, and product type of MIFARE IC) of the detected tags are printed out and it also prints information when it gets activated as a target by an external initiator/reader. Whenever multiple technologies are detected, example select first detected technology and resolve it.

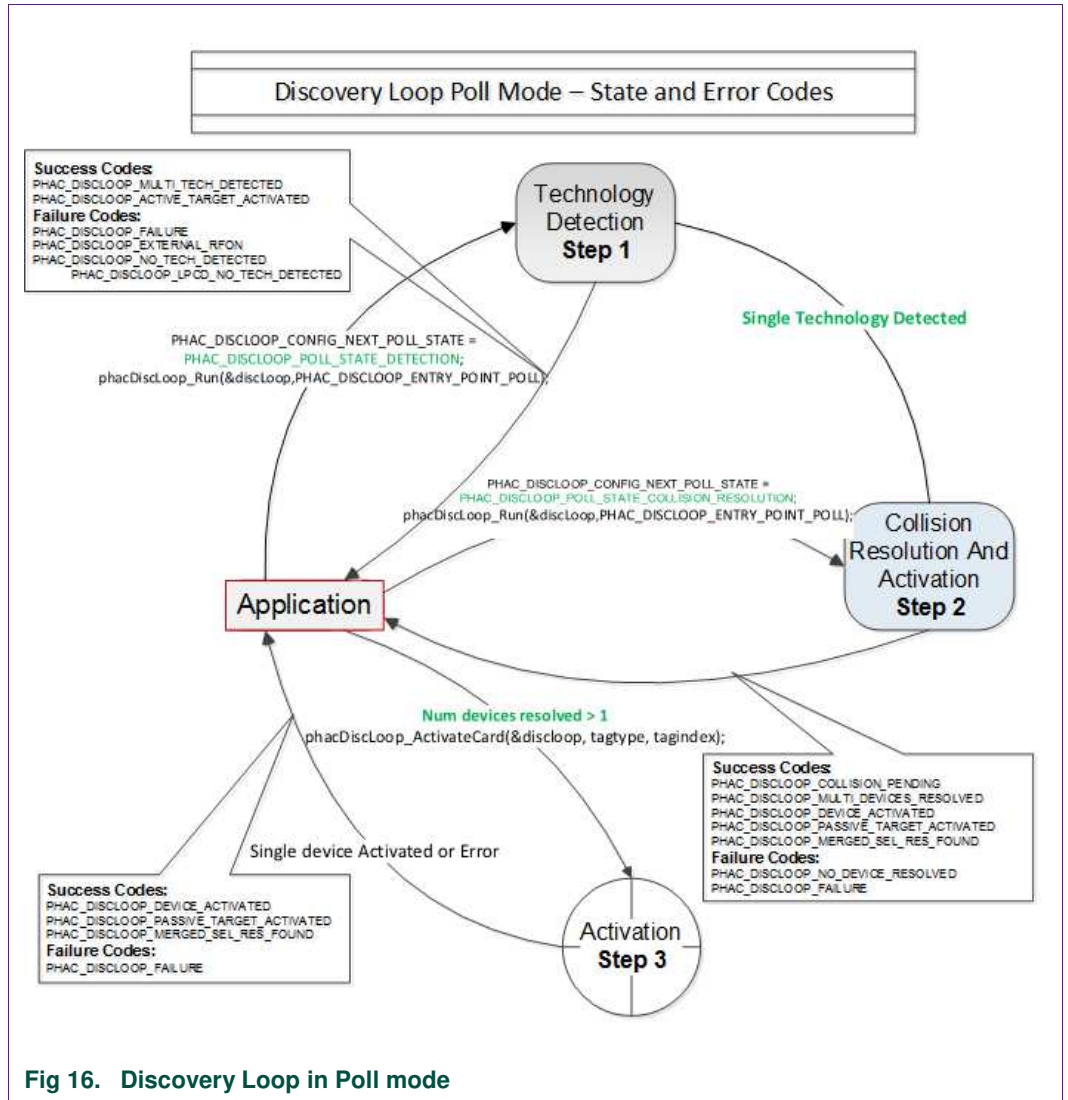In passive poll mode, Low Power Card Detection (LPCD) is enabled.

UM10954

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

© NXP B.V. 2018. All rights reserved.

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

14 of 32

Fig 16.   Discovery Loop in Poll mode

The core function of this example is *"BasicDiscoveryLoop_Demo()",* where initialization of the NFC Reader library and polling for NFC technologies is implemented. After each polling loop, application is checking polling result and printout information about the detected tags or devices.

This example is using default DiscoveryLoop configuration, which enables all supported technologies and it is limited to one device for each technology.

Table 3.   Supported technologies

ISO14443P3A	ISO15693- SLI	FeliCa	TYPEF_TARGET_PASSIVE
ISO14443P4A	ISO18000P3M3	TYPEA_TARGET_PASSIVE	TYPEF_TARGET_ACTIVE
ISO18092MPI	ISO14443P3B	TYPEA_TARGET_ACTIVE	

UM10954

© NXP B.V. 2018. All rights reserved.

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

15 of 32

4.2 Example 2 – Advanced Discovery Loop

Additionally to Example 1 the Advanced Discovery Loop example explains the different configuration options of the Discovery Loop and configure DiscoveryLoop with default values based on the interested profile, NFC or EMVCo.

The configuration of the "DiscoveryLoop" is implemented in "*LoadProfile()*" function.

4.3 Example 3 – NFC Forum

Explains how to configure the NFC Reader Library for different P2P modes such as Active Mode, Target Mode, Initiator Mode and SNEP Client/Server.

In SNEP Server mode the example waits for a connection from a SNEP Client. When the connection between client and server is establish, client send a data and server read it. The application displays read data in the console window of the LPCXpresso IDE.

In SNEP Client mode, the application tries to connect to a SNEP Server. Once the connection is established, it transmits an NDEF message to the server.

4.4 Example 4 – MIFARE Classic card communication

This example demonstrates how to configure "DiscoveryLoop" to poll for only one technology and how to resolve detected card, in this example MIFARE Classic is used.

Once MIFARE Classic card is activated, application printout information like UID, ATQA and SAK and perform the authentication with MIFARE Classic card default key. After successful authentication, basic read/write operations are implemented.

This example is good start in case of working with only one card or to see how to manage MIFARE Classic cards.

4.5 Example 5 - ISO15693

Similar to the previous example, this one is also using only one technology, in that case ISO15693. "*DiscoveryLoop*" is configured to resolve only one device and in the example it is shown how to change Tx Guard Time for T5T cards, this is implemented in "phApp_Init()" function.

Once ICODE SLI is resolved and activated, application printout card information like type of the card and UID, and it will read and write from/to the memory block.

This example is good start in case of working with only one card or to see how to manage ISO15693 type of the cards.

For a much more extensive example, demonstrating the use of ISO/IEC 15693 and ISE/IEC 18000-3 Mode 3 tags (ICODE SLI and ICODE ILT). In order to assure ICODE SLI and ILT detection please check HAL digital delay define settings as described in chapter 4.

4.6 Example 7 – EMVCo Polling

The EMVCo Polling example it is demonstrated how to configure NFC Reader Library as specified by EMVCo specifications and starts polling for EMVCo cards.

Once an EMVCo compatible card is resolved and activated, it demonstrates the exchange of APDU commands. This example shall help the developers getting started more quickly when working with EMVCo cards.

4.7 Example 8 – HCE T4T

Example 8 implements a Type 4 Tag card emulation according to NFC Forum Type 4 Tag specification. The example supports all specified commands such as *Select*, *ReadBinary*, *UpdateBinary*.

With this example our reader is in card emulation mode (HCE) and it support reading and writing data. Default data is configured as an NDEF message as a url www.nxp.com.

The maximum NDEF length the reader can write is limited by NDEF file size used in example (default configured as 1024 bytes).

4.8 Example 9 – NTAG-I2C

The NTAG-I2C example demonstrates the use of special features which are supported by NTAG-I2C. By using POLL mode of the discovery loop, example detect the NTag I2C cards and displays detected tag information like UID, ATQA, SAK, Version info and perform "*Page Read*" and "*PageWrite*" commands.

For more details about the NTAG-I2C and its functionalities please consult the related product page [2]

4.9 Example 10 – MIFARE DESFire card communication

The MIFARE DESFire example demonstrates how to use MIFARE DESFire EV1 cards.

Once MIFARE DESFire card is resolved and activated, it displays MIFARE DESFire applications created by this example previously and it displays 32bit signed integer which is incremented after each successful detection of tag.

In case no application is present on the tag, new application will be created with two new files to hold NXPNFCRDLIB version used to create this application and another file to hold 32bit signed integer.

Note: This example including the required modules of the NFC Reader Library is only available via NXP Docstore.

4.10 Example 11 – ISO10373 PCD

This example is used to perform ISO 10373-6 PCD compliance validation. This example has to be executed in the DUT which has an ISO 14443 based PCD implementation. The ISO 10373-6 test methods verifies the compliance to the ISO 14443 protocols. An

external tool like Micropross MP300 implements the test methods for the ISO 10373-6 and is used as the counterpart for this testing.

4.11 Simplified API EMVCo

This application will configure Reader Library as per EMVCo specification and start EMVCo polling. This loop back application will send SELECT_PPSE command and is used to test EMVCo.3.1a(L1) digital compliance. Simplified approach, after library initialization, is using only three commands:

- *phNfcLib_Activate()*

- *phNfcLib_Transmit()*

- *phNfcLib_Receive()*

4.12 Simplified API EMVCo Analog

This example provides command line interface so user can choose one of three operation modes as below.

- *EMVCo LoopBack Application*

- *Trans send Type A application*

- *Trans send Type B application*

Above Application modes are used to perform EMVCo2.6(L1) Analog compliance validation.

4.13 Simplified API ISO

This example is a reference application to demonstrate the usage of Simplified API with ISO profile. Application contains example of Type A Layer 4, Type B Layer 4, ISO/IEC15693 and ISO/IEC180003m3 and MIFARE DESFire card, MIFARE Ultralight card and MIFARE Classic card communication.

Example demonstrates how to use simplified API, which require, after successful library initialization, only three commands:

- *phNfcLib_Activate()*

- *phNfcLib_Transmit()*

- *phNfcLib_Receive()*

UM10954

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

18 of 32

5. Flashing Firmware on the LPC1769

Flashing VCOM firmware is the procedure needed to prepare PN5180 evaluation board to be used with the Cockpit tool. The Cockpit tool is software design tool for prototyping NXP NFC card reader applications.
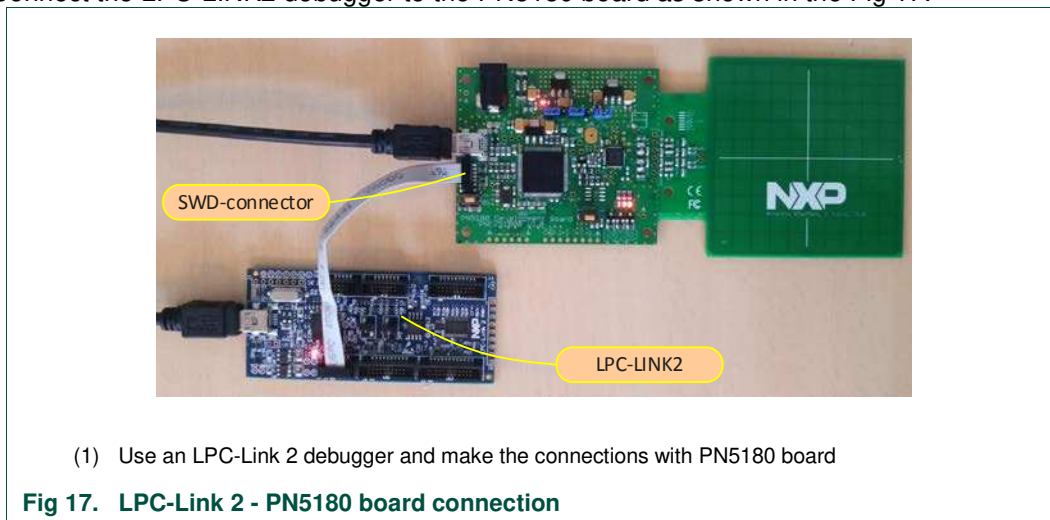
VCOM firmware which needs to be flashed on the LPC1769 is provided with the installer package of the NXP NFC Cockpit application and it can be found in "\NxpNfcCockpit_v3.10.0.0\firmware\Secondary_PN5180" folder.

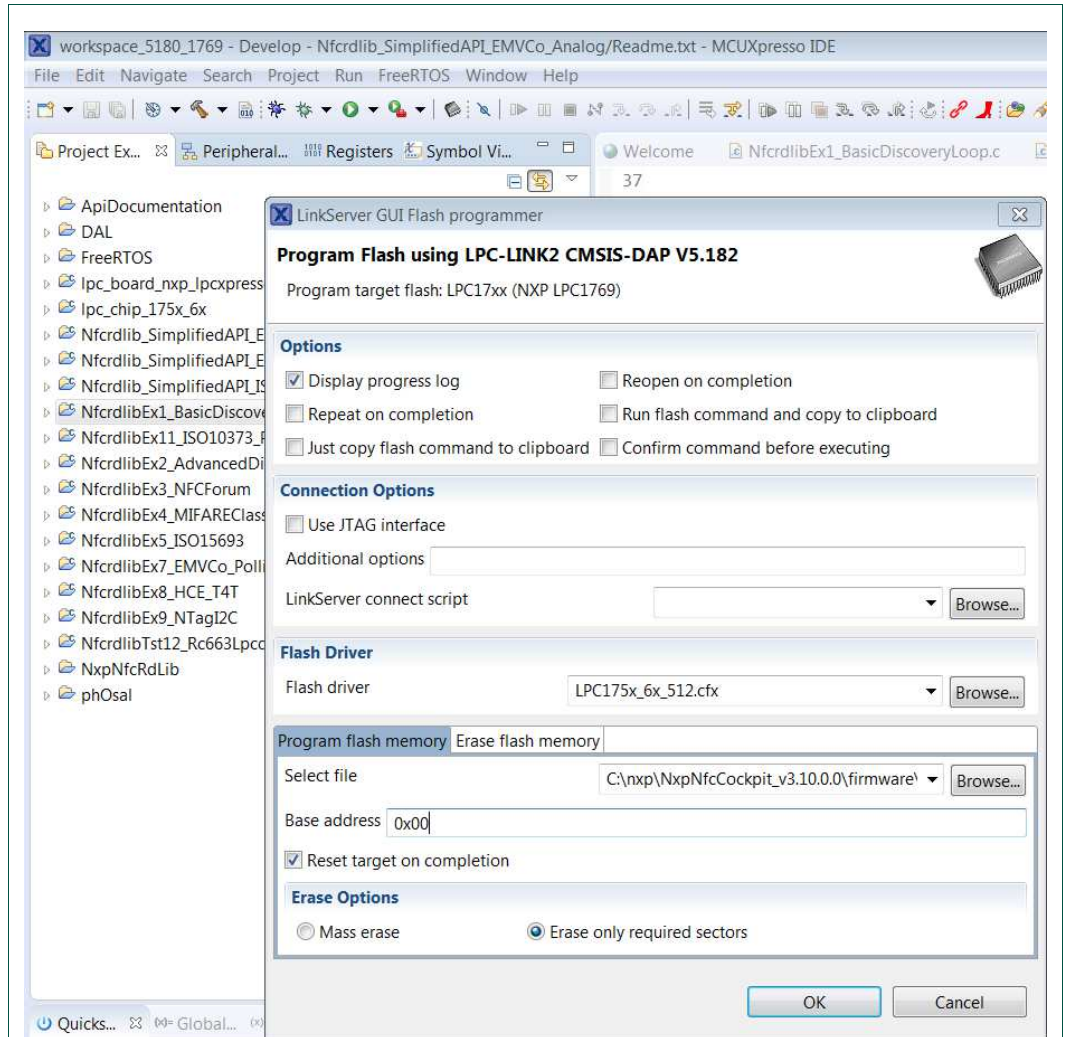This folder contains several binaries which can be used as VCOM firmware application:

• BootLoader_And_Nfcrdlib_SimplifiedAPI_EMVCo_Secondary.bin

• BootLoader_And_phRfOnOff_Secondary.bin

• BootLoader_And_phUcBal_Secondary.bin

Steps required to flash VCOM application:
Connect the LPC-LINK2 debugger to the PN5180 board as shown in the Fig 17.

(1) Use an LPC-Link 2 debugger and make the connections with PN5180 board

Fig 17. LPC-Link 2 - PN5180 board connection

UM10954

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

© NXP B.V. 2018. All rights reserved.

User manual
**COMPANY PUBLIC**

1.6 — 7 May 2018
345816

19 of 32

Use the MCUXpresso Program Flash utility flash to flash bootloader binary to the MCU.

(2) Create (or Open an existing MCUXpresso project) and set the MCU type to LPC1769

(3) Chose "Program flash" from menu bar

(4) Select bootloader binary, e.g.
'NxpNfcCockpit_v3.10.0.0\firmware\Secondary_PN5180\BootLoader_And_Nfcrdlib_SimplifiedAPI
_EMVCo_Secondary.bin'

(5) Set base address to 0x00

(6) Press "OK" button and flash the bootloader

Fig 18. Flashing Bootloader.bin

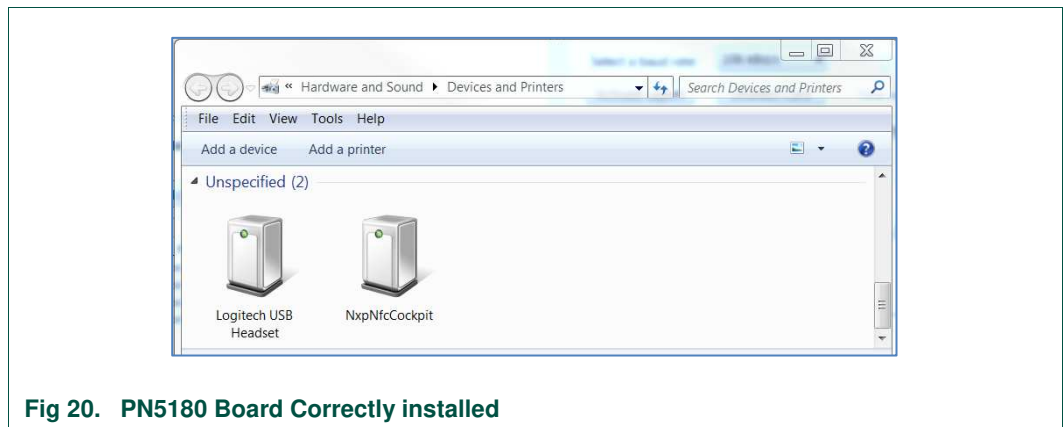Flash the binary on the PNEV5180B and ensure that the process succeeds.
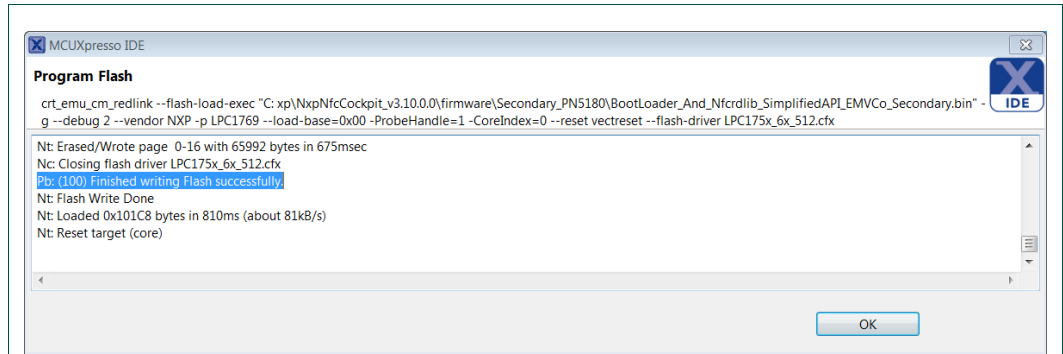
(7)   Inspect program flash output

Fig 19.   Flash report

Disconnect the USB cable, remove LPC-Link 2 connection and reset the board.

The board appears as a VCOM device:

Fig 20.   PN5180 Board Correctly installed

The PN5180 based evaluation board is ready now to be used with the NXP NFC Cockpit tool.

6. Supplementary Notes

6.1 General Software Architecture

The software of the reference reader is based on the NFC Reader Library Fig 21. It intends to be simple, modular, easily readable and quickly portable by all the customers. This philosophy is reflected in its architecture which is divided into 4 layers:

- BAL (Bus Abstraction Layer),
- HAL (Hardware Abstraction Layer)
- PAL (Protocol Abstraction Layer)
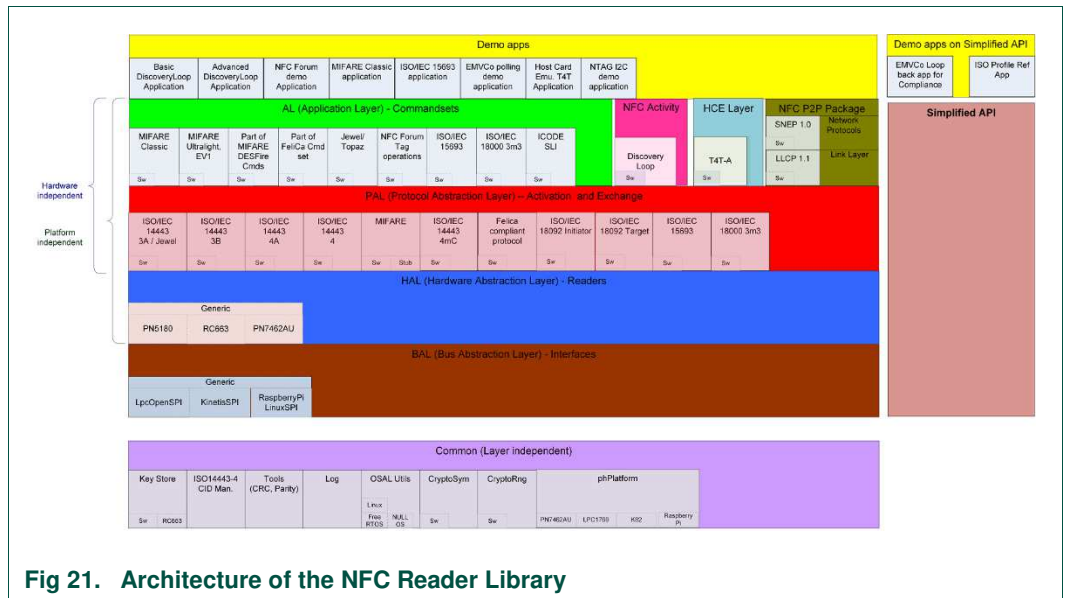- AL (Abstraction Layer)

Fig 21.   Architecture of the NFC Reader Library

6.1.1 Bus abstraction layer

This layer offers functions to abstract the hardware parts of the LPC1XXX microcontroller.

These functions use the specific libraries available for the LPC1XXX family microcontroller. Based on these stacks, the communication routines for the relevant physical media I2C/SPI can be easily designed. These drivers are specific for the LPC1XXX family and therefore cannot be ported to other microcontrollers.

6.1.2 Hardware abstraction layer

This layer offers functions to abstract the hardware parts of the transceiver CLRC663.

6.1.3 Protocol abstraction layer

Every PAL function is a low-level function realizing a single functionality. It is encapsulated in a module which is independent from the others. The user can easily design his application by doing a drag-and-drop of the relevant module.

UM10954

© NXP B.V. 2018. All rights reserved.

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

22 of 32

The following PAL modules are available in this software package:

- ISO/IEC 14443-3A,
- ISO/IEC 14443-3B,
- ISO/IEC 14443-4A/B,
- MIFARE products
- FeliCa
- NFC Initiator
- NFC Target

6.1.4 Application layer

Lying on the previous software layers, the application layer is on top of the reader software package. It combines elements of the previous three parts into high level functionalities.

6.2 Build configuration

All the projects mentioned in Chapter 0 are available in debug configuration. Additionally, the Polling project comprises the release configuration.

- Debug configuration

  This configuration is mainly used when the target board is attached to the PC with the JTAG debugger. It allows the display of debug messages in the console window, which is useful in the early stage of the project.

- Release configuration

  Once the project is debugged and mature, it might be interesting to use the release configuration, to use the hardware stand alone. No debug messages are displayed in the console window.

The build configuration can be selected as follows:

- Click on the project in the project window of the MCUXpresso IDE,
- Right click of the mouse → Select Build Configuration,
- Set active DebugLPC1769 build (or ReleaseLPC1769 build) for LPC1769.
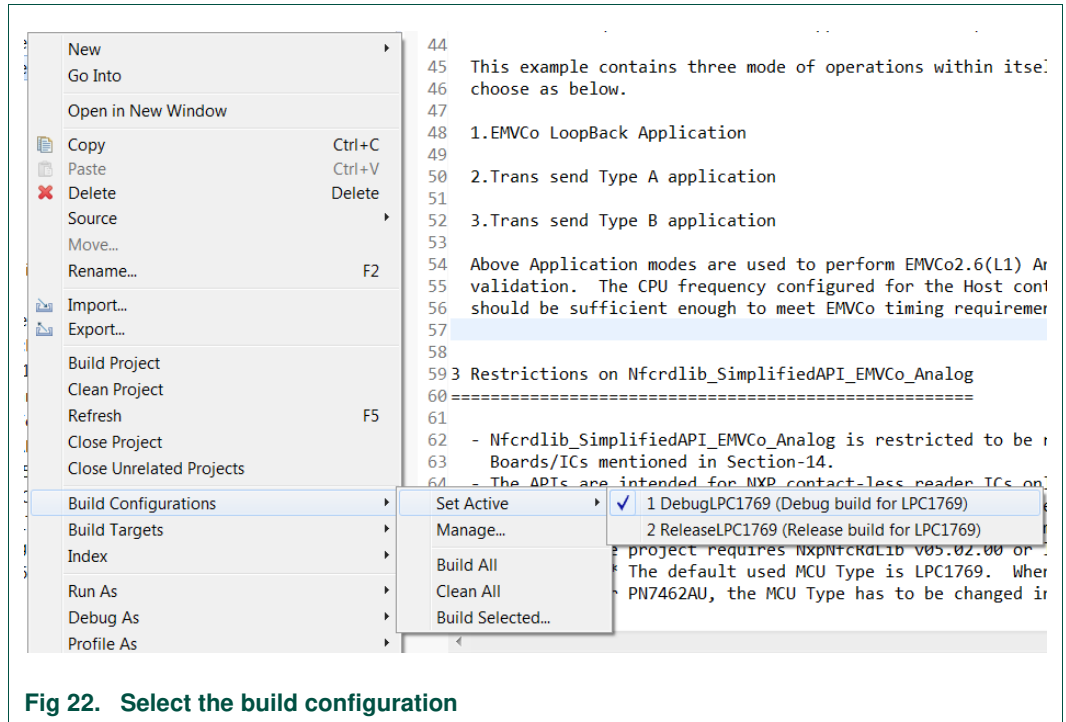
Fig 22. Select the build configuration

6.3 Setting the MCU

There are many LPC microcontrollers supported by the MCUXpresso IDE build in compiler. Before compiling a project, the correct MCU need to be set.

- Right click the project → choose properties (at the bottom)
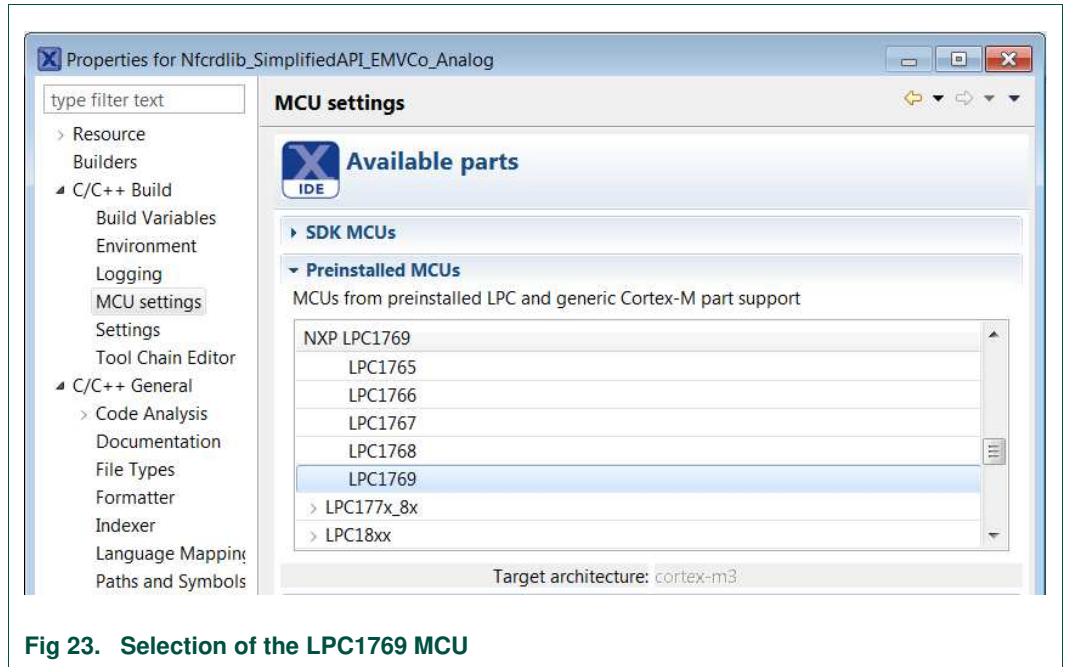- C/C++ build → MCU settings → expand desired LPC700 MCU group → choose the correct microcontroller → click OK

Fig 23.   Selection of the LPC1769 MCU

UM10954

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

© NXP B.V. 2018. All rights reserved.

User manual
COMPANY PUBLIC

1.6 — 7 May 2018
345816

25 of 32