UM10954

PN5180 SW quick start guide

Rev. 1.7 — 5 December 2022 User manual

Document information

Information	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

Revision history

Rev	Date	Description
1.7	20221130	• Updated to match latest released NFC Reader Library v07.05.00. • Section 7 "Radio Equipment Directive (RED)": added • Editorial updates
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

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 reinstall 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 based on the NFC Reader Library v07.05.00 NDA version:

Table 1. Example projects

Example projects delivered with the NFC Reader Library

Example	Description
NfcrdlilbEx1_Basic Discovery Loop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. Default configuration parameters are used.
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.
NfcrdlilbEx6_LPCD	Explains and demonstrates the usage of LPCD (Low-Power Card Detection) and its calibration.
NfcrdlibEx7_MIFAREPlus	Explains the usage of a MIFARE Plus card and pre-personal ization of it.
NfcrdlilbEx8_ HCE_T4T	Explains how to emulate an 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 Secure Files on NXP.com)
Nfcrdlib_SimplifiedAPI ISO	Explains how to use simplified API with different types of cards.

Table 2. Compliance applications

Compliance applications aim to help in certain certification scenarios.

Example	Description
Nfcrdlib_EMVCo_AnalogComplApp	Used to perform EMVCo 3.0 L1 Analog compliance validation.

Table 2. Compliance applications*...continued*

Compliance applications aim to help in certain certification scenarios.

Example	Description
Nfcrdlib_EMVCo_InteropComplApp	This example is an Interoperability loopback Application which is used to perform EMVCo IOP(L1) with add-on(TTA Bulletin No.195) compliance validation.
Nfcrdlib_EMVCo_LoopBackComplApp	This example is a loopback Application which is used to perform EMVCo3.0 L1 digital/analog compliance validation.
Nfcrdlib_ISO10373_6_PCD_ComplApp	This example is an application which is used to perform ISO 10373-6 PCD compliance validation.
Nfcrdlib_ISO10373_6_PICC_ComplApp	This application is used to perform ISO 10373-6 PICC compliance validation.

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 MCUs based on Arm Cortex-M cores. It 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 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 the PN5180 prepared software package, all components listed in Table 3 are required.

Table 3. Development environment

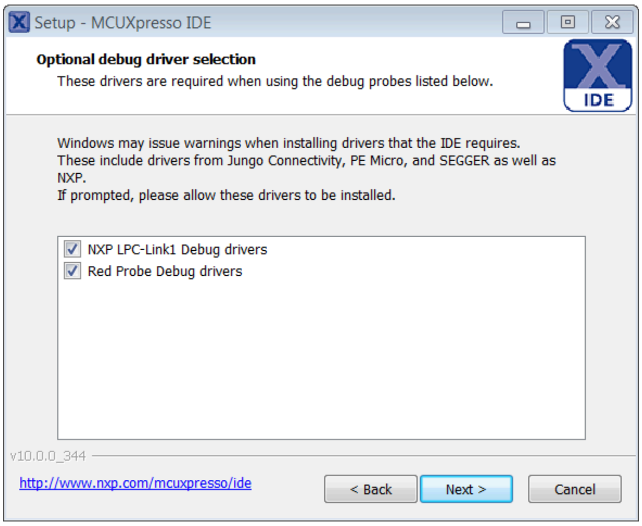
Item	Version	Description
PN5180EV5180B	1.0 or higher	PN5180 Customer evaluation board (hardware)
LPC-Link 2 or MCU-Link	1.0	Standalone debug adapter (hardware)
MCUXpresso IDE	11.6.0 or higher	Development IDE (PC software)

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.



Setup - MCUXpresso IDE

Optional debug driver selection

These drivers are required when using the debug probes listed below.

Windows may issue warnings when installing drivers that the IDE requires.
These include drivers from Jungo Connectivity, PE Micro, and SEGGER as well as NXP.
If prompted, please allow these drivers to be installed.

☑ NXP LPC-Link1 Debug drivers

☑ Red Probe Debug drivers

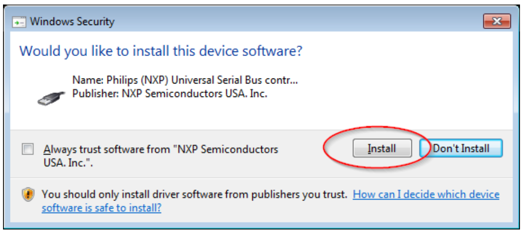
v10.0.0_344

http://www.nxp.com/mcuxpresso/ide

Figure 1. Checkbox for NXP debug drivers

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

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



Windows Security

Would you like to install this device software?

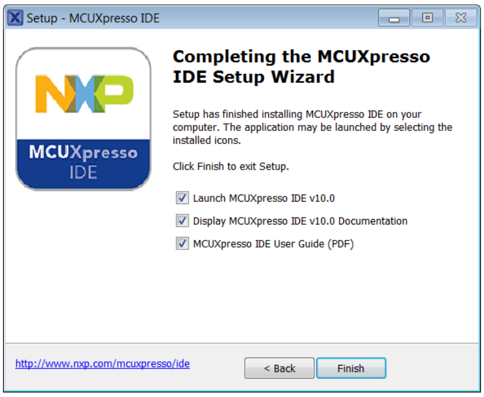
Name: Philips (NXP) Universal Serial Bus contr...
Publisher: NXP Semiconductors USA. Inc.

□ Always trust software from "NXP Semiconductors USA. Inc.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Figure 2. Windows security dialog

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



Setup - MCUXpresso IDE

Completing the MCUXpresso IDE Setup Wizard

Setup has finished installing MCUXpresso IDE on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch MCUXpresso IDE v10.0

☑ Display MCUXpresso IDE v10.0 Documentation

☑ MCUXpresso IDE User Guide (PDF)

http://www.nxp.com/mcuxpresso/ide

Figure 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 root directory of the previously unzipped NFC Reader Library.
  - The NFC Reader Library package, provided as .zip file, has to be unpacked first. Importing as archive will not work and result in broken links.
- The software package is ready for use.

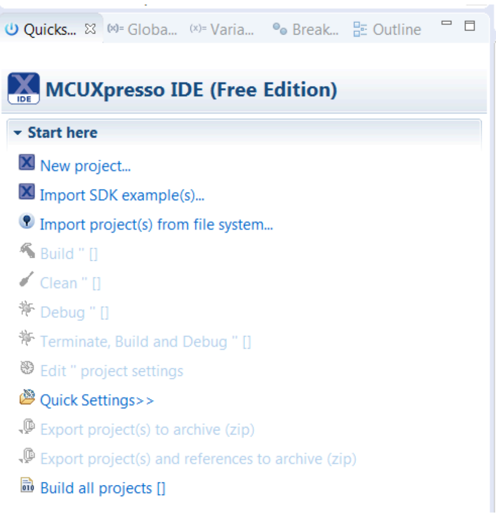
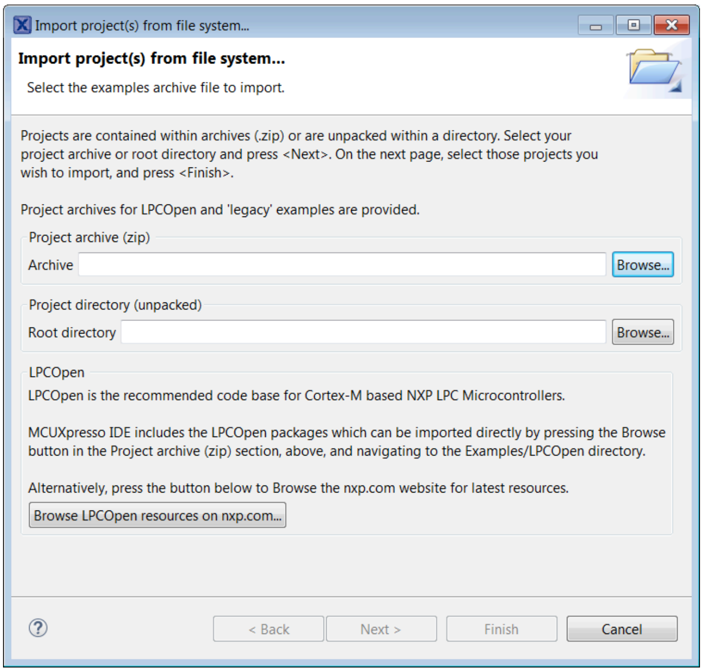


Figure 4. Importing project (1)

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



Import project(s) from file system...

Import project(s) from file system...

Select the examples archive file to import.

Projects are contained within archives (.zip) or are unpacked within a directory. Select your project archive or root directory and press <Next>. On the next page, select those projects you wish to import, and press <Finish>.

Project archives for LPCOpen and 'legacy' examples are provided.

Project archive (zip)

Archive

Project directory (unpacked)

Root directory

LPCOpen

LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.

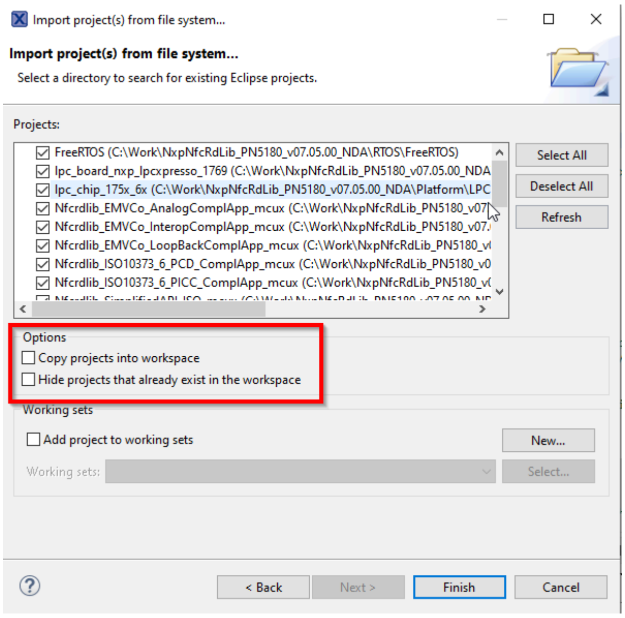
MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.

Alternatively, press the button below to Browse the nxp.com website for latest resources.

Figure 5. Importing project (2)

Browse the desired package and click “Next”.

Note: Make sure that "Copy projects into workspace" is not checked.



Import project(s) from file system...

Import project(s) from file system...

Select a directory to search for existing Eclipse projects.

Projects:

☑ FreeRTOS (C:\Work\NxpNfcRdLib_PN5180_v07.05.00_NDA\RTOS\FreeRTOS)
☑ lpc_board_nxp_lpcxpresso_1769 (C:\Work\NxpNfcRdLib_PN5180_v07.05.00_NDA
☑ lpc_chip_175x_6x (C:\Work\NxpNfcRdLib_PN5180_v07.05.00_NDA\Platform\LPC
☑ Nfcrdlib_EMVCo_AnalogComplApp_mcux (C:\Work\NxpNfcRdLib_PN5180_v07
☑ Nfcrdlib_EMVCo_InteropComplApp_mcux (C:\Work\NxpNfcRdLib_PN5180_v07.
☑ Nfcrdlib_EMVCo_LoopBackComplApp_mcux (C:\Work\NxpNfcRdLib_PN5180_v
☑ Nfcrdlib_ISO10373_6_PCD_ComplApp_mcux (C:\Work\NxpNfcRdLib_PN5180_v0
☑ Nfcrdlib_ISO10373_6_PICC_ComplApp_mcux (C:\Work\NxpNfcRdLib_PN5180_v

Options

☐ Copy projects into workspace

☐ Hide projects that already exist in the workspace

Working sets

☐ Add project to working sets

Working sets:

Figure 6. Importing project (3)

For a working demo project, you must import at least four subprojects. 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 Figure 7.

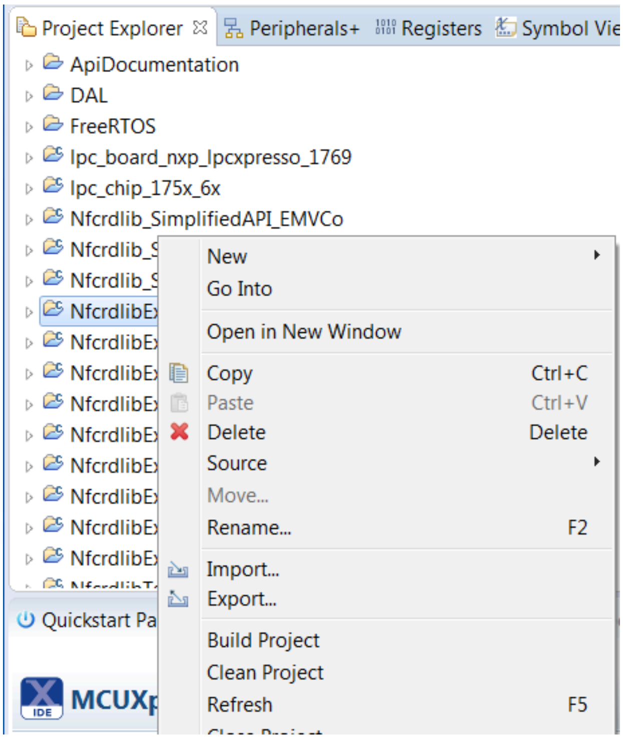


Figure 7. Building the project

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.

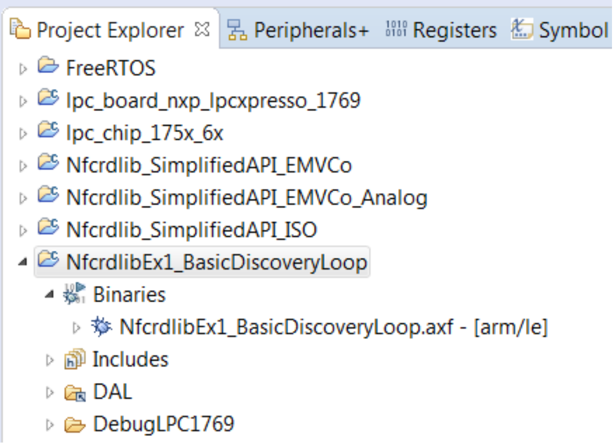

Project Explorer
Peripherals+
Registers
Symbol
FreeRTOS
lpc_board_nxp_lpcxpresso_1769
lpc_chip_175x_6x
Nfcrdlib_SimplifiedAPI_EMVCo
Nfcrdlib_SimplifiedAPI_EMVCo_Analog
Nfcrdlib_SimplifiedAPI_ISO
NfcrdlibEx1_BasicDiscoveryLoop
Binaries
NfcrdlibEx1_BasicDiscoveryLoop.axf - [arm/le]
Includes
DAL
DebugLPC1769

Figure 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 apply for all other examples. In cases where example needs 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 Figure 9.

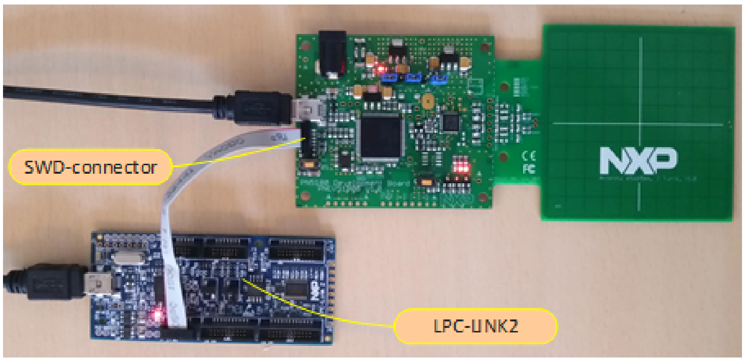


Figure 9. PNEV5180B with LPC-Link 2

When debug is started, the program is automatically downloaded to the target and it is 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 Figure 10. The MCUXpresso IDE will first build application, flash application binary and then will start with debugging.

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

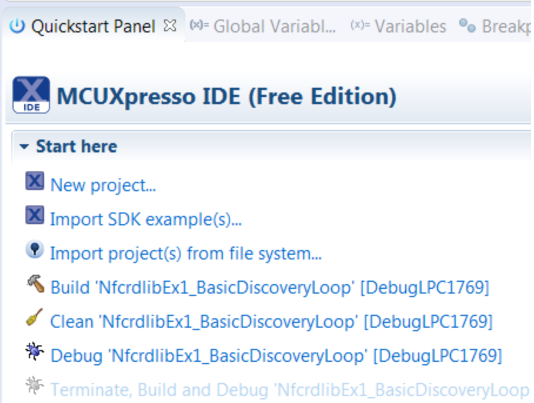
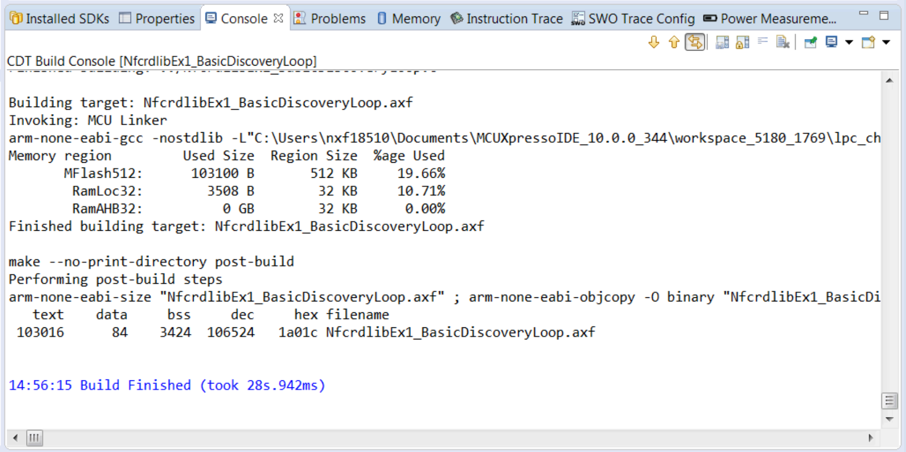

Quickstart Panel
Global Variabl...
Variables
Breakp
MCUXpresso IDE (Free Edition)
Start here
New project...
Import SDK example(s)...
Import project(s) from file system...
Build 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Clean 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Debug 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Terminate, Build and Debug 'NfcrdlibEx1_BasicDiscoveryLoop

Figure 10. Launch debug session

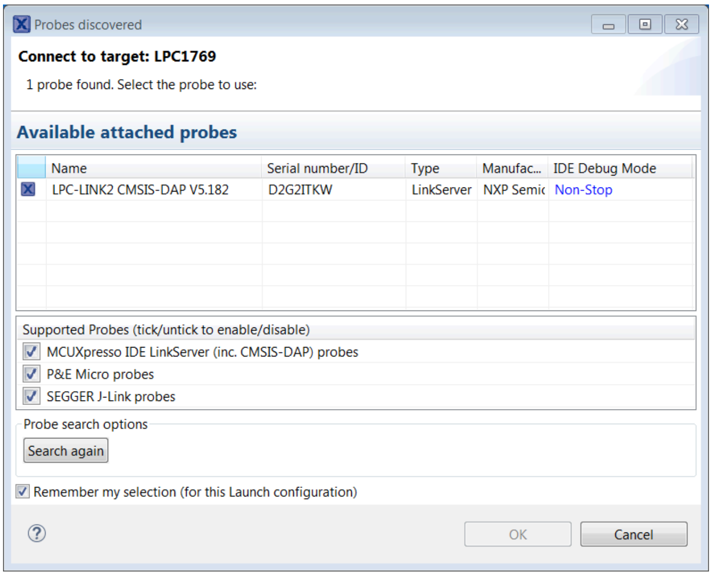


CDT Build Console [NfcrdlibEx1_BasicDiscoveryLoop]

```
Building target: NfcrdlibEx1_BasicDiscoveryLoop.axf
Invoking: MCU Linker
arm-none-eabi-gcc -nostdlib -L"C:\Users\nxf18510\Documents\MCUXpressoIDE_10.0.0_344\workspace_5180_1769\lpc_ch
Memory region         Used Size  Region Size  %age Used
       MFlash512:      103100 B       512 KB     19.66%
        RamLoc32:        3508 B        32 KB     10.71%
        RamAHB32:          0 GB        32 KB      0.00%
Finished building target: NfcrdlibEx1_BasicDiscoveryLoop.axf

make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size "NfcrdlibEx1_BasicDiscoveryLoop.axf" ; arm-none-eabi-objcopy -O binary "NfcrdlibEx1_BasicDi
   text    data     bss     dec     hex filename
 103016      84    3424  106524   1a01c NfcrdlibEx1_BasicDiscoveryLoop.axf


14:56:15 Build Finished (took 28s.942ms)
```

Figure 11. Successful compile

Select “LPC-Link 2” as a debug probe.



Probes discovered

Connect to target: LPC1769

1 probe found. Select the probe to use:

Available attached probes

	Name	Serial number/ID	Type	Manufac...	IDE Debug Mode
☒	LPC-LINK2 CMSIS-DAP V5.182	D2G2ITKW	LinkServer	NXP Semic	Non-Stop

Supported Probes (tick/untick to enable/disable)

☑ MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes

☑ P&E Micro probes

☑ SEGGER J-Link probes

Probe search options

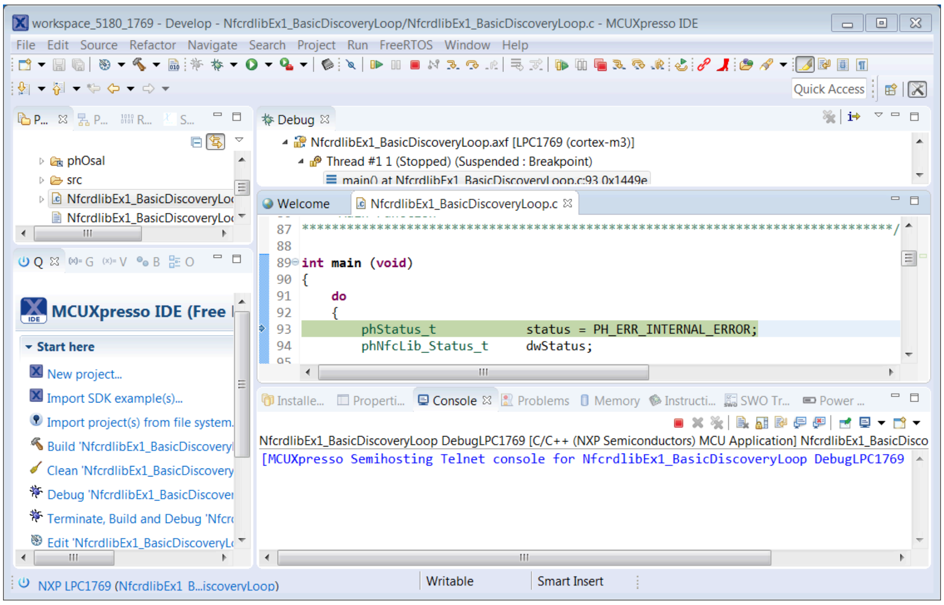
Search again

☑ Remember my selection (for this Launch configuration)

OK Cancel

Figure 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.



workspace_5180_1769 - Develop - NfcrdlibEx1_BasicDiscoveryLoop/NfcrdlibEx1_BasicDiscoveryLoop.c - MCUXpresso IDE

File Edit Source Refactor Navigate Search Project Run FreeRTOS Window Help
Quick Access
P... P... R... S...
phOsal
src
NfcrdlibEx1_BasicDiscoveryLo
NfcrdlibEx1_BasicDiscoveryLo
Q G V B O
MCUXpresso IDE (Free
Start here
New project...
Import SDK example(s)...
Import project(s) from file system...
Build 'NfcrdlibEx1_BasicDiscovery
Clean 'NfcrdlibEx1_BasicDiscovery
Debug 'NfcrdlibEx1_BasicDiscover
Terminate, Build and Debug 'Nfcr
Edit 'NfcrdlibEx1_BasicDiscoveryL
Debug
Welcome
NfcrdlibEx1_BasicDiscoveryLoop.c
Installe... Properti... Console Problems Memory Instructi... SWO Tr... Power ...
NXP LPC1769 (NfcrdlibEx1_B...iscoveryLoop)
Writable
Smart Insert

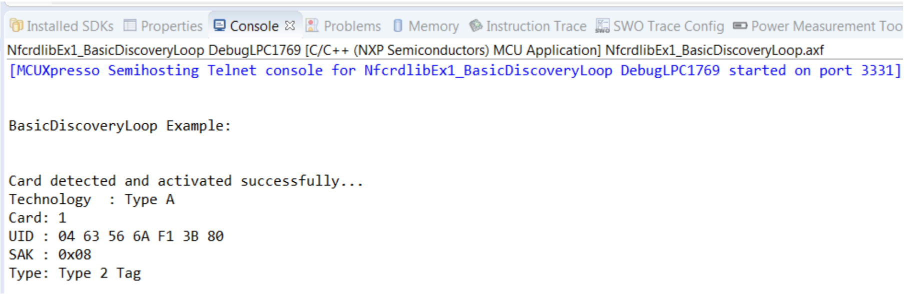
```
NfcrdlibEx1_BasicDiscoveryLoop.axf [LPC1769 (cortex-m3)]
  Thread #1 1 (Stopped) (Suspended : Breakpoint)
    main() at NfcrdlibEx1_BasicDiscoveryLoop.c:93 0x1449e
```

```
*****************************************************************************/

int main (void)
{
    do
    {
        phStatus_t              status = PH_ERR_INTERNAL_ERROR;
        phNfcLib_Status_t       dwStatus;
```

```
NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 [C/C++ (NXP Semiconductors) MCU Application] NfcrdlibEx1_BasicDisco
[MCUXpresso Semihosting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769
```

Figure 13. Debug project

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

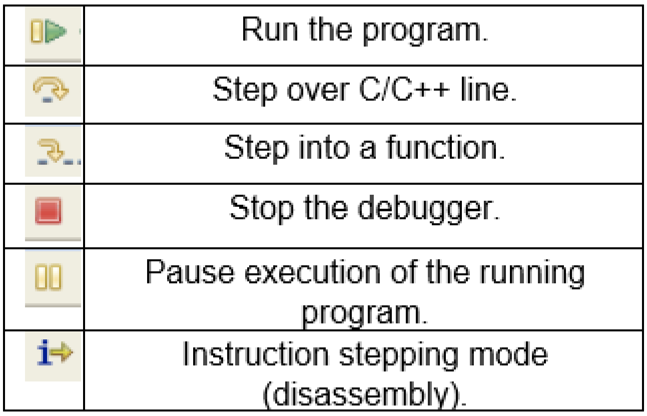


```
NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 [C/C++ (NXP Semiconductors) MCU Application] NfcrdlibEx1_BasicDiscoveryLoop.axf
[MCUXpresso Semihosting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 started on port 3331]


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
```

Figure 14. Application printouts in console window

After the execution has reached the end of the main function, 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).

Figure 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, see [3]
- AN11908 - NFC Reader Library for FRDM-K82F Installation Guidelines, see [4]

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 subchapters 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. 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.

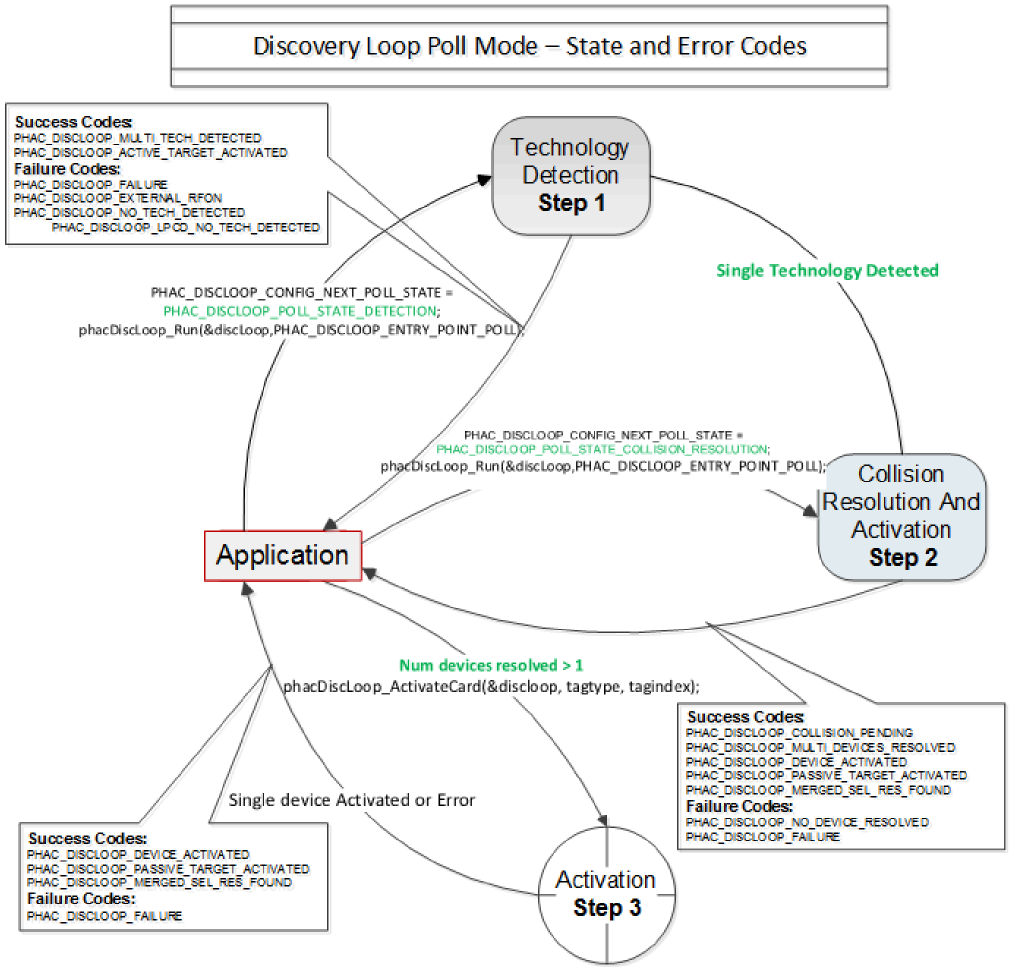

Discovery Loop Poll Mode – State and Error Codes
Success Codes:
PHAC_DISCLOOP_MULTI_TECH_DETECTED
PHAC_DISCLOOP_ACTIVE_TARGET_ACTIVATED
Failure Codes:
PHAC_DISCLOOP_FAILURE
PHAC_DISCLOOP_EXTERNAL_RFON
PHAC_DISCLOOP_NO_TECH_DETECTED
PHAC_DISCLOOP_LPCD_NO_TECH_DETECTED
Technology Detection Step 1
Single Technology Detected
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE =
PHAC_DISCLOOP_POLL_STATE_DETECTION;
phacDiscLoop_Run(&discLoop,PHAC_DISCLOOP_ENTRY_POINT_POLL);
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE =
PHAC_DISCLOOP_POLL_STATE_COLLISION_RESOLUTION;
phacDiscLoop_Run(&discLoop,PHAC_DISCLOOP_ENTRY_POINT_POLL);
Collision Resolution And Activation Step 2
Application
Num devices resolved > 1
phacDiscLoop_ActivateCard(&discloop, tagtype, tagindex);
Success Codes:
PHAC_DISCLOOP_COLLISION_PENDING
PHAC_DISCLOOP_MULTI_DEVICES_RESOLVED
PHAC_DISCLOOP_DEVICE_ACTIVATED
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND
Failure Codes:
PHAC_DISCLOOP_NO_DEVICE_RESOLVED
PHAC_DISCLOOP_FAILURE
Single device Activated or Error
Success Codes:
PHAC_DISCLOOP_DEVICE_ACTIVATED
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND
Failure Codes:
PHAC_DISCLOOP_FAILURE
Activation Step 3

Figure 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 4. Supported technologies

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

4.2 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 established, client sends 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.

4.3 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.

4.4 Example 5 - ISO/IEC 15693

Similar to the previous example, this one is also using only one technology, in that case ISO/IEC 15693. “*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 the ICODE SLI tag is resolved and activated, the application prints out the related card information, like type and UID, and it will read from and write to the memory of the active tag.

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, check delay-related definitions in the hardware abstraction layer. Settings are described in Section 4.

4.5 Example 6 - Low-Power Card Detection (LPCD)

This examples shows calibration and usage of the LPCD.

The LPCD calibration and actual card detection loops are separated to show the capabilities and handling.

For further information on LPCD see chapter "Low-Power Card Detection" (LPCD) in the PN5180 Datasheet.

4.6 Example 7 – MIFARE Plus

The MIFARE Plus example polls and checks for MIFARE Plus EV1/EV2 cards. After the detection of a tag, according to the security level, it will demonstrate pre-personalization in security level-0, SL1-SL3 mixed mode in security level 1 and EVO/EV1 secure messaging in security level-3.

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 that 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 I^2C 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, 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. 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 is 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 Secure Files on NXP.com.

4.10 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/IEC 15693 and ISO/IEC 180003m3 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()`*

4.11 EMVCo Analog Compliance Application

This example contains three modes of operations within itself for the user to choose as below:

- EMVCo loopback application

- Trans send Type A application
- Trans send Type B application

Above application modes are used to perform EMVCo 3.0 L1 analog compliance validation. The CPU frequency configured for the host controller platforms should be sufficient enough to meet EMVCo timing requirements.

4.12 EMVCo Interoperability Compliance Application

This example is an Interoperability loopback Application which is used to perform EMVCo IOP (L1) with add-on (TTA Bulletin No.195) compliance validation.

The CPU frequency configured for the Host controller platforms should be sufficient enough to meet EMVCo timing requirements.

The running application can now be used for EMVCo IOP with add-on (TTA Bulletin No.195) Test suite validation. Test success indicated by Green LED and GPIO and fail indicated by Red LED and GPIO.

PNEV5180B v2.0 Customer evaluation board LED's and GPIO's indication:

- Red LED - LD201

- Green LED - LD203

- GPIO (indicates Success) - TXD

- GPIO (indicates Fail) - RXD

Success: Green LED On, GPIO high, 500 milliseconds delay GPIO Low and Green LED Off

Fail: Red LED On, GPIO High, 500 milliseconds delay, GPIO Low and RED LED Off

4.13 EMVCo LoopBack Compliance Application

This example is a LoopBack Application which is used to perform EMVCo3.0 L1 digital/ analog compliance validation.

The CPU frequency configured for the host controller platforms should be sufficient enough to meet EMVCo timing requirements.

4.14 ISO10373 6 PCD Compliance Application

This example is an application which 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 verify the compliance with the ISO 14443 protocols.

This application is developed based on Micropross user manual of ISO 10373-6 PCD Test Suite v2.0.2 which is executed using MP500 TCL3. It is not executed with other tool ISO 10373-6 PCD test suite.

4.15 ISO10373 6 PICC Compliance Application

This application is used to perform ISO 10373-6 PICC compliance validation. This application has to be executed using DUT which must be tested against ISO 10373-6 PICC test methods that verifies the compliance with ISO 14443 protocols.

This application is developed based on Micropross user manual of ISO 10373-6 PICC Test Suite v1.3.2 which is executed using MP500 TCL3. It is not executed with other tool ISO 10373-6 PICC test suite.

5 Flashing firmware on the LPC1769

Flashing VCOM firmware is the procedure needed to prepare PN5180 evaluation board to be used with the NFC Cockpit tool. The NFC Cockpit tool is a GUI tool designed to help finding the right analog settings for your PCB.

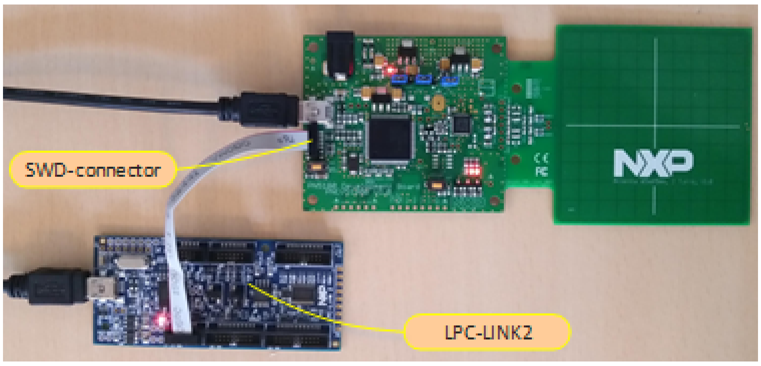
VCOM firmware which must be flashed on the LPC1769 is provided with the installer package of the NXP NFC Cockpit application. It can be found in “\NxpNfcCockpit_ v6.8.0.0\firmware\Secondary_PN5180” folder.

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:

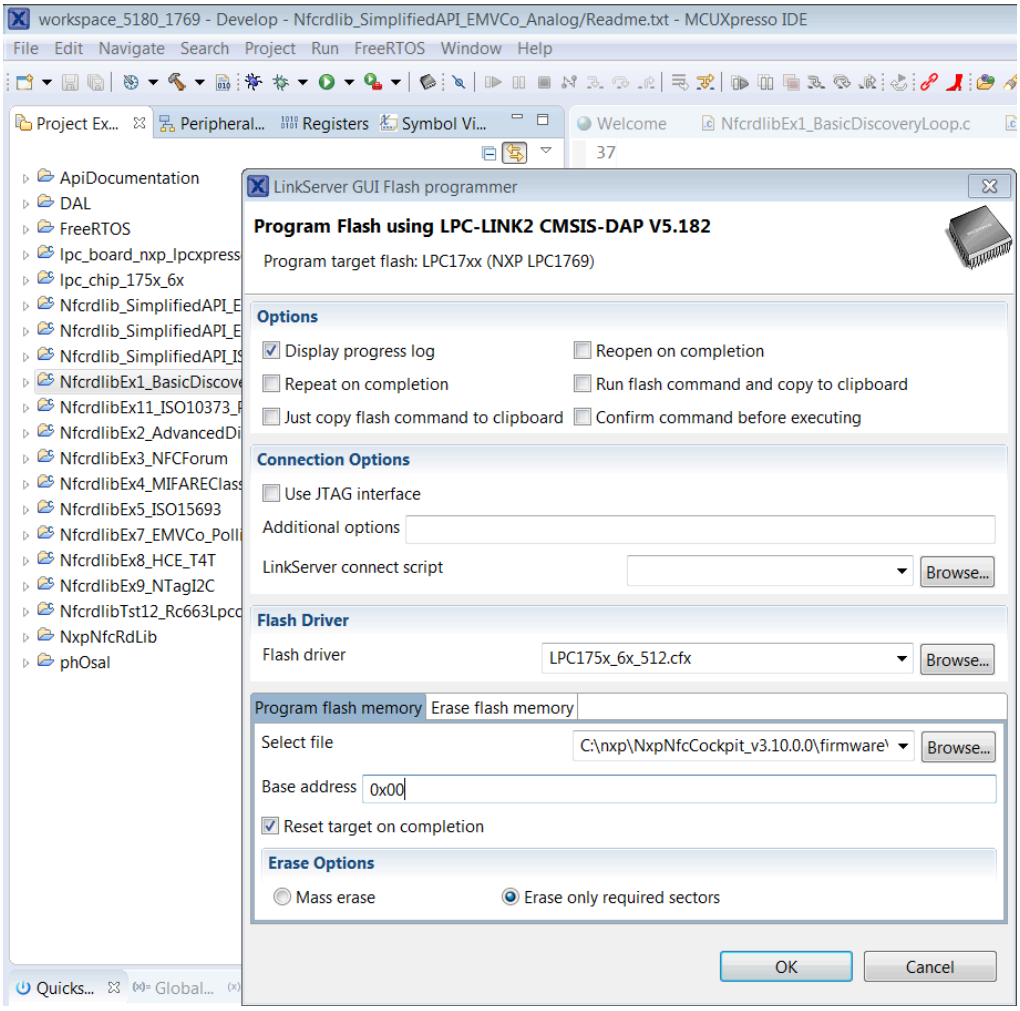
Connect the LPC-LINK2 debugger to the PN5180 board as shown in Figure 17.



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

Figure 17. LPC-Link 2 - PN5180 board connection

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



LinkServer GUI Flash programmer

Program Flash using LPC-LINK2 CMSIS-DAP V5.182

Program target flash: LPC17xx (NXP LPC1769)

Options

☑ Display progress log   □ Reopen on completion

□ Repeat on completion   □ Run flash command and copy to clipboard

□ Just copy flash command to clipboard   □ Confirm command before executing

Connection Options

□ Use JTAG interface

Additional options

LinkServer connect script   Browse...

Flash Driver

Flash driver: LPC175x_6x_512.cfx   Browse...

Program flash memory | Erase flash memory

Select file: C:\nxp\NxpNfcCockpit_v3.10.0.0\firmware\   Browse...

Base address: 0x00

☑ Reset target on completion

Erase Options

○ Mass erase   ● Erase only required sectors

OK   Cancel

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

3. Choose “Program flash” from menu bar

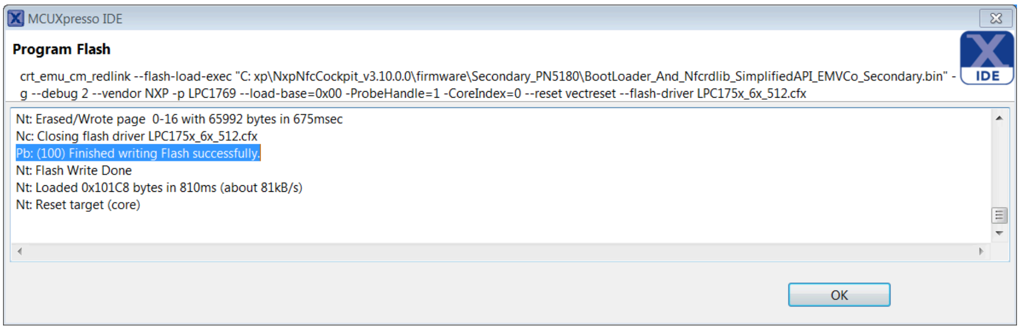
4. Select bootloader binary, for example, ‘NxpNfcCockpit_v3.10.0.0\firmware\Secondary_PN5180\BootLoader_And_Nfcrdlib_SimplifiedAPI_EMVCo_Secondary.bin’

5. Set base address to 0x00

6.Press “OK” button and flash the bootloader

Figure 18. Flashing Bootloader.bin

Flash the binary on the PNEV5180B and ensure that the process succeeds.



```
crt_emu_cm_redlink --flash-load-exec "C: xp\NxpNfcCockpit_v3.10.0.0\firmware\Secondary_PN5180\BootLoader_And_Nfcrdlib_SimplifiedAPI_EMVCo_Secondary.bin" -
g --debug 2 --vendor NXP -p LPC1769 --load-base=0x00 -ProbeHandle=1 -CoreIndex=0 --reset vectreset --flash-driver LPC175x_6x_512.cfx
```

```
Nt: Erased/Wrote page  0-16 with 65992 bytes in 675msec
Nc: Closing flash driver LPC175x_6x_512.cfx
Pb: (100) Finished writing Flash successfully.
Nt: Flash Write Done
Nt: Loaded 0x101C8 bytes in 810ms (about 81kB/s)
Nt: Reset target (core)
```

7. Inspect program flash output

Figure 19. Flash report

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

The board appears as a VCOM device:

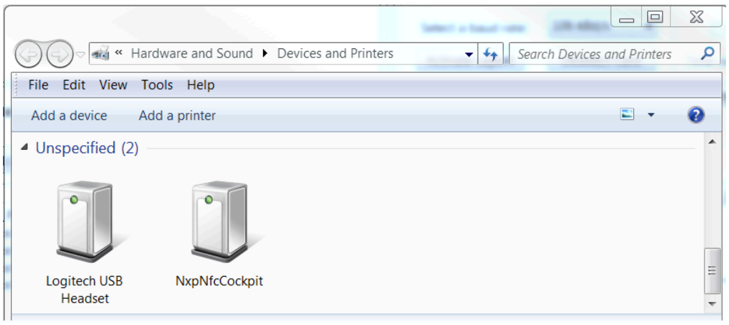

« Hardware and Sound ▸ Devices and Printers
Search Devices and Printers
File Edit View Tools Help
Add a device
Add a printer
Unspecified (2)
Logitech USB Headset
NxpNfcCockpit

Figure 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, see Figure 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)

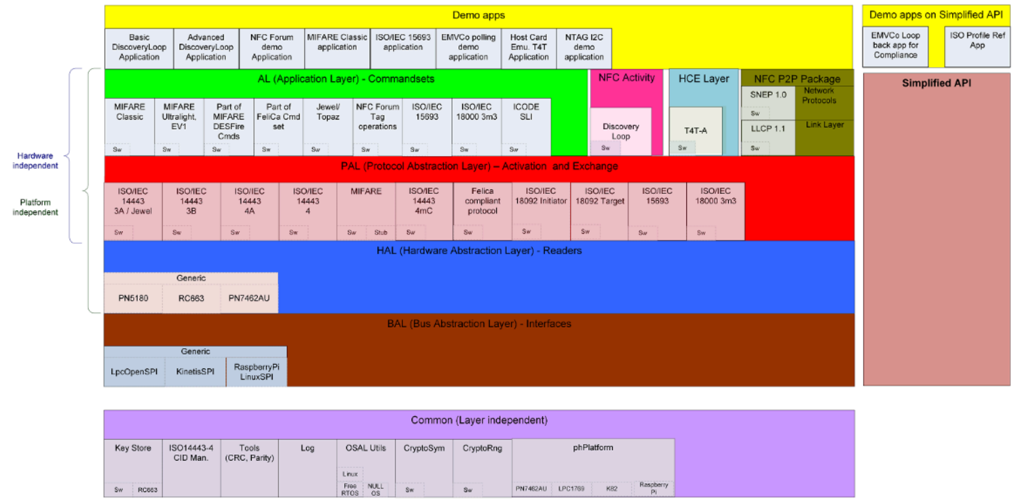


Figure 21. Architecture of the NFC Reader Library

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 their application by doing a drag-and-drop of the relevant module.

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 Section 3 are available in debug configuration. Additionally, the Polling project comprises the release configuration.

- Debug configuration

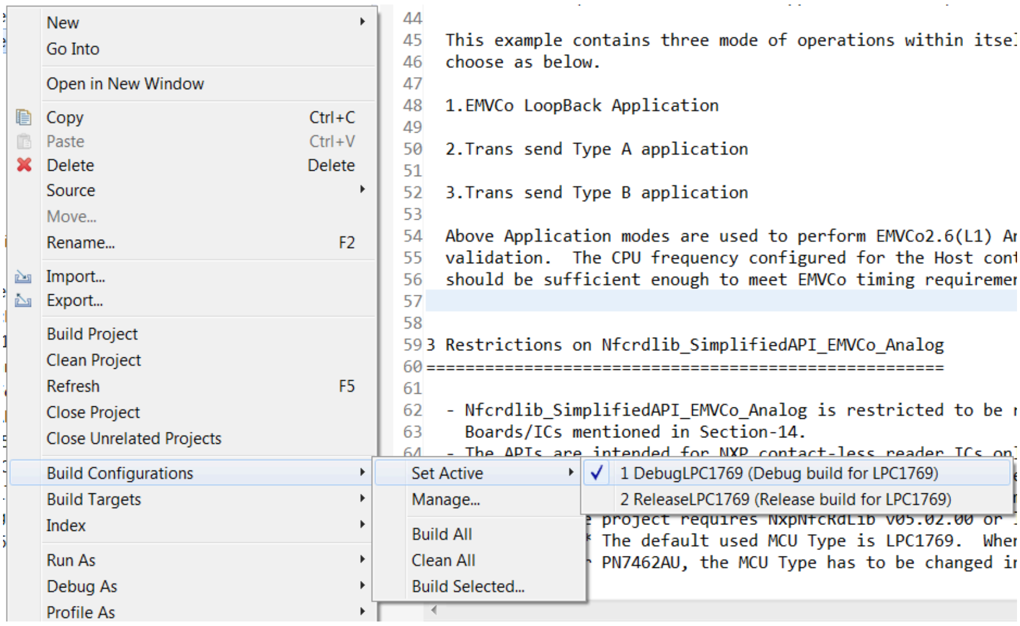
This configuration is mainly used when the target board is attached to the PC with the JTAG debugger. It allows the display of debug messages in the console window, which is useful in the early stage of the project.

- Release configuration

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

The build configuration can be selected as follows:

- Click 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.



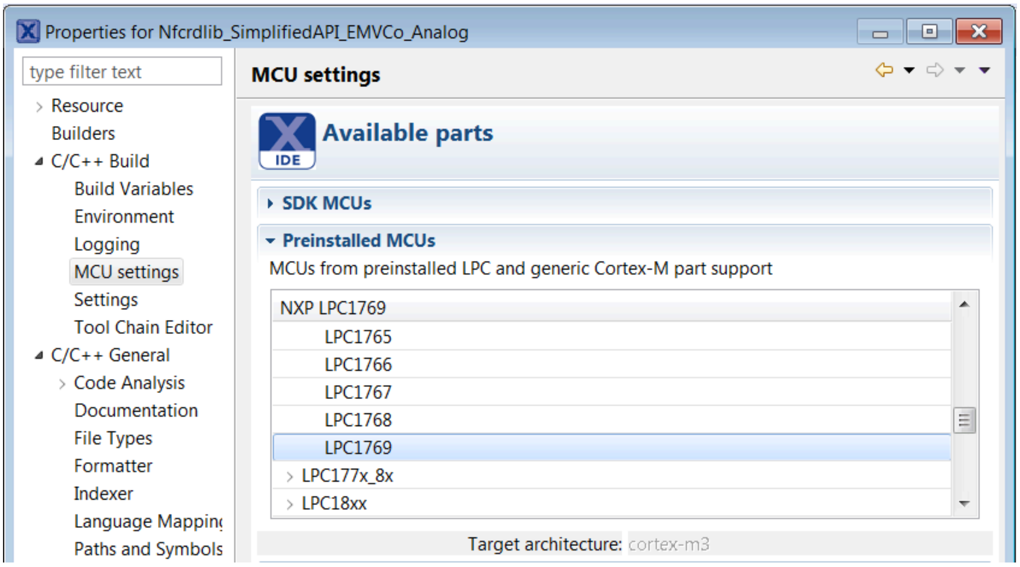
```
This example contains three mode of operations within itse
choose as below.

1.EMVCo LoopBack Application

2.Trans send Type A application

3.Trans send Type B application

Above Application modes are used to perform EMVCo2.6(L1) Ar
validation.  The CPU frequency configured for the Host cont
should be sufficient enough to meet EMVCo timing requiremer


3 Restrictions on Nfcrdlib_SimplifiedAPI_EMVCo_Analog
=====================================================

 - Nfcrdlib_SimplifiedAPI_EMVCo_Analog is restricted to be r
   Boards/ICs mentioned in Section-14.
 - The APIs are intended for NXP contact-less reader ICs on
   project requires NxpNfcRdLib v05.02.00 or
   The default used MCU Type is LPC1769.  Wher
   PN7462AU, the MCU Type has to be changed ir
```

Figure 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 must 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



MCUs from preinstalled LPC and generic Cortex-M part support

Target architecture: cortex-m3

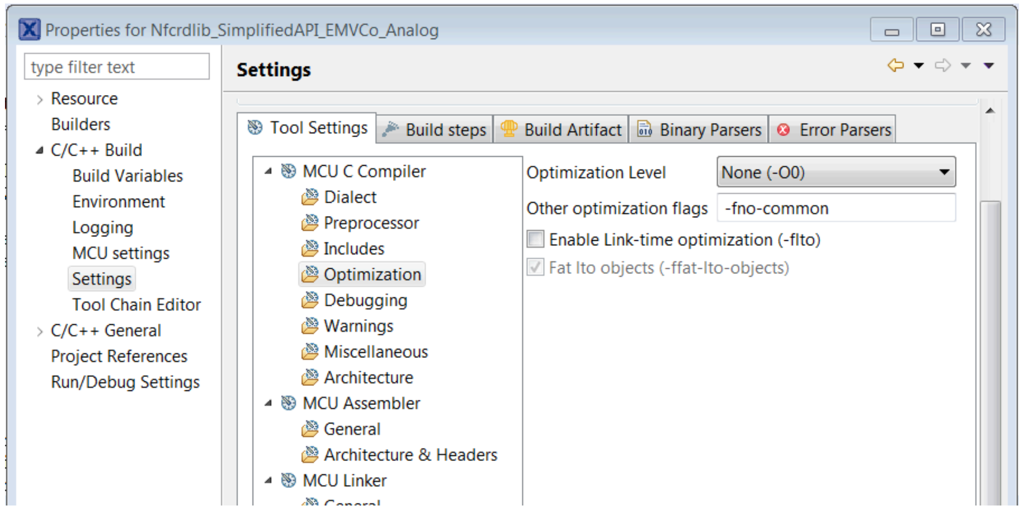
Figure 23. Selection of the LPC1769 MCU

6.4 Level of compiler optimization

When the code size at the current compiler level overloads the FLASH size of the target board (512 K for the Arm-based microcontroller LPC1769), a higher compiler optimization level can be selected to reduce the code size of the project.

The following steps can be followed to select a level of compiler optimization:

- Click the application project in the project window of the LPCXpresso IDE,
- Right click of the mouse ➔ Select properties ➔ Select C/C++ build,
- Select Settings ➔ Optimization,
- Choose the desired level in the combo box.



Optimization Level: None (-O0)

Other optimization flags: -fno-common

☐ Enable Link-time optimization (-flto)

☑ Fat lto objects (-ffat-lto-objects)

Figure 24. Select the level of optimization

6.4.1 Optimization issues

When optimization is enabled, it reorders code. What this means is that the code from multiple C lines will be intermingled. In addition, assignments and initializations might be pulled out of loops so they are only executed once. Changes like these make the code confusing to debug. Some symptoms one might see are breakpoints that only work the first time through, or seeing the debugger's current line indicator fail to advance or even move backward when clicking step. It is best to always use –O0 for debugging.

6.5 Removing the initial breakpoint on debug startup

When the debugger starts, it automatically sets a breakpoint at the first statement in the "*main()*" function. One can remove this breakpoint as follows:

1. Right-click the project and choose Launch Configurations ➔ Edit current…

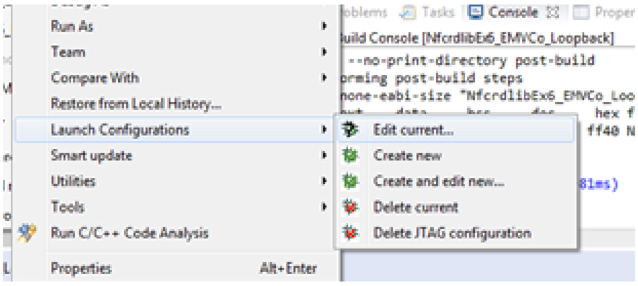
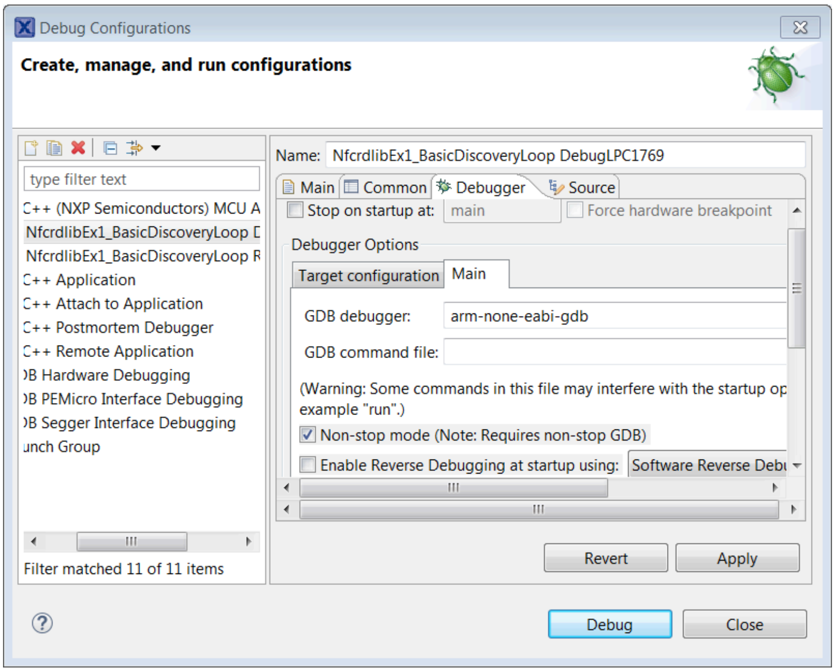


Figure 25. Edit Launch Configurations

Uncheck “Stop on startup at main” option.


Debug Configurations
Create, manage, and run configurations
type filter text
C++ (NXP Semiconductors) MCU A
NfcrdlibEx1_BasicDiscoveryLoop D
NfcrdlibEx1_BasicDiscoveryLoop R
C++ Application
C++ Attach to Application
C++ Postmortem Debugger
C++ Remote Application
DB Hardware Debugging
DB PEMicro Interface Debugging
DB Segger Interface Debugging
unch Group
Filter matched 11 of 11 items
Main
Common
Debugger
Source
Debugger Options
Target configuration
Main
Revert
Apply
Debug
Close

Name: NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769

□ Stop on startup at: main □ Force hardware breakpoint

GDB debugger: arm-none-eabi-gdb

GDB command file:

(Warning: Some commands in this file may interfere with the startup op example "run".)

☑ Non-stop mode (Note: Requires non-stop GDB)

□ Enable Reverse Debugging at startup using: Software Reverse Debu

Figure 26. Disable breakpoint at main()

7 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 5. Characteristics

PN	RF Technology	(a) Freq Ranges (EU)	(b) Max Transmitted Power
PNEV5180B	Near Field Communication	13.56 MHz +/- 7 kHz	-9 dBm

EUROPEAN DECLARATION OF CONFORMITY (Simplified DoC per Article 10.9 of the Radio Equipment Directive 2014/53/EU). This apparatus, namely PNEV5180B demo board, conforms to the Radio Equipment Directive 2014/53/EU.

The full EU Declaration of Conformity for this apparatus can be found at this location: https://www.nxp.com/products/:OM25180FDK.

8 References

[1] **MCUXpresso download website** https://www.nxp.com/design/:MCUXPRESSO

[2] **NTAG-I2C** https://www.nxp.com/products/:NT3H1101_NT3H1201

[3] Application note - AN11802 NFC Reader Library for Linux Installation Guidelines

[4] Application note - AN11908 NFC Reader Library for FRDM guidelines

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.

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.

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.

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

DESFire — is a trademark of NXP B.V.

FeliCa — is a trademark of Sony Corporation.

ICODE and I-CODE — are trademarks of NXP B.V.

Kinetis — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

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

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

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

NTAG — is a trademark of NXP B.V.

Tables

Tab. 1. Example projects 3
Tab. 2. Compliance applications 3
Tab. 3. Development environment 5
Tab. 4. Supported technologies 16
Tab. 5. Characteristics 29

Figures

Fig. 1. Checkbox for NXP debug drivers 6
Fig. 2. Windows security dialog 6
Fig. 3. MCUXpresso IDE 6
Fig. 4. Importing project (1) 7
Fig. 5. Importing project (2) 8
Fig. 6. Importing project (3) 8
Fig. 7. Building the project 9
Fig. 8. List of projects in “Project Explorer view 9
Fig. 9. PNEV5180B with LPC-Link 2 10
Fig. 10. Launch debug session 11
Fig. 11. Successful compile 11
Fig. 12. Select the launch configuration 12
Fig. 13. Debug project 12
Fig. 14. Application printouts in console window 13
Fig. 15. Debug buttons 13
Fig. 16. Discovery Loop in Poll mode 16
Fig. 17. LPC-Link 2 - PN5180 board connection 21
Fig. 18. Flashing Bootloader.bin 22
Fig. 19. Flash report 23
Fig. 20. PN5180 Board Correctly installed 23
Fig. 21. Architecture of the NFC Reader Library 24
Fig. 22. Select the build configuration 26
Fig. 23. Selection of the LPC1769 MCU 26
Fig. 24. Select the level of optimization 27
Fig. 25. Edit Launch Configurations 28
Fig. 26. Disable breakpoint at main() 28

Contents

1 Introduction 3
2 Managing the PN5180 SW projects with MCUXpresso IDE 5
2.1 Development environment 5
2.2 Installation procedure of the MCUXpresso IDE 5
2.3 Importing provided SW example projects 7
2.4 Building projects 9
2.5 Running and debugging a project 10
3 Managing the PN5180 SW projects with Linux and Kinetis platform 14
4 Associated projects 15
4.1 Example 1 – Basic Discovery Loop 15
4.2 Example 3 – NFC Forum 16
4.3 Example 4 – MIFARE Classic card communication 17
4.4 Example 5 - ISO/IEC 15693 17
4.5 Example 6 - Low-Power Card Detection (LPCD) 17
4.6 Example 7 – MIFARE Plus 17
4.7 Example 8 – HCE T4T 18
4.8 Example 9 – NTAG-I2C 18
4.9 Example 10 – MIFARE DESFire card communication 18
4.10 Simplified API ISO 18
4.11 EMVCo Analog Compliance Application 18
4.12 EMVCo Interoperability Compliance Application 19
4.13 EMVCo LoopBack Compliance Application 19
4.14 ISO10373 6 PCD Compliance Application 19
4.15 ISO10373 6 PICC Compliance Application 19
5 Flashing firmware on the LPC1769 21
6 Supplementary notes 24
6.1 General software architecture 24
6.1.1 Bus abstraction layer 24
6.1.2 Hardware abstraction layer 24
6.1.3 Protocol abstraction layer 24
6.1.4 Application layer 25
6.2 Build configuration 25
6.3 Setting the MCU 26
6.4 Level of compiler optimization 27
6.4.1 Optimization issues 27
6.5 Removing the initial breakpoint on debug startup 27
7 Radio Equipment Directive (RED) 29
8 References 30
9 Legal information 31

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

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

Date of release: 5 December 2022
Document identifier: UM10954