UM10954

PN5180 SW Quick start guide

1.4 — 17 January 2017
345814

User manual
COMPANY PUBLIC

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
1.4	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

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.

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 standard MIFARE 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 ISO	Explains how to use simplified API with different types of cards.

2. Managing the PN5180 SW projects with LPCXpresso IDE

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

The LPCXpresso 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. LPCXpresso 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 LPCXpresso IDE can build an executable of any size with full code optimization.

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

This tool can freely be downloaded from the LPCXpresso website [1]. Before one can download the software, it is necessary to create an account. Creating an account is absolutely 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)
LPCXpresso IDE	8.1.4 or higher	Development IDE (PC software)

2.2 Installation procedure of the LPCXpresso IDE

The LPCXpresso IDE is installed into a single directory, of your choice. Unlike many software packages, the LPCXpresso 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.

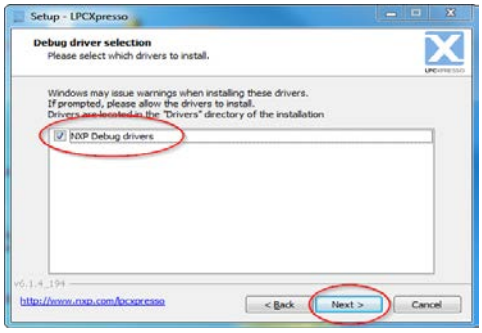


Fig 1. Check box for NXP debug drivers

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. LPCXpresso IDE

Directly after the first start of the LPCXpresso IDE, an info dialog will appear with the message of an unregistered copy of the LPCXpresso IDE.

Confirm the dialog and follow the instructions on the Welcome Screen to get a registered version with the debug limit of 256k. The registration is free of a charge. The link to the

registration page is shown in the menu, Help → Activate LPCXpresso → Create Serial number and register.

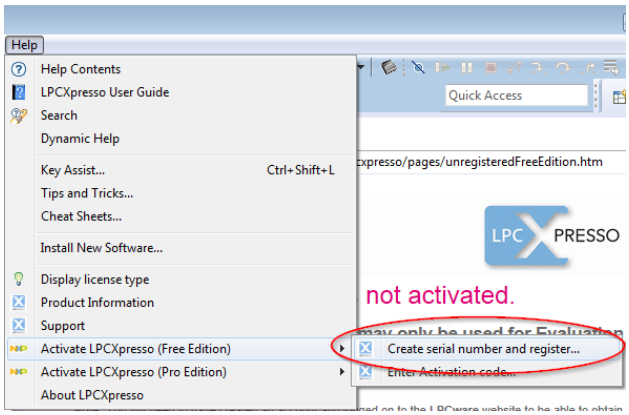


Fig 4. Product activation (1)

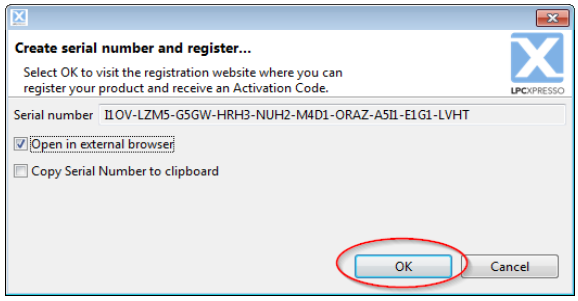


Fig 5. Product activation (2)

In case that you do not have an account on the http://www.nxp.com/redirect/lpcware.com, simply sign up to get an activation code. The code will be sent to the provided e-mail address.

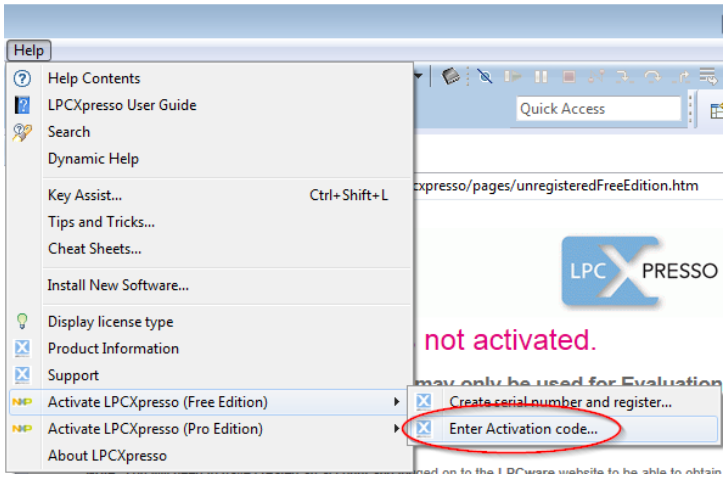


Fig 6. Product activation (3)

Once you receive the activation code open the activation window by pointing to Help → Activate LPCXpresso → Enter Activation code, and enter the code.

The success of the product activation will be confirmed by an info dialogue.

2.3 Importing provided SW example projects

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

The sequence of installing the software projects is indicated:

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

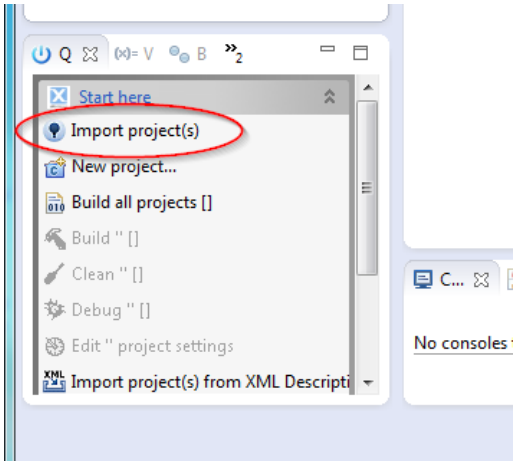


Fig 7. Importing project (1)

In the Quick Panel on the left hand side, choose “Import projects(s)”.

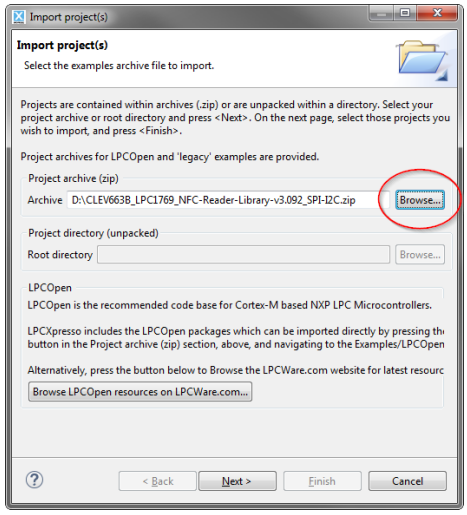


Fig 8. Importing project (2)

Browse the desired package and click “Next”.

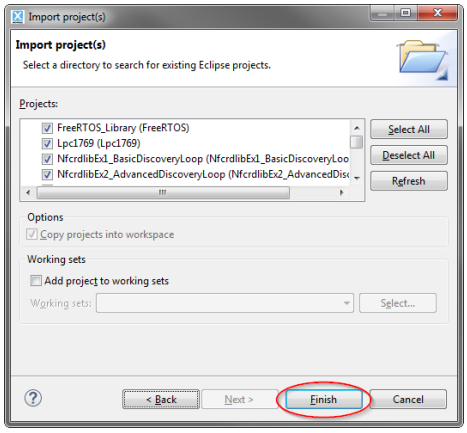


Fig 9. Importing project (3)

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.

Before one can run the project, the LPCXpresso board containing the PN5180 Blueboard needs to be connected to the computer. Wait until the adequate drivers have been installed.

2.4 Building projects

Building projects in a workspace is a simple case of using the Quick start 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 10.

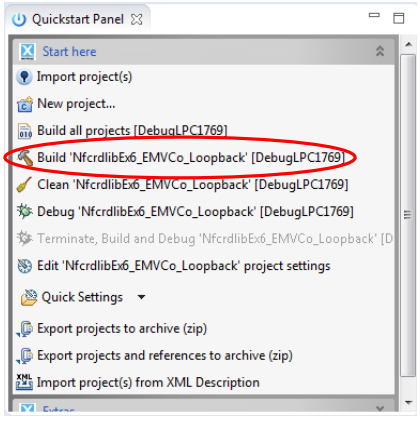


Fig 10. Building the project

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 PN5180 User Flash via USB mass storage interface.

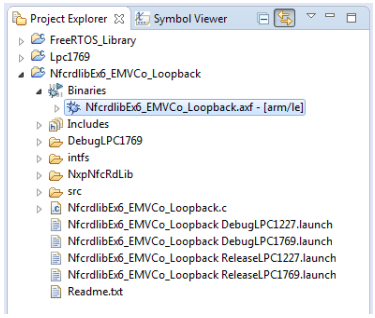


Fig 11. Figure title here

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 “*NfcrdlibEx6_EMVCo_Loopback*” 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.

First of all you need to ensure that your PN5180 evaluation board is connected to the computer via LPC-LINK2, as shown in Fig 12.

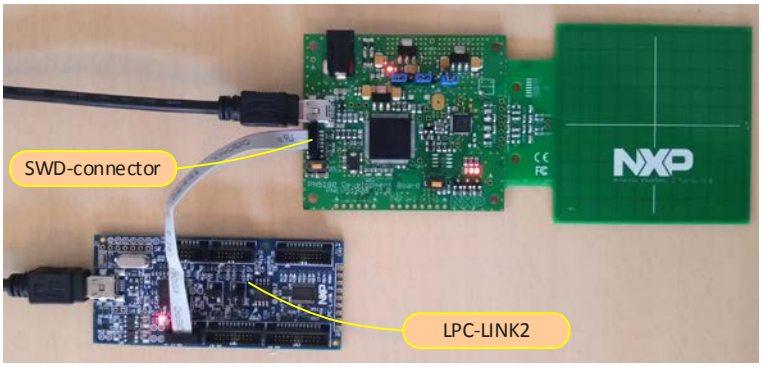


Fig 12. PNEV5180B with LPC-Link2

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 Quick start Panel click Debug, as shown in Fig 13. The LPCXpresso 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 26 and chapter 6.3.

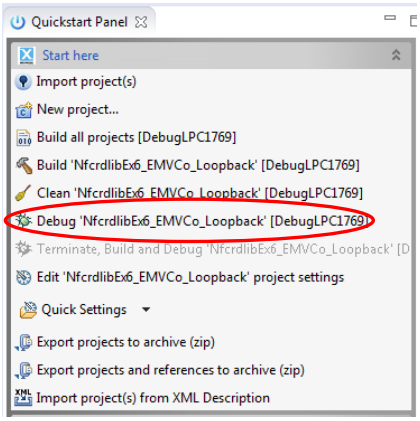


Fig 13. Launch debug session

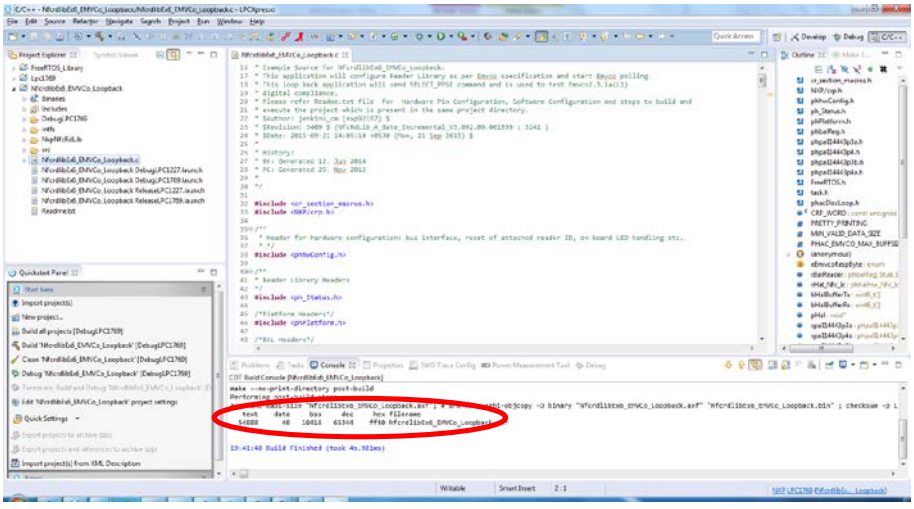


Fig 14. Successful compile

Select “LPC-LINK2 REDLINK” as a debug emulator.

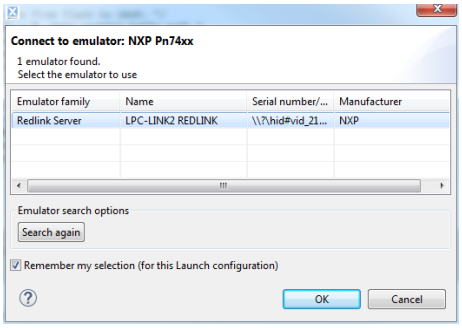


Fig 15. Select the launch configuration

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

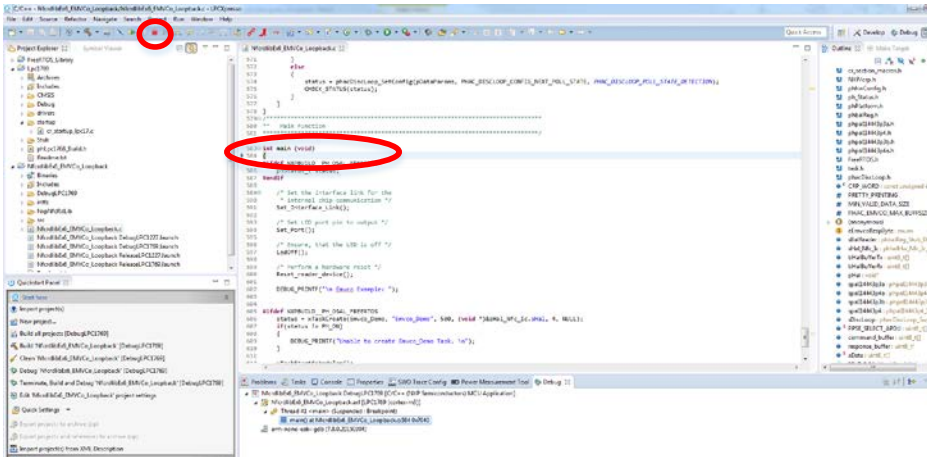


Fig 16. Debug project

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

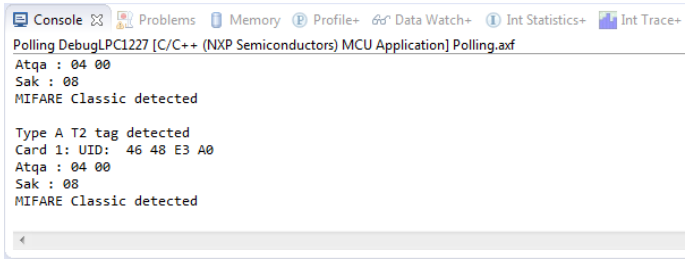


Fig 17. Application printouts in Console window

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.
	Instruction stepping mode (disassembly).

Fig 18. Debug Buttons

3. Managing the PN5180 SW projects with Linux and KDS IDE

Detailed description and guideline, how to import and manage NFC NXP Reader Library projects in Linux and Kinetis Design Studio (KDS) environment, check:

- AN11802 - NFC Reader Library for Linux Installation Guidelines
- AN11908 - NFC Reader Library for KDS Installation Guidelines

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 make a copy of all documents and SW on the hard disk.

By default the projects are preconfigured to be run on the LPCXpresso LPC1769 development board. For instructions about how to run the projects on the LPCXpresso LPC11U68 development board, refer to chapter 6.2 and 6.3, please.

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 settings in the file "*../intfs/ph_NxpBuild_App.h*" needs to be configured properly.

E.g. enable FreeRTOS

```
//#define NXPBUILD__PH_OSAL_NULLOS
#define NXPBUILD__PH_OSAL_FREERTOS
```

ICODE support

In order to detect ICODE SLI, SLIX and SLIX2 cards appropriate HAL digital delay should be enabled. Define settings in the file "*../intfs/ph_NxpBuild_App.h*" needs to be configured properly.

E.g. enabling HAL digital delay

```
#ifdef NXPBUILD__PHHAL_HW_PN5180
    #define PN5180_HAL_DIGITAL_DELAY    /**< Enable to …
#endif /* NXPBUILD__PHHAL_HW_PN5180 */
```

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 for MIFARE Cards) 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.

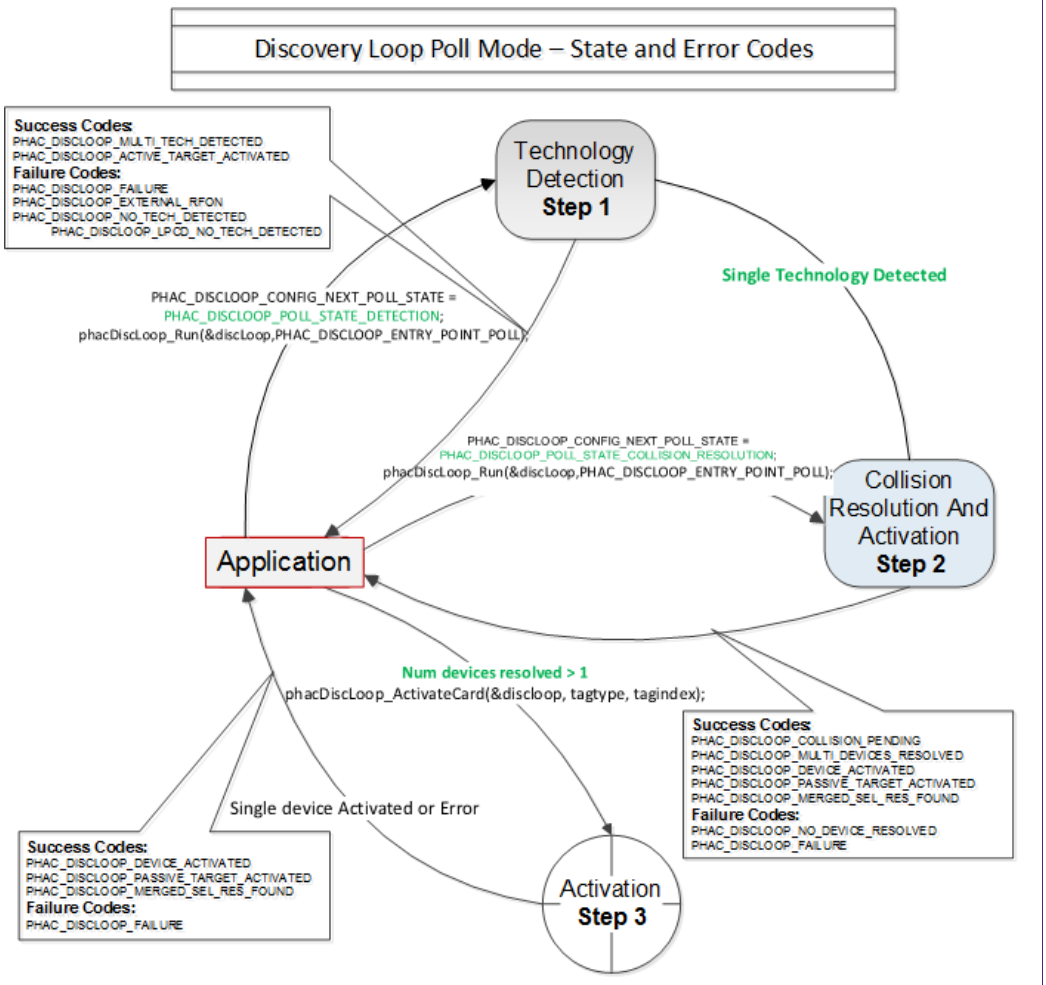


Fig 19. Discovery Loop in Poll mode

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	

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

This example demonstrate 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 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

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 Microprosss 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 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, MIFARE DESFire, MIFARE Ultralight, MIFARE Classic, ISO5693 and ISO18000p3m3.

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

- *phNfcLib_Activate()*
- *phNfcLib_Transmit()*
- *phNfcLib_Receive()*

5. Flashing Firmware on the LPC1769

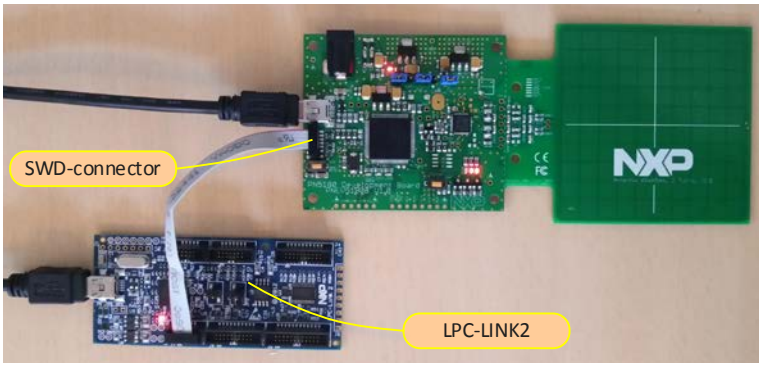
Flashing VCOM firmware is the procedure needed to prepare PN5180 evaluation board to be used with the Cockpit tool. The Cockpit tool is software design tool for prototyping NXP NFC card reader applications.

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_v3.7.0.0\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

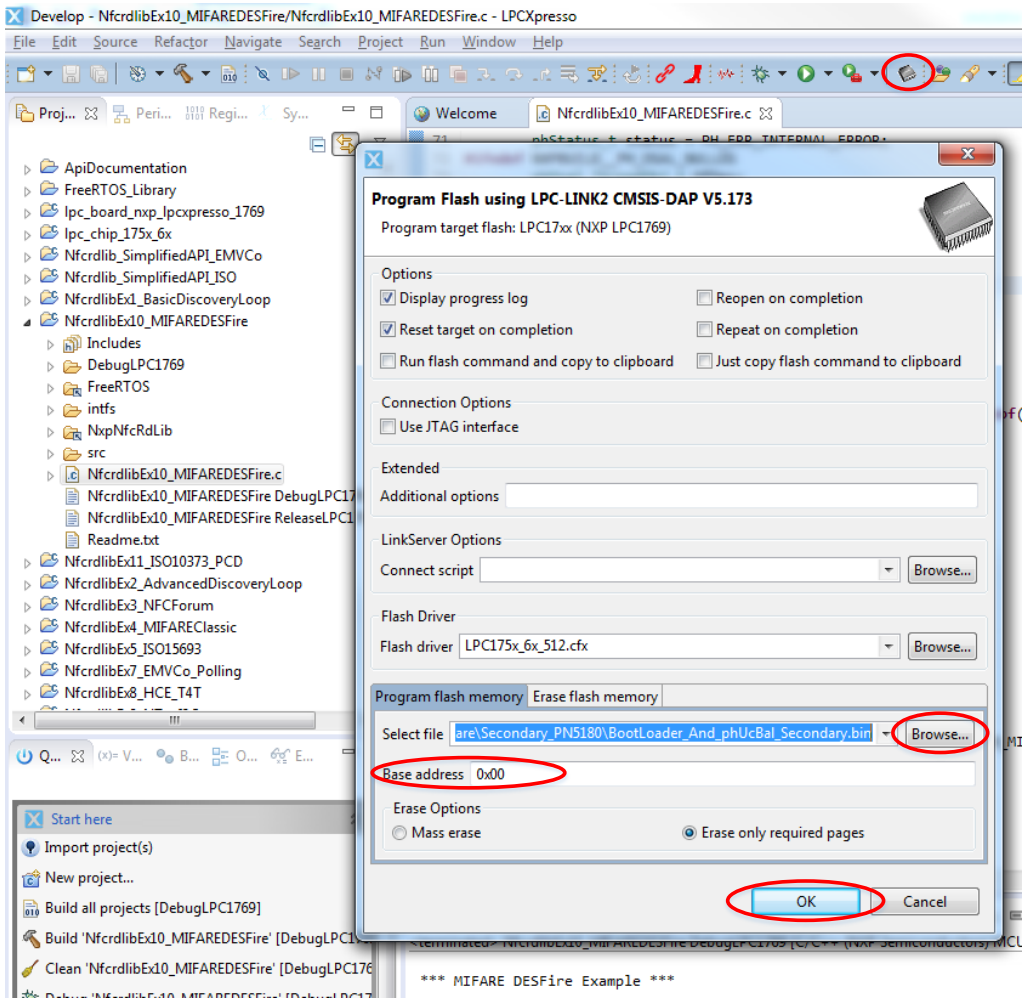
Steps required to flash VCOM application:
Connect the LPC-LINK2 debugger to the PN5180 board as shown in the Fig 20.



(1) Use an LPC-LINK2 debugger and make the connections with PN5180 board

Fig 20. LPC-LINK2 - PN5180 board connection

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



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

(3) Chose “Program flash” from menu bar

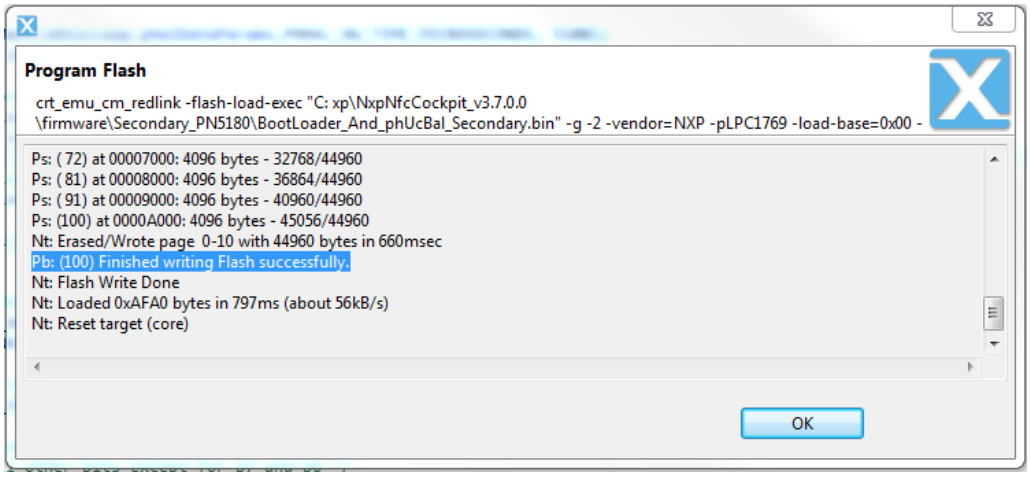
(4) Select bootloader binary, e.g. ‘NxpNfcCockpit_v3.7.0.0\firmware\Secondary_PN5180\BootLoader_And_Nfcrdlib_SimplifiedAPI_EMVCo_Secondary.bin’

(5) Set base address to 0x00

(6) Press “OK” button and flash the bootloader

Fig 21. Flashing Bootloader.bin

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



(7) Inspect program flash output

Fig 22. Flash report

Disconnect the USB cable, remove LPC-LINK2 connection and reset the board.

The board appears as a VCOM device:

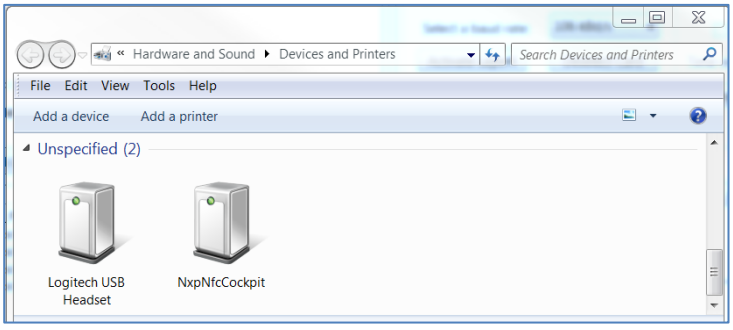


Fig 23. PN5180 Board Correctly installed

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 24. 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)

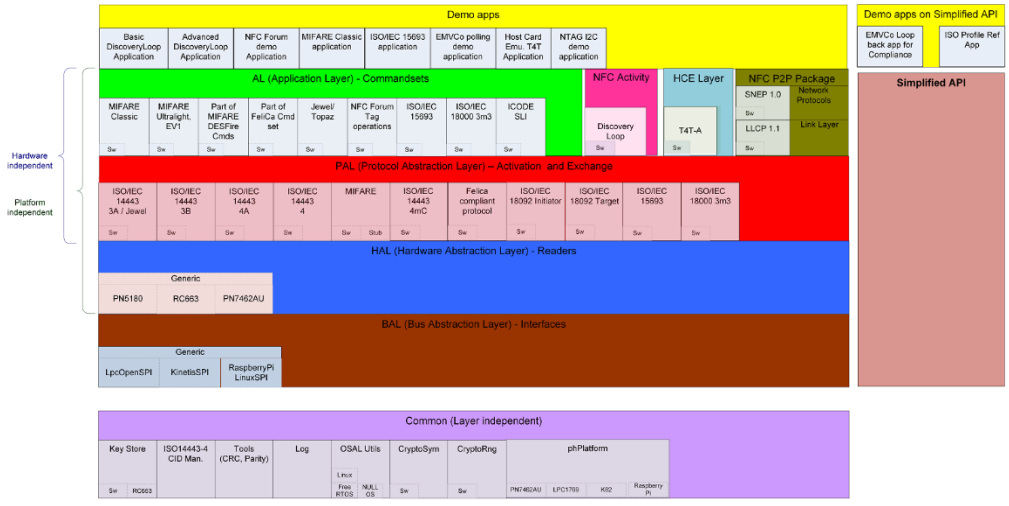


Fig 24. Architecture of the NFC Reader Library

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.

The following PAL modules are available in this software package:

- ISO/IEC 14443-3A,
- ISO/IEC 14443-3B,
- ISO/IEC 14443-4A/B,
- MIFARE,
- 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 LPCXpresso IDE,
- Right click of the mouse → Select Build Configuration,
- Set active DebugLPC1769 build (or ReleaseLPC1769 build) for LPC1769.

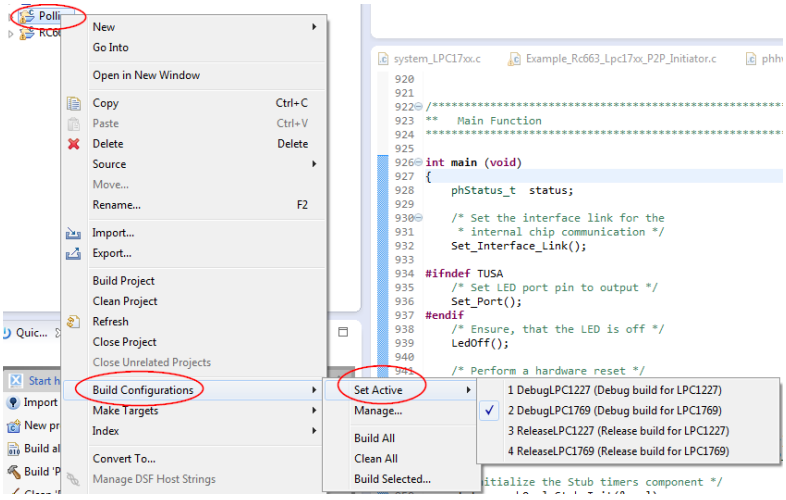


Fig 25. Select the build configuration

6.3 Setting the MCU

There are many LPC microcontrollers supported by the LPCXpresso 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 LPC1xxx MCU group → choose the correct microcontroller → click OK

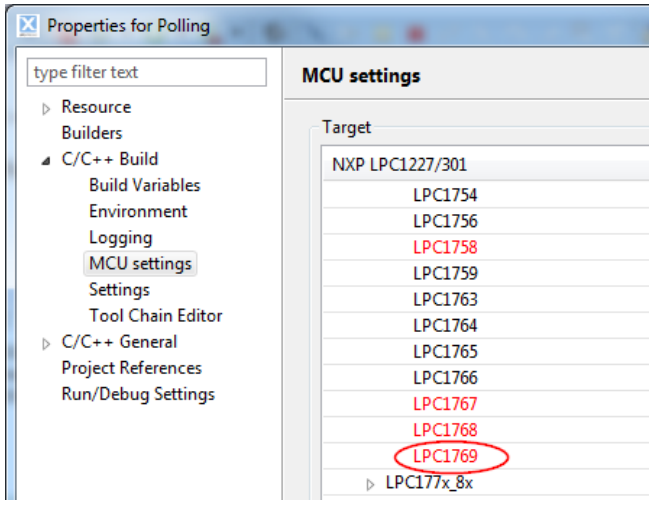


Fig 26. Selection of the LPC1769 MCU

6.4 Level of compiler optimization

When the code size at the current compiler level overloads the FLASH size of the target board (512K 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 on 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.

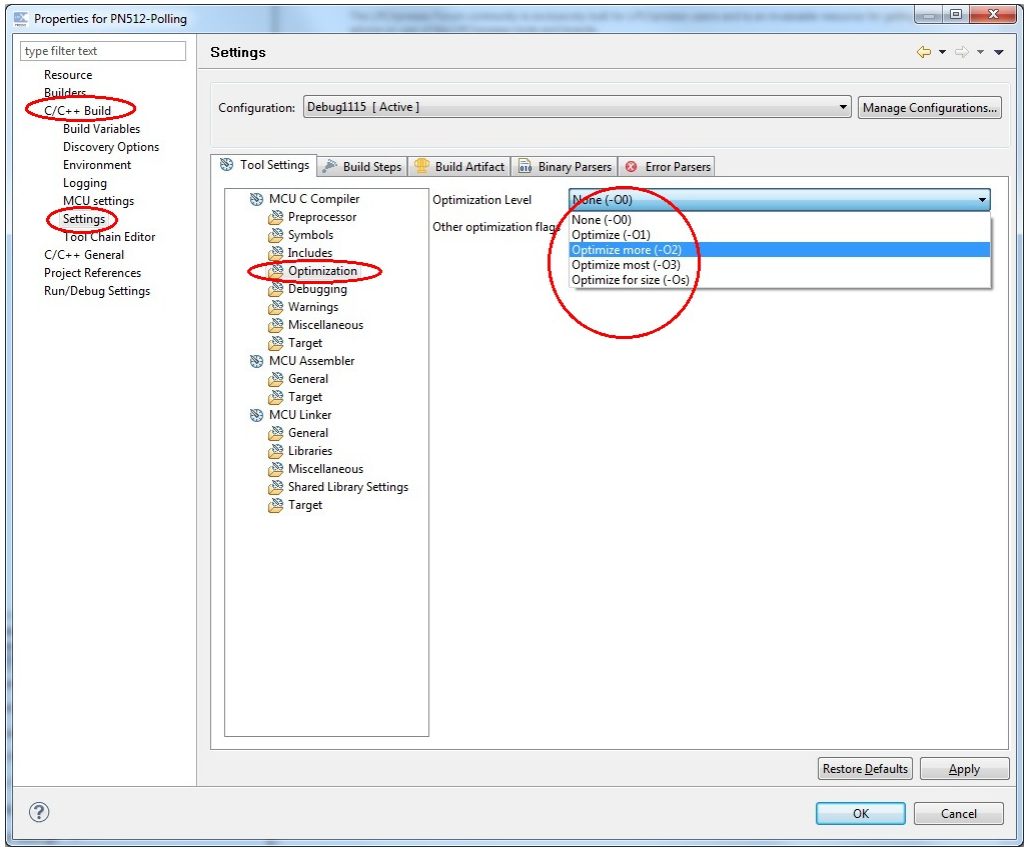


Fig 27. Select the level of optimization

6.4.1 Optimization issues

When optimization is enabled, it will reorder 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 will 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 backwards 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 on the project and choose Launch Configurations → Edit current…

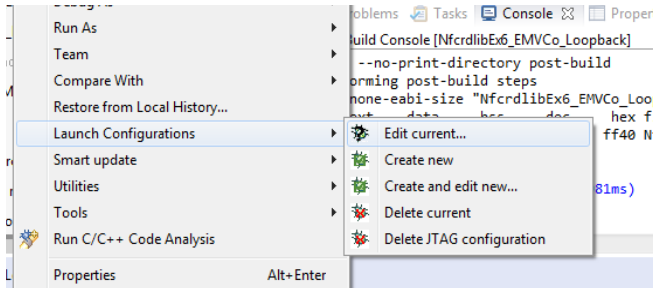


Fig 28. Edit Launch Configurations

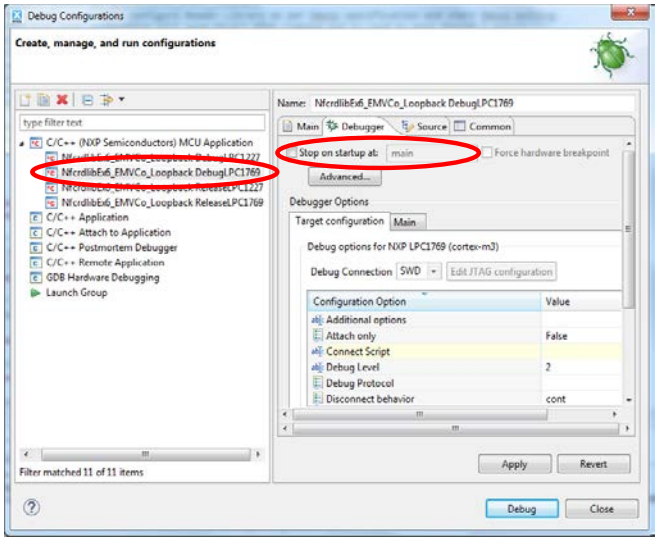


Fig 29. Disable break point at main()

7. References

[1] **LPCXpresso download website**
http://www.nxp.com/redirect/lpcware.com/lpcxpresso/download

[2] **NTAG-I2C**
http://www.nxp.com/products/identification_and_security/nfc_and_reader_ics/connected_tag_solutions/series/NT3H1101_NT3H1201.html

[3] AN11744 PN5180 evaluation board quick start guide, www.nxp.com

[4] AN11742 PN5180 Dynamic Power Control, www.nxp.com

8. Legal information

8.1 Definitions

Draft — The document is a draft version only. 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 herein and shall have no liability for the consequences of use of such information.

8.2 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.

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 accepts 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.

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.

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.

8.3 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.

8.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

MIFARE — is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V

ICODE — is a trademark of NXP B.V

Kinetis — is a trademark of NXP B.V

9. List of figures

10. List of tables

11. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 17 January 2017
345814
Document identifier: UM10954