



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

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

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



## Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



# AN11281

## Quick Start Up Guide PREV601 Demo Board

Rev. 1.5 — 10 November 2013  
243915

Application note  
COMPANY PUBLIC

### Document information

Info	Content
<b>Keywords</b>	PREV601 board, PRH-601, LPC1227, CLRC663, LPCXpresso, LPCD, NxpRdLib, HTRC110, MCU, Code Red, eclipse, LPC1227, reader library
<b>Abstract</b>	This application note is related to the installation procedures of the PREV601 board. It describes the actions to be done to become acquainted with the demo reader.



**Revision history**

Rev	Date	Description
1.5	20131110	Added a note about the LPCXpresso IDE version in chapter 4
1.4	20130613	Added description about the P2P Snep Client
1.3	20130608	Small alignments between new project structure and this document
1.2	20130116	Small alignments between the referenced code and this guide.
1.1	20130107	Added chapter 6.4 about the hardware setting Added chapter 6.5 about the LPCD mode
1.0	20121019	First release

**Contact information**

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

## 1. Introduction

This application note gives a detailed overview of the hardware for working with the PRH601 integrated reader module (**Chapter 2**), the installation procedures of the Development Environment (**Chapter 4.1**) and the handling of the reader projects using the NXP Reader Library (**Chapter 4.2**).

The PREV601 board can be used for software development for the PR601 as well as the PRH601 reader IC. The only difference between these ICs is the additional 125kHz capability of the PRH601 IC.

The projects used in this documentation are:

- Communication with MIFARE Ultralight → **Chapter 5.1**
- Communication with MIFARE Classic → **Chapter 5.2**
- Communication with MIFARE DESFire → **Chapter 5.3**
- Polling for Tags in the RF - field → **Chapter 5.4**

All projects can also operate in Low Power Card Detection Mode [11].

## 2. Hardware overview of the Demo Reader

The demo reader is made up of one single board including two reader ICs and one processor with flash memory. The IC that contains these reader ICs and the processor is the PRH601HL. For more details, please see our website at [9]. The multi chip package contains the following ICs:

- A CLRC663 Reader IC. For detailed information please see [2].
- A HITAG HTRC110. For detailed information please see [7].  
The HTRC110 is supported by breakout pads. A developer has to attach his own antenna incl. matching to make use of the 125KHz functionality of the PRH601. Software for the 125KHz functionality can be found on the NXP website [8].
- A LP1227 IC. For detailed information please see [3]

To flash the LPC1227 micro controller, one also needs a LPC-Link board (LPC3154)

### 2.1 PREV601 board

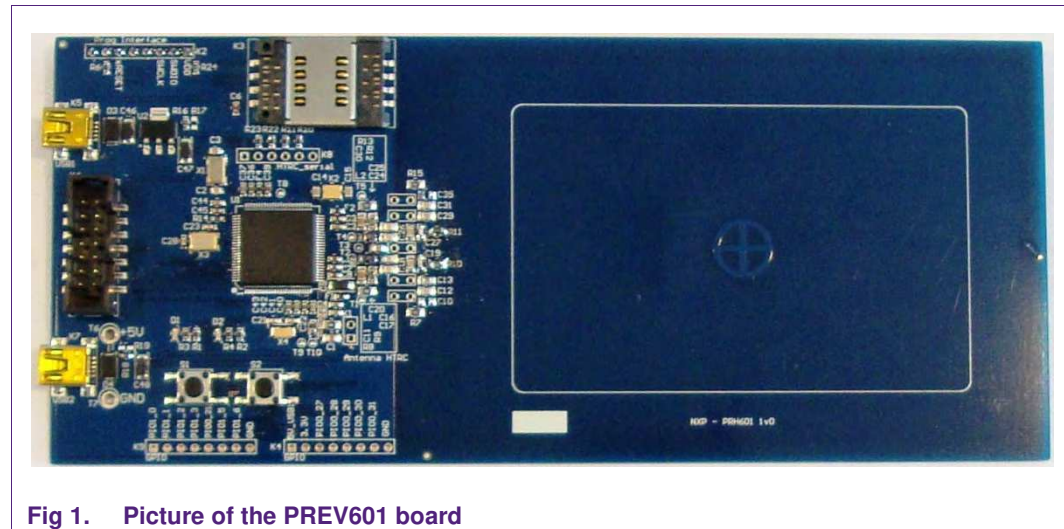


Fig 1. Picture of the PREV601 board

The PREV601 board embeds the contactless communication transceiver IC CLRC663, HTRC and LPC1227 processor.

The software provided with this evaluation board also works with the significantly smaller PREV601M Microboard.

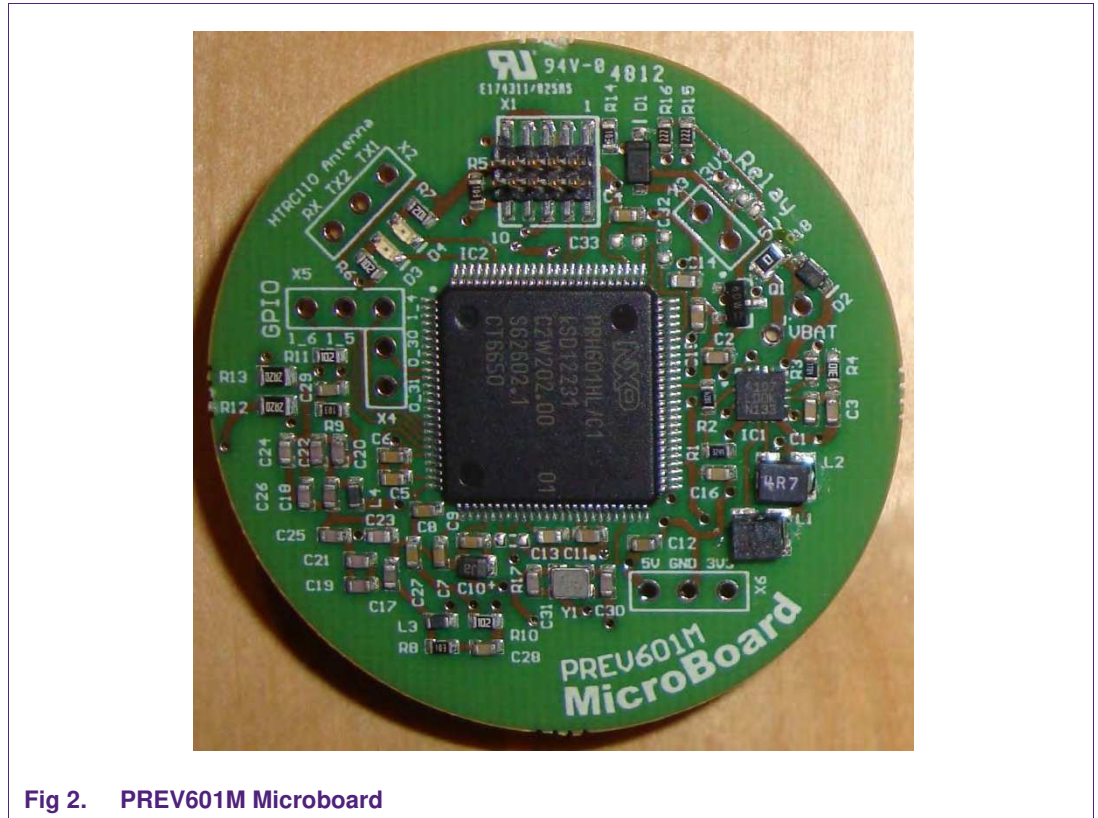


Fig 2. PREV601M Microboard

## 2.2 LPC-Link LPC3154

The LPC-Link is equipped with an 8-pin JTAG header and it seamlessly connects the target via USB (the USB interface and other debug features are provided by NXP’s ARM9 based LPC3154 MCU).

Since this LPC-Link is not available as single device, one has to cut that part from any LPCXpresso board. The LPCXpresso boards containing the LPC-Link with the LPC3154 MCU are as follows:

- LPC11C24 LPCXpresso Board
- LPC11U14 LPCXpresso Board
- LPC1114 LPCXpresso Board
- LPC1115 LPCXpresso Board
- LPC1227 LPCXpresso Board
- LPC1347 LPCXpresso Board
- LPC1343 LPCXpresso Board
- LPC1769 LPCXpresso Board

To use the LPC-Link, one has to cut it off from the LPCXpresso board.

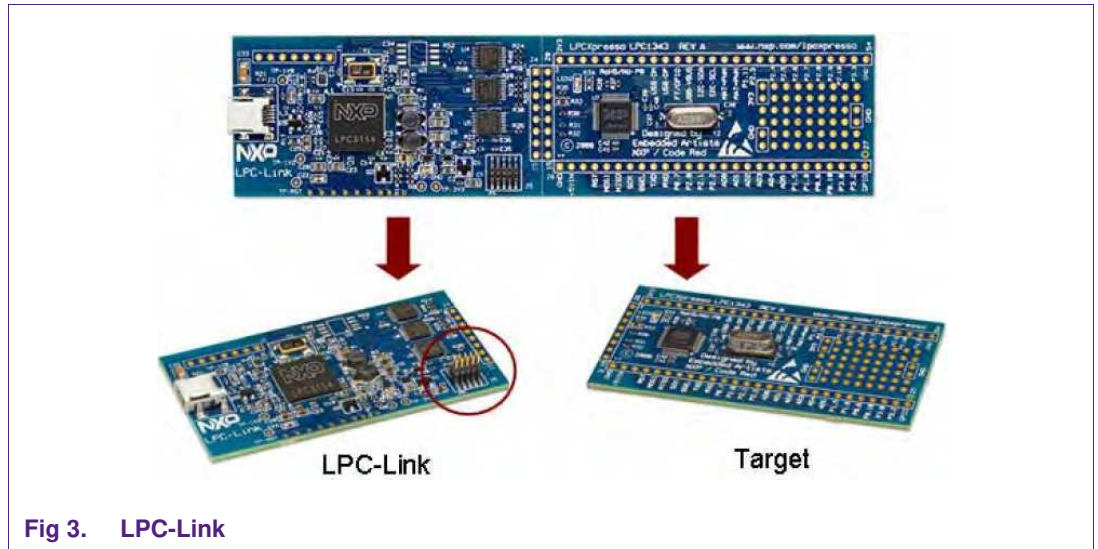


Fig 3. LPC-Link

### 2.3 Preparation of the hardware

The first step after unpacking the PREV601 board and the LPC-Link board is soldering the connectors onto the boards in order to attach them. In our example we use a multipoint connector as one can see on the pictures below.

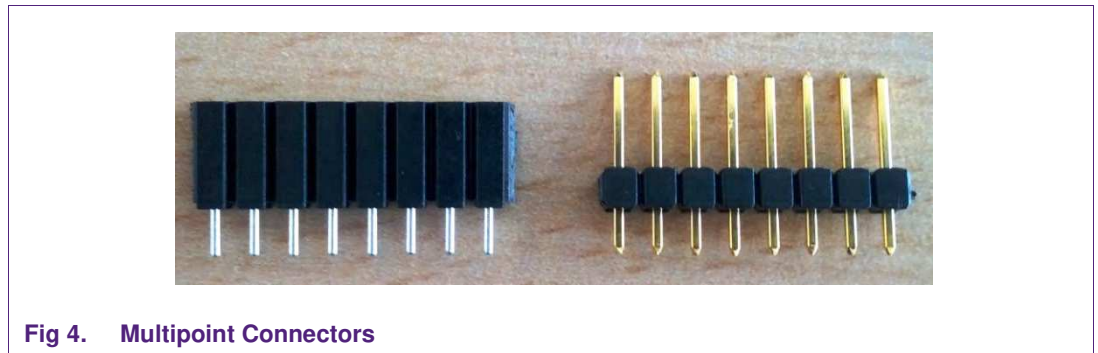


Fig 4. Multipoint Connectors

One may buy these connectors at any electronic store. Here are some examples [\[4\]](#). After soldering the connectors, connect the boards as shown on the following figure.

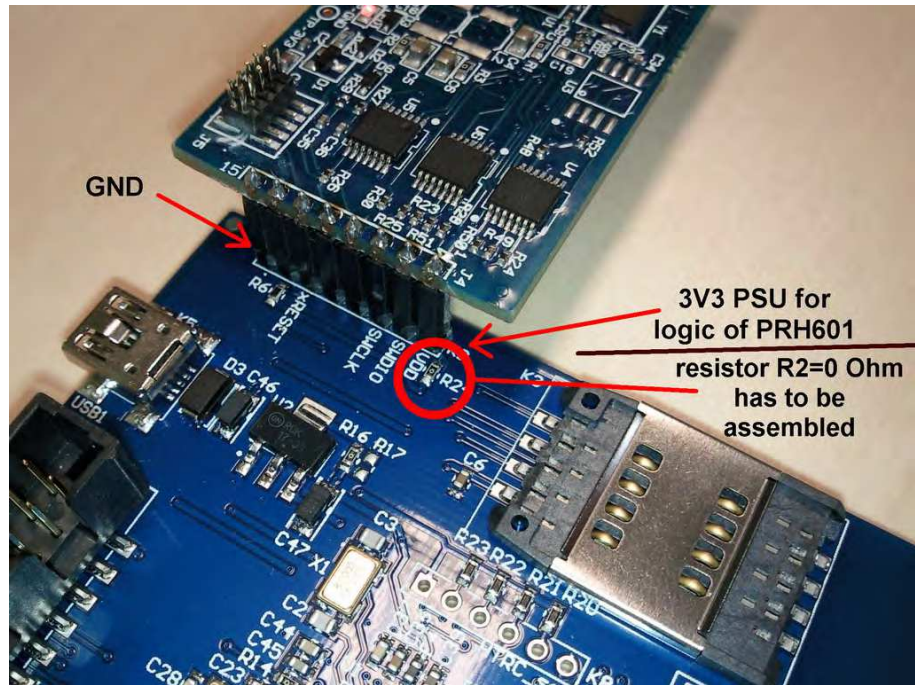


Fig 5. Connecting the PREV601 board to the LPC Link



Fig 6. LPCpresso with the Multipoint Connectors

While soldering the connectors onto the board, an opportunity arises to solder either a bridge or - as shown on Fig 5- a zero ohm resistor near the program interface (R24). After bridging this connection, you'll need two instead of three USB cables connected like shown on Fig 6.

Now the hardware is ready for use.

### 3. Connecting the hardware to a computer

The guidelines to install the reader are as follows:

- Connect the LPC-Link board to a real USB2.0 port of the PC using the mini-USB connector. The PC detects and installs the board automatically.
- Once the board has been installed, open the Device Manager of the PC to check that the installation has been successful. The item “USB Device with DFU Capabilities” is being displayed.

After connecting the LPC-Link board, please also connect the second USB connector on the PREV601 board to the computer. In summary one will need to connect 2 USB cables (see picture above).

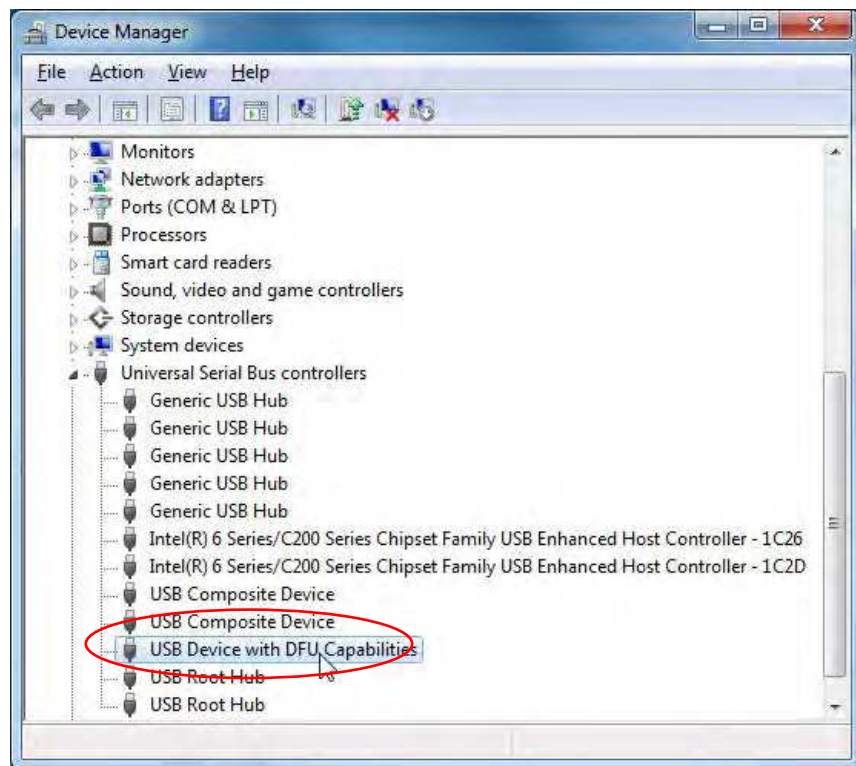


Fig 7. Enumeration of the LPCXpresso Board in Device Manager Window

### 4. Managing the Demo Reader project with LPCXpresso IDE

The demo reader project is delivered in a zip package. It can be extracted, edited, compiled and linked with LPCXpresso™ IDE.

LPCXpresso™ is a new, low-cost development platform available at NXP. It supports NXP's ARM-based LPC microcontrollers. The platform comprises a simplified Eclipse-based IDE and low-cost target boards which include an attached JTAG debugger.

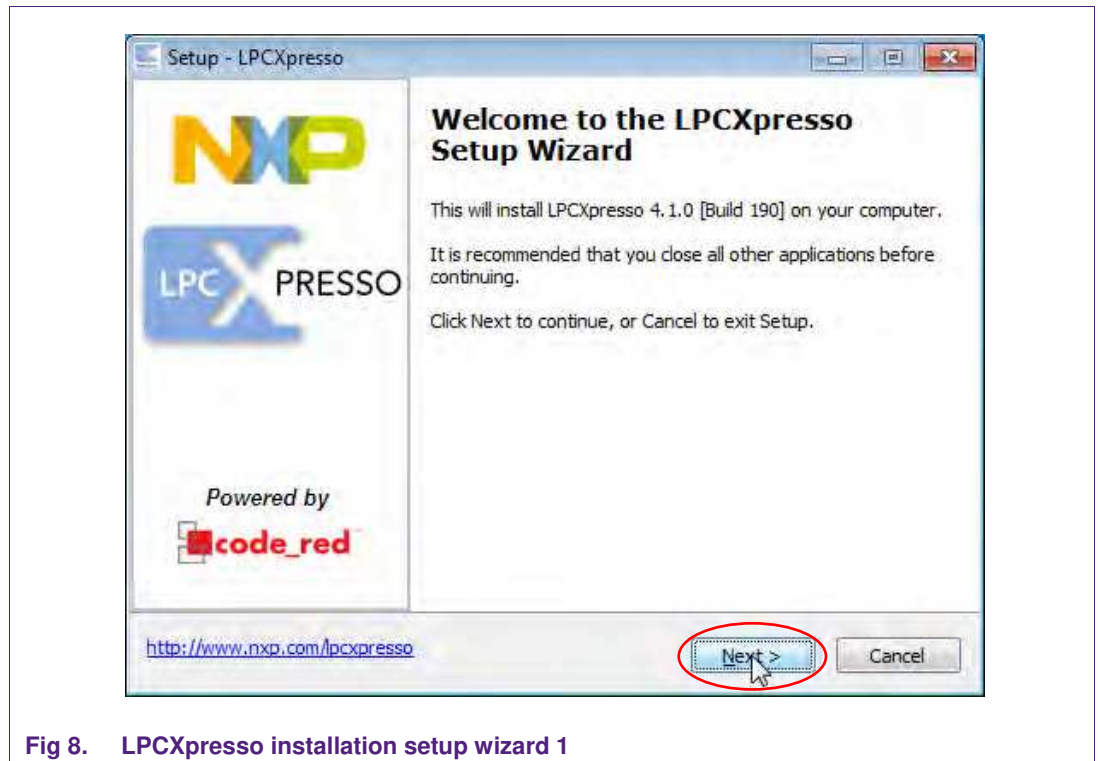
**For development please use the LPCXpresso version 4.x. Newer versions of the IDE are known not to work correctly with the provided software examples.**



This tool can be freely 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.

#### 4.1 Installation of LPCXpresso IDE

The IDE is installed into a single directory, of one's choice. Multiple versions can be installed simultaneously without any issues. The installation starts after double-clicking the installer file. Then click "next" on the setup wizard.



Read the license agreement then click next.

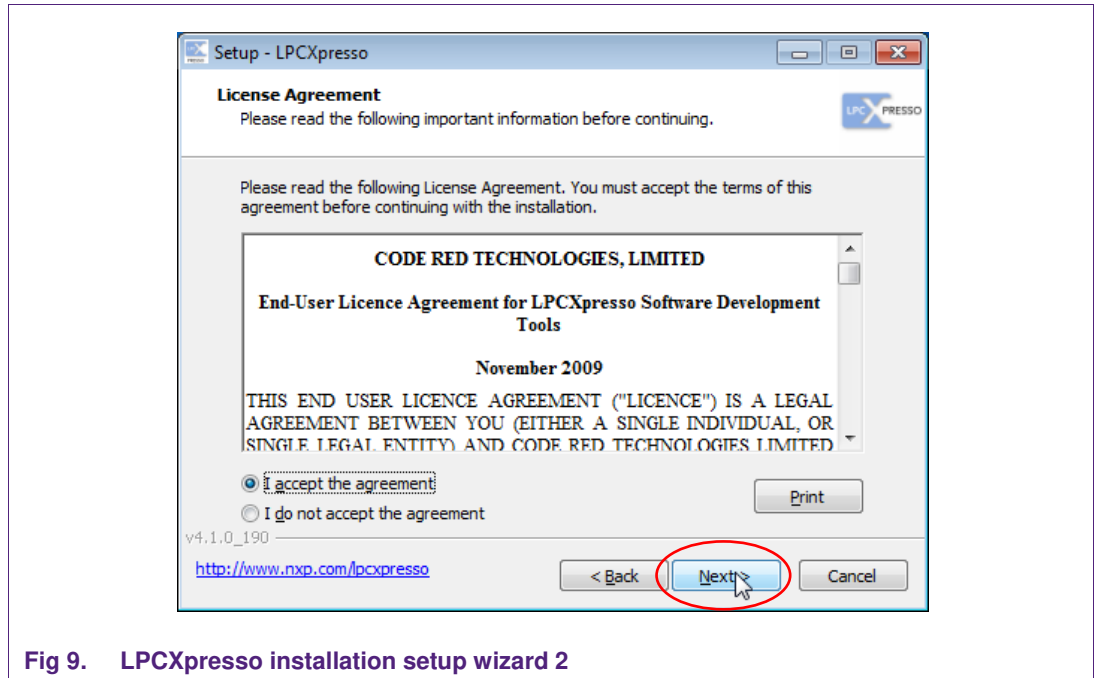


Fig 9. LPCXpresso installation setup wizard 2

There are numbers of other screens on the setup wizard but generally the default options can be accepted. After installation, an information file will be displayed. Click “Next” to accomplish the installation.

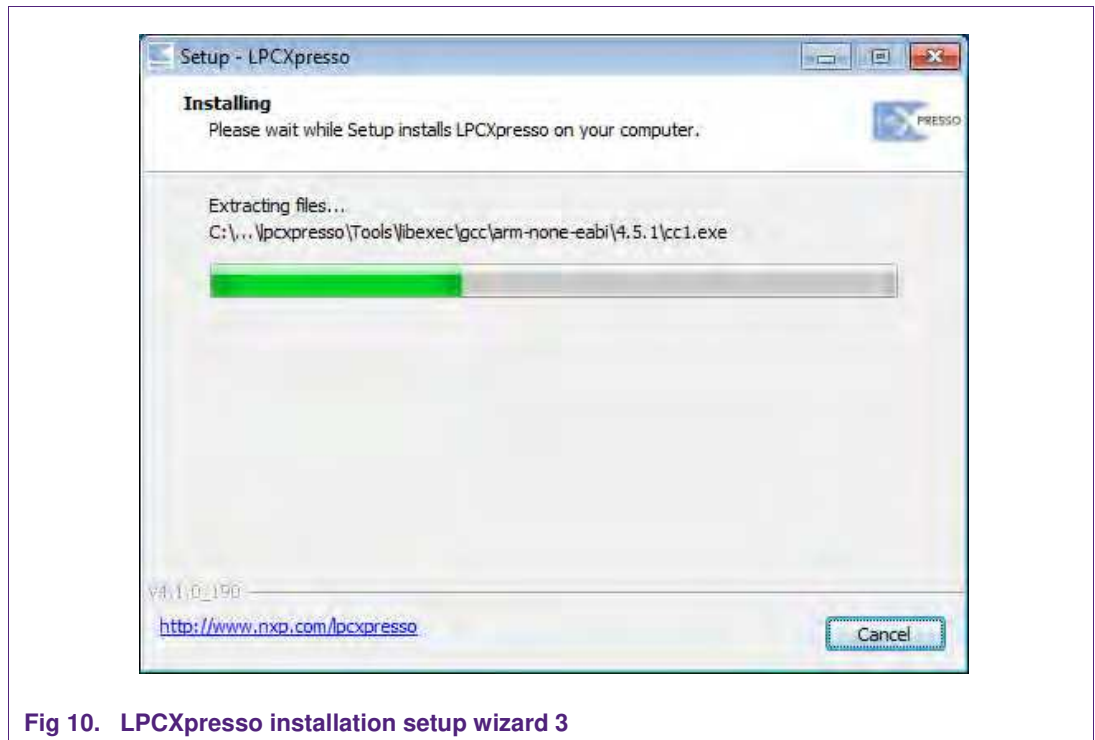


Fig 10. LPCXpresso installation setup wizard 3

After this installation step one will be asked if he wants to install some required drivers. The installation of these drivers should be accepted.

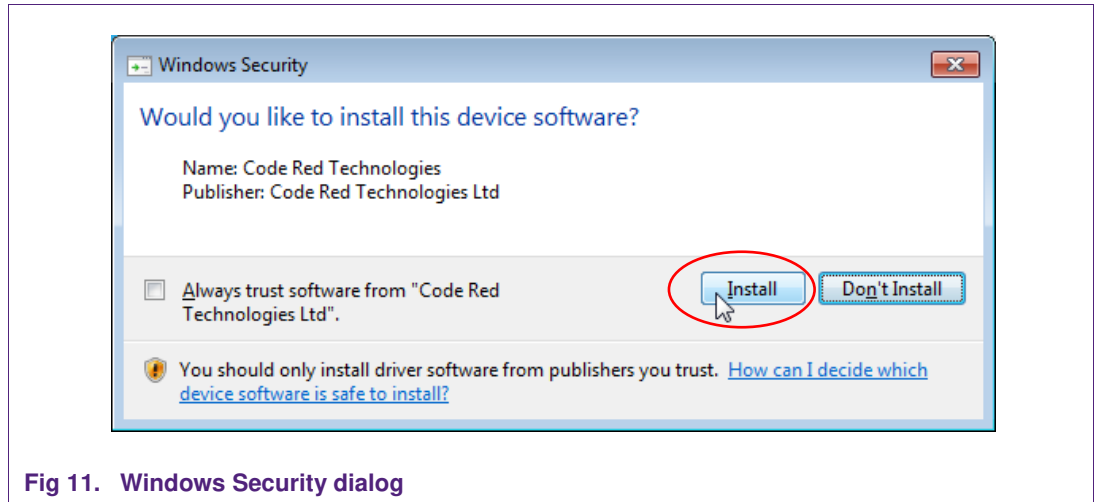


Fig 11. Windows Security dialog

After the setup wizard has finished one can launch the newly installed IDE.



Fig 12. LPCXpresso installation setup wizard 4

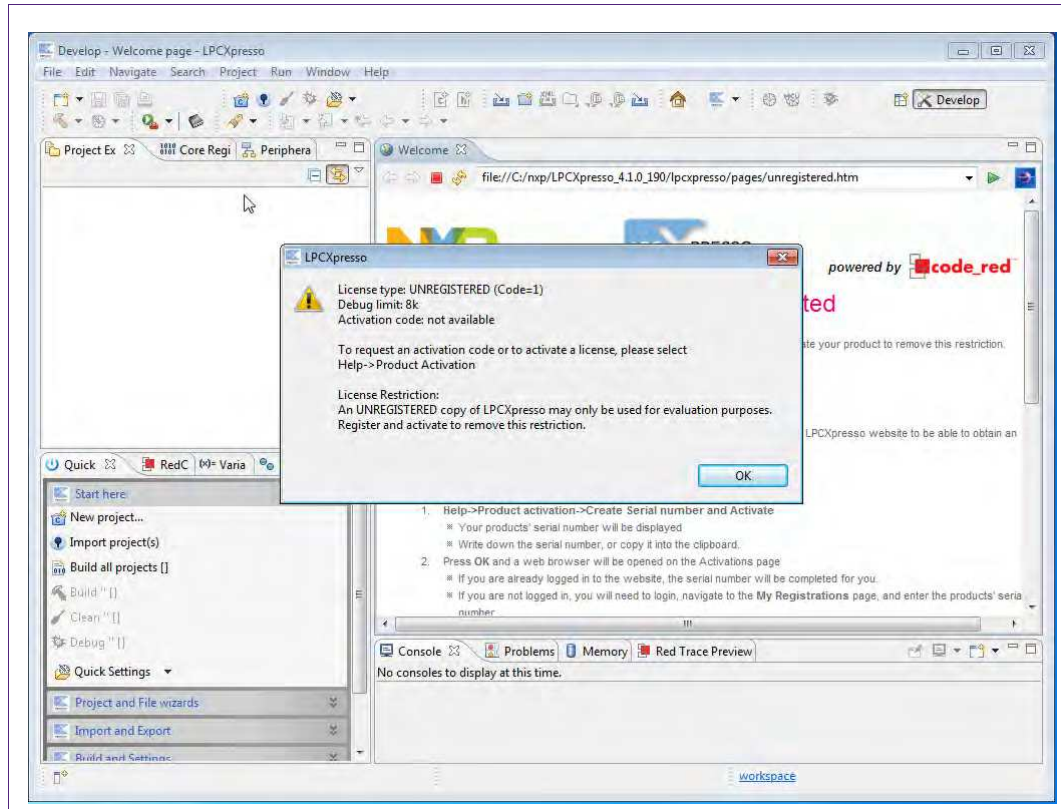


Fig 13. LPCXpresso IDE

Directly after the first start of the Eclipse IDE one will see an info dialogue that this is only an unregistered copy of LPCXpresso IDE. Just confirm the dialog and follow the instructions on the Welcome Screen to get a registered version without the debug limit of 8k. The registration is free and is needed to navigate to the website of Code Red. The Link is shown in the menu, Help → Product activation → Create Serial number and Activate...

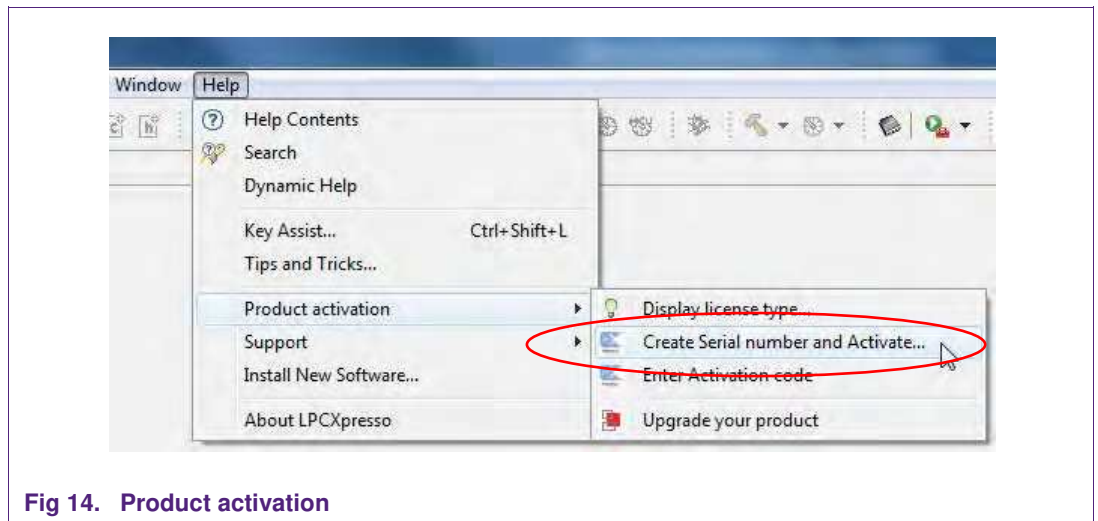


Fig 14. Product activation

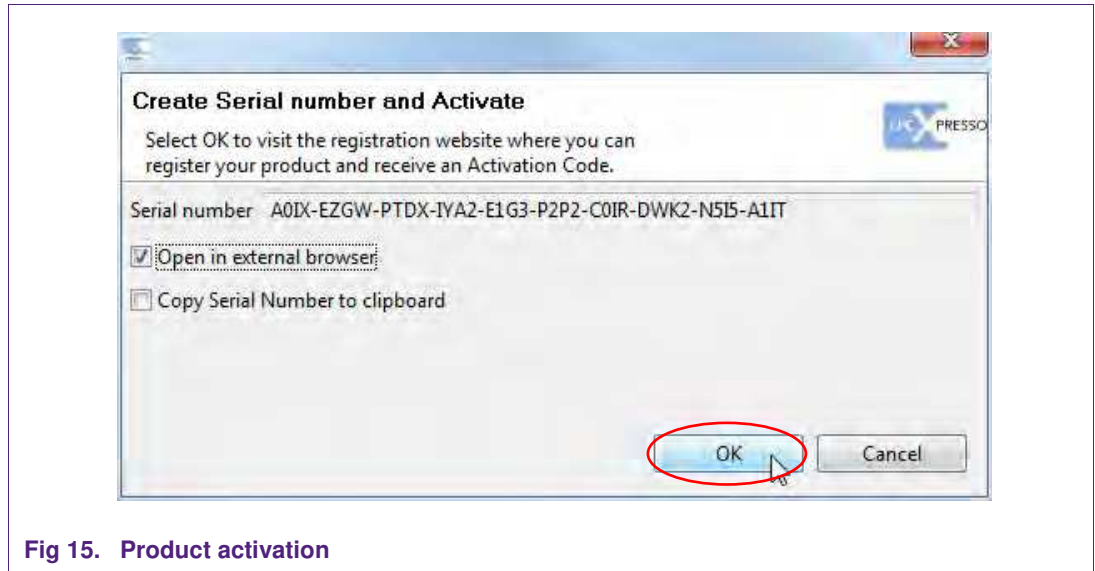


Fig 15. Product activation

If one doesn't already have an account at Code Red, please sign up to get an activation code. The code will be sent to the provided e-mail address.

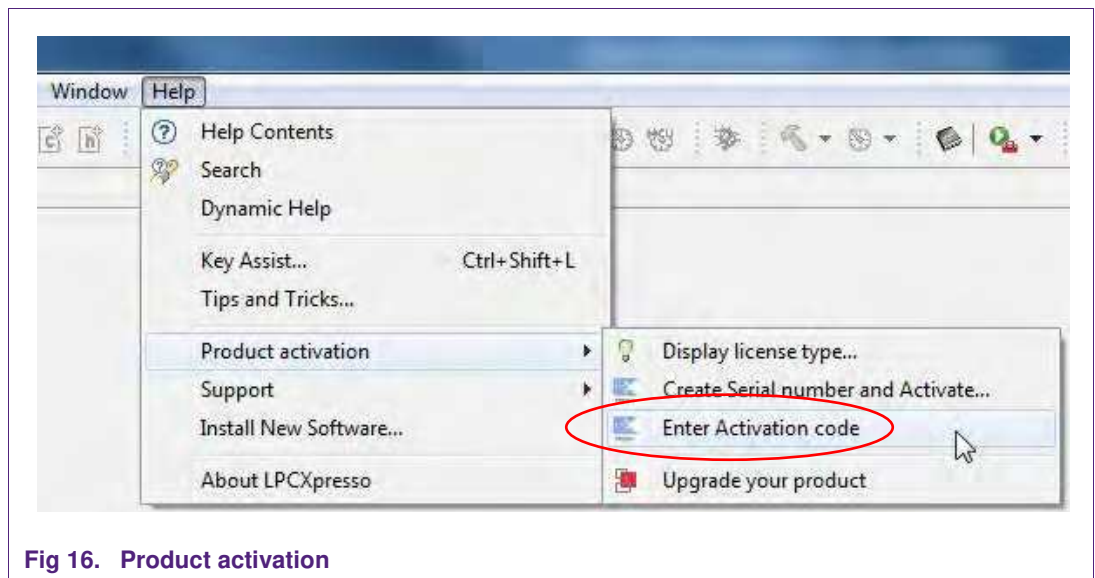


Fig 16. Product activation

Once the activation code arrives please open the activation window by pointing to Help → Product activation → Enter Activation code, and enter the code.

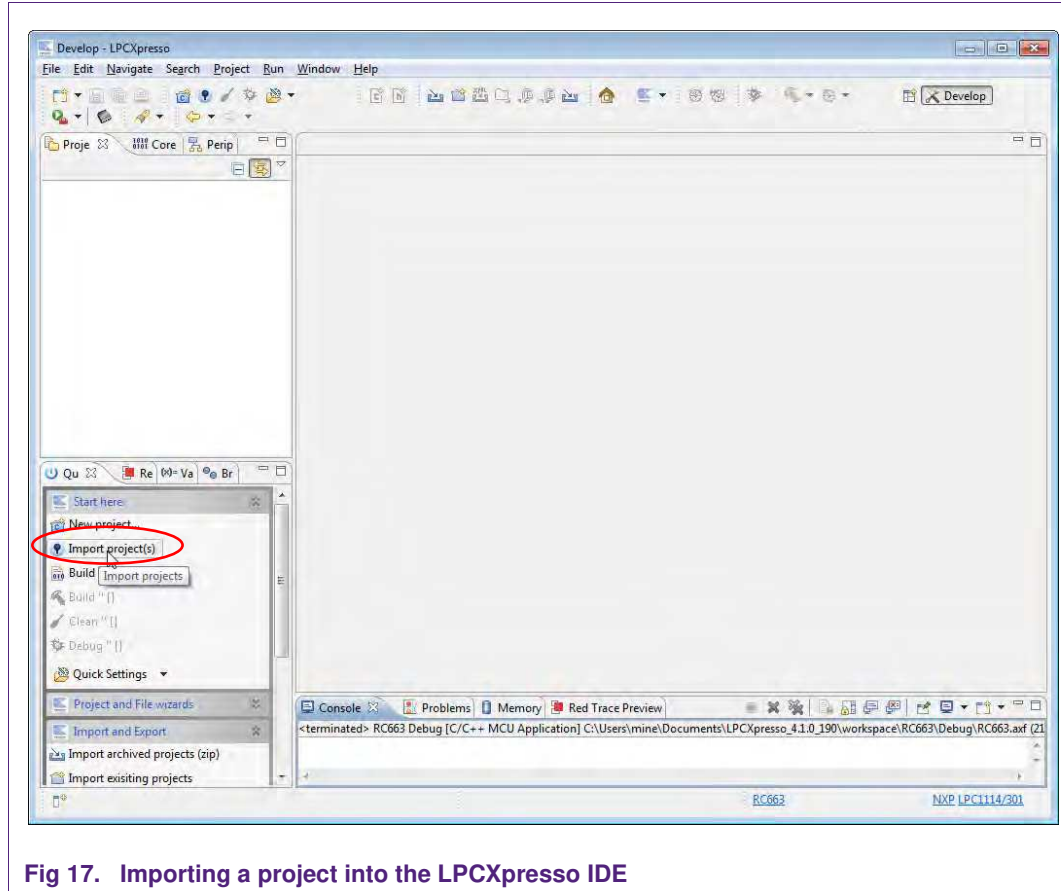
The success of the product activation will be confirmed by an info dialogue.

## 4.2 Extraction of the demo reader project

Once the LPCXpresso™ IDE has been installed on a computer, the sequence of installing the reference reader project is indicated:

- Start the LPCXpresso™ IDE.
- 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 17. Importing a project into the LPCXpresso IDE**

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

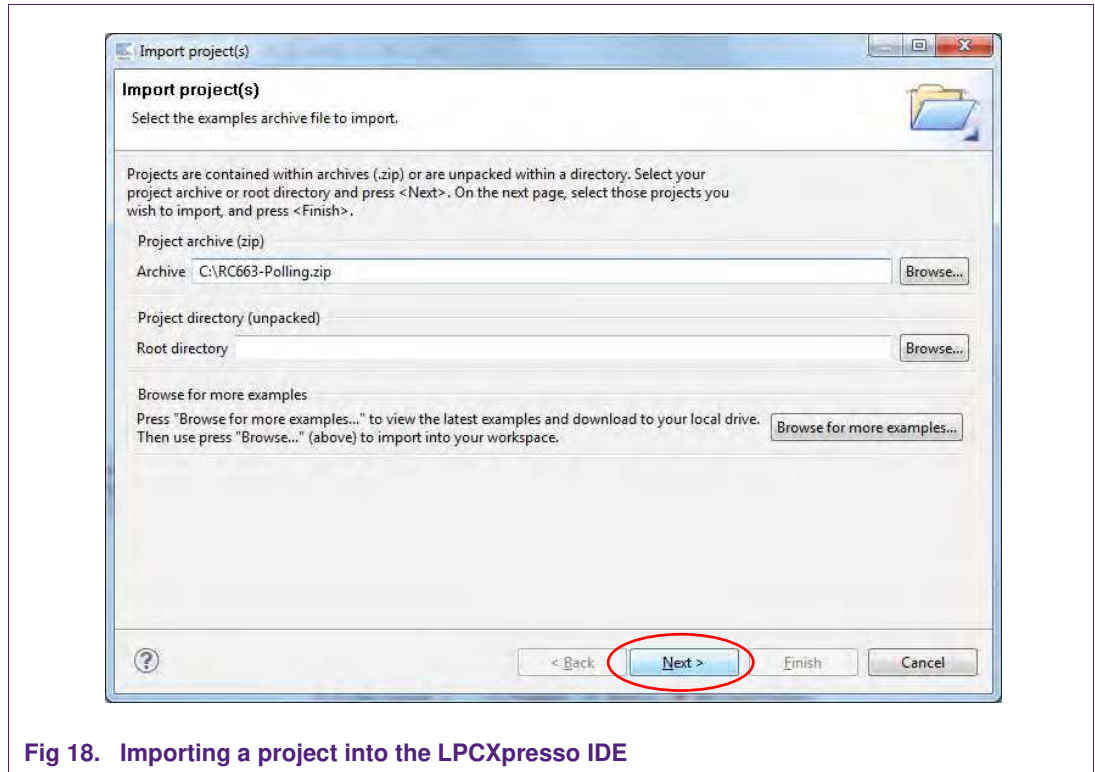


Fig 18. Importing a project into the LPCXpresso IDE

Browse the desired project and click “Next”.

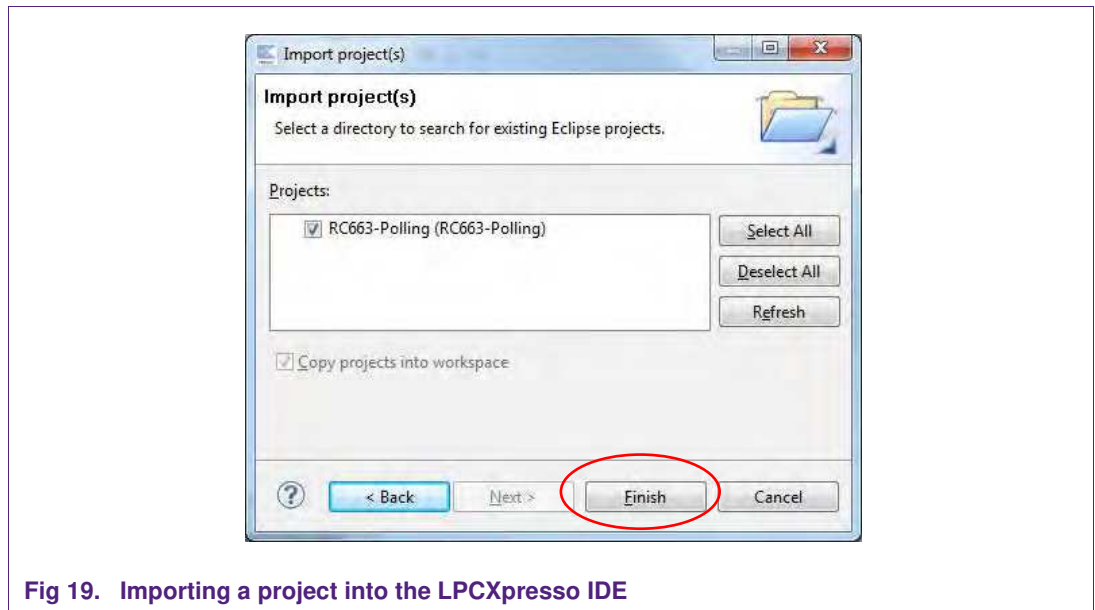


Fig 19. Importing a project into the LPCXpresso IDE

When the import process has finished one can start browsing the code. Most interesting might be the main.c which is located in ../src/main.c in the project.

Before one can run the project, the PREV601 board with the PRH601 needs to be connected to the computer. Wait until the adequate drivers have been installed.

### 4.3 Start the project

One can quickly start the reader project by editing the main function in the module **main.c**. This function first performs the hardware initializations of the LPC1227 and the RF transceiver RC663.

Detailed descriptions of the code in the form of comments have been provided in the **main.c** file. This should provide a detailed overview of how to initialize certain components and get data out of and onto the card in the RF field.

#### 4.3.1 Run the project

Before running the project, please ensure that the LPCXpresso with the Blueboard is connected to the computer.

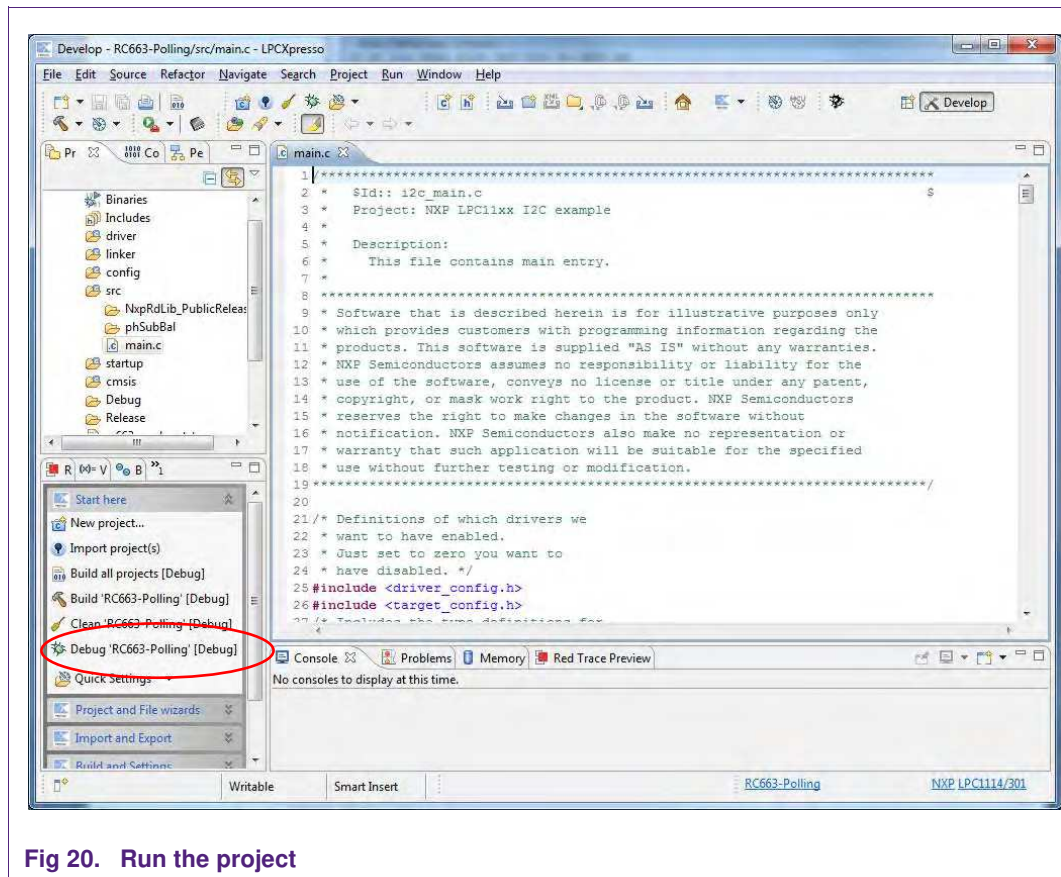


Fig 20. Run the project

Choose the desired project and click the Debug Button on the left hand side as shown in the example picture.



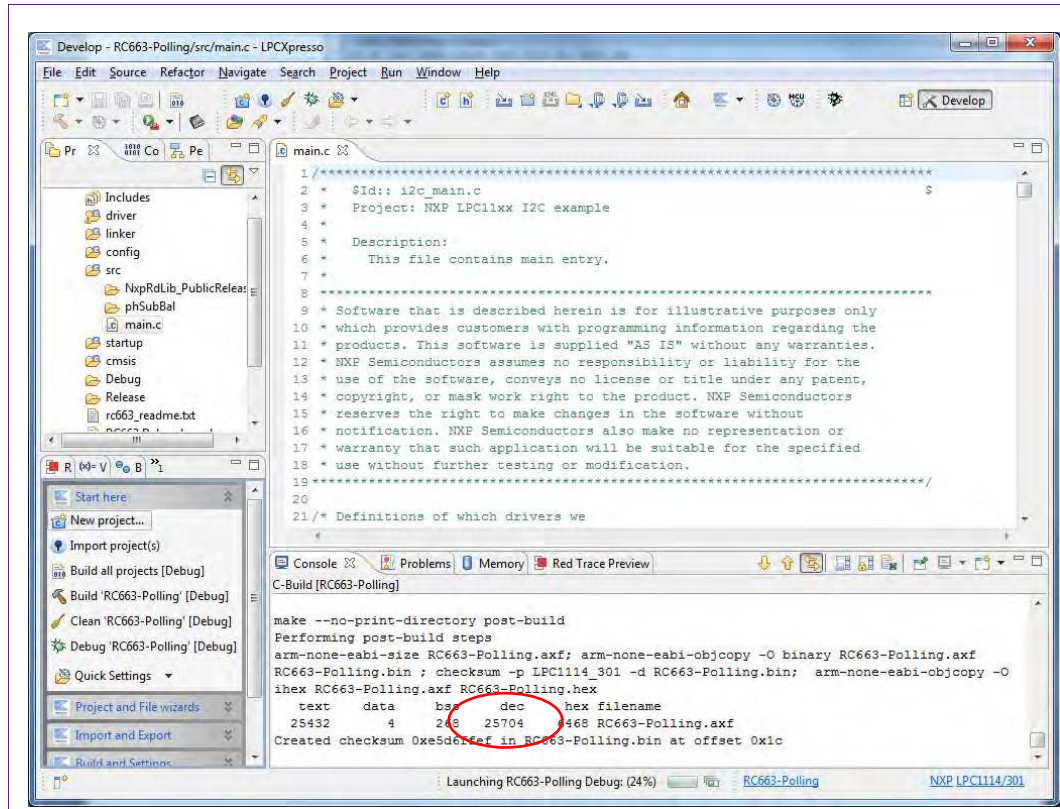


Fig 21. Run the project

After the build process one can see the size of the image in the console window.

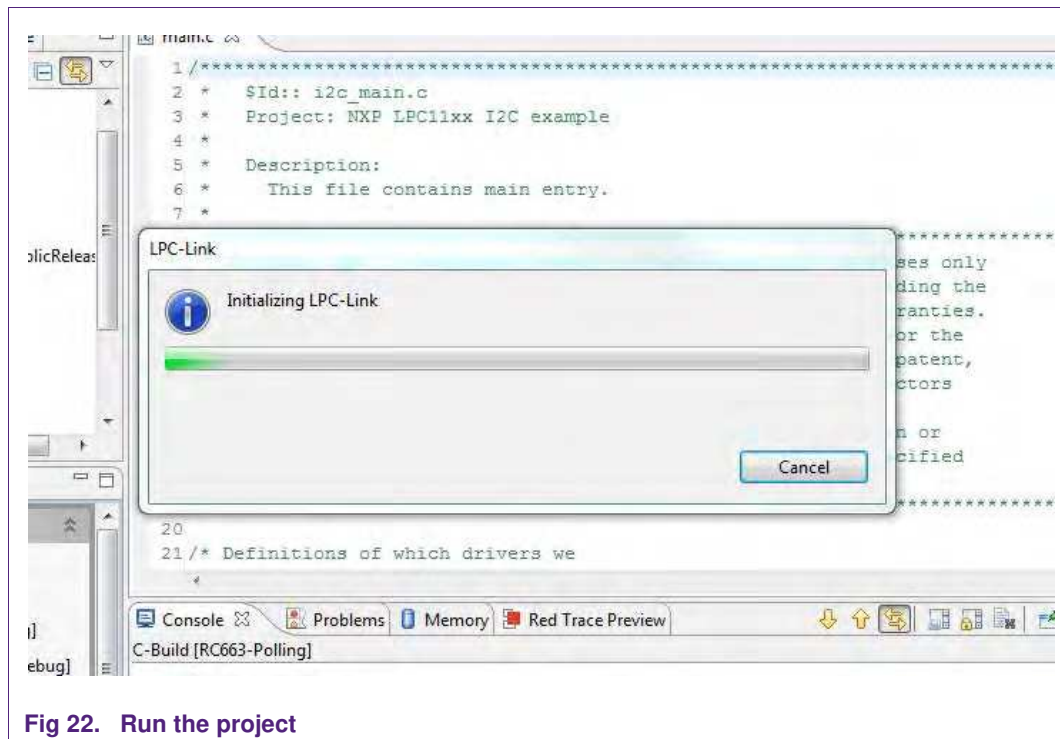


Fig 22. Run the project

The initialization of the LPC-Link can take a few seconds.

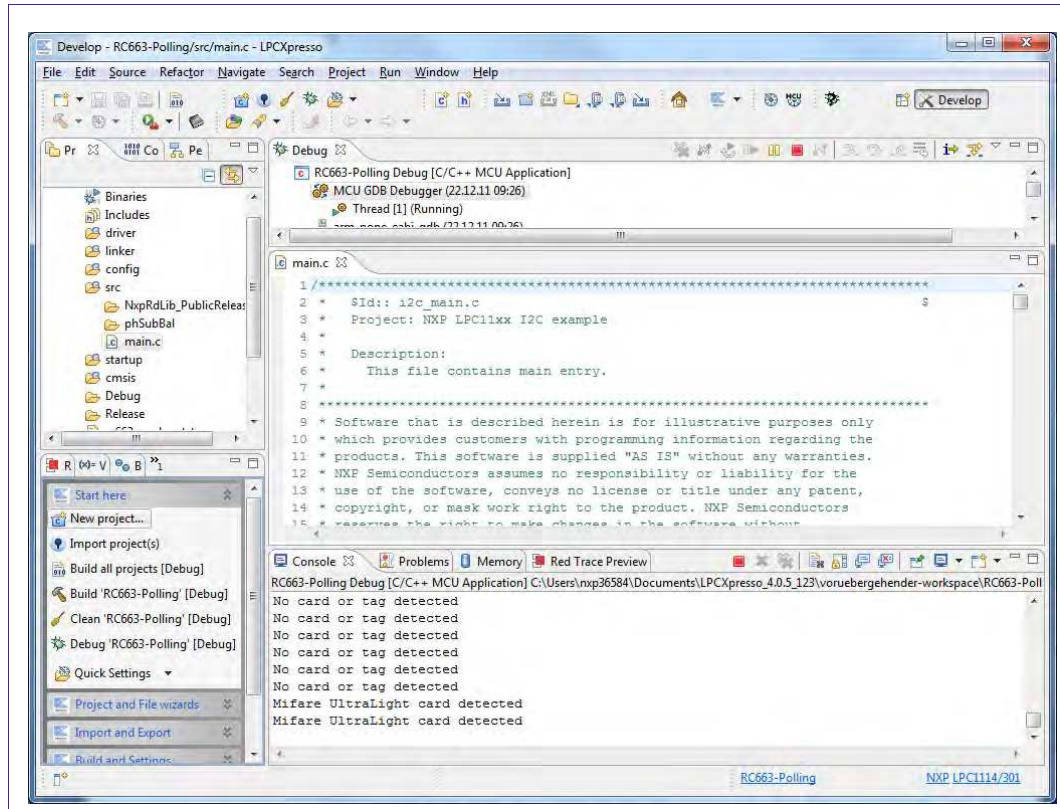


Fig 23. Run the project

After the software upload, the execution of the project starts immediately.

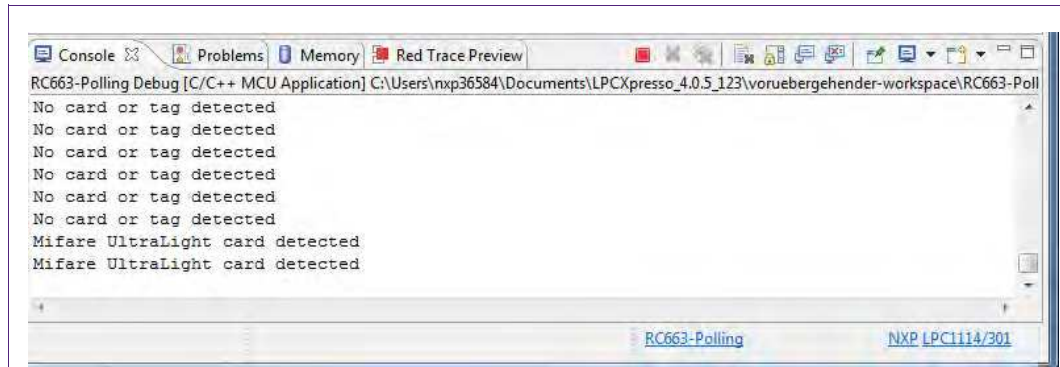


Fig 24. Run the project

In the console window at the bottom one will see the debug output of the execution.

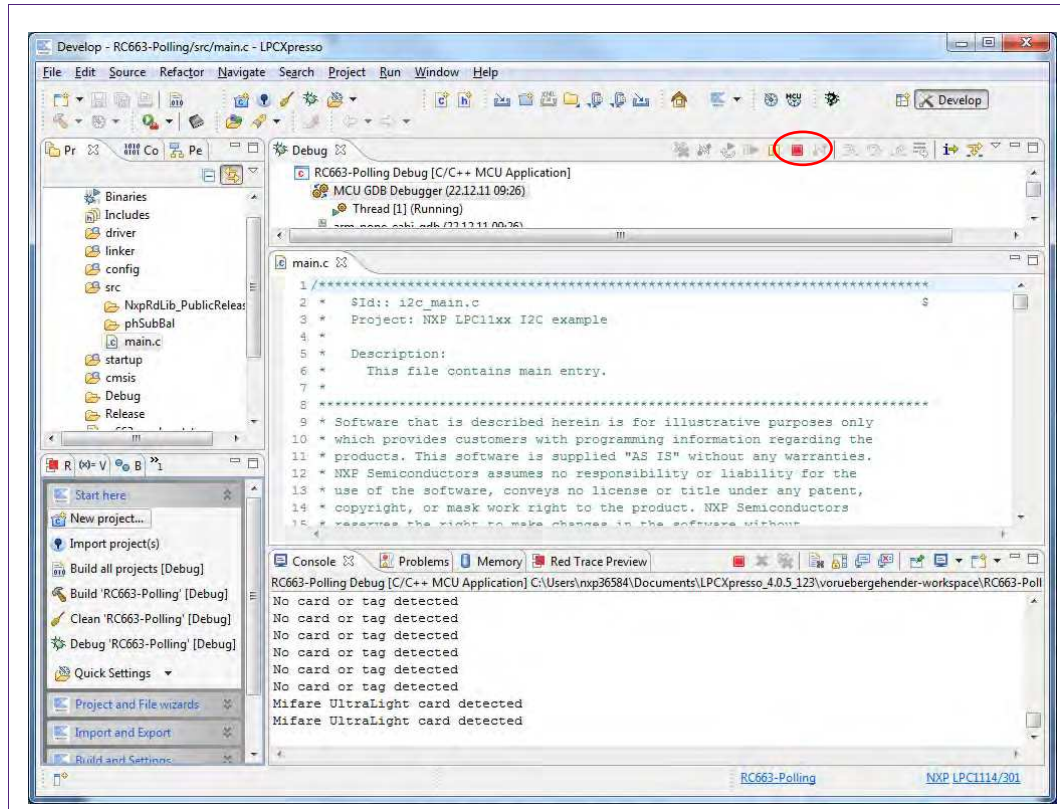


Fig 25. Stop the project

After the execution has reached the end of the main function please click the Terminate button to stop the execution. Otherwise one won't be able to rerun the project.

One can now do the following with the buttons near the top of the "Debug" view:







	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 26. Debug Buttons

## 5. Associated Projects

### 5.1 Communication with MIFARE Ultralight

Based on examples the MIFARE Ultralight project shows how read-write access can be achieved on this type of card.

If one uses a card which is not write protected or secured the example program writes a valid NDEF message onto the card. One can read this message with any NFC enabled mobile phone which can read NDEF messages.

### 5.2 Communication with MIFARE Classic

Based on examples this project shows how read-write access can be achieved on this type of card.

### 5.3 Communication with MIFARE DESFire

Based on examples this project shows how read access can be achieved on this type of card.

### 5.4 Polling

Based on examples this project shows how to initiate a basic communication with the following cards:

- MIFARE Ultralight
- MIFARE Classic
- MIFARE Plus
- MIFARE DESFire
- FeliCa compliant cards
- ISO/IEC 14443-B cards
- ISO/IEC 15693 Tags

This example project also looks for cards in range of the RF field in a continuous loop and returns the type of the detected card or tag to the console window.

### 5.5 Peer to Peer functionality

Based on examples this project shows an implementation of Peer to Peer (P2P) functionality.

Because the P2P implementation is still in alpha phase, one will find very rudimentary support for the protocol.

At the moment the project supports the communication via the SPI protocol and runs on the LPCXpresso LPC1227 development board.

#### 5.5.1 Installation

After downloading and unpacking the zip file, please run the installer. The installer just copies the LPCXpresso project files to the file system. After the installation has finished, please run the included batch file located in the installation directory

```
...\NxpRdLibP2PExtensions-x.x.x\NxpRdLib_P2PExtensions\ex\Rc663_Lpc12xx_P2P_Demo
```

After the batch file has been executed successfully, please start the LPCXpresso IDE and import the project. Browse to the root of the installation directory.

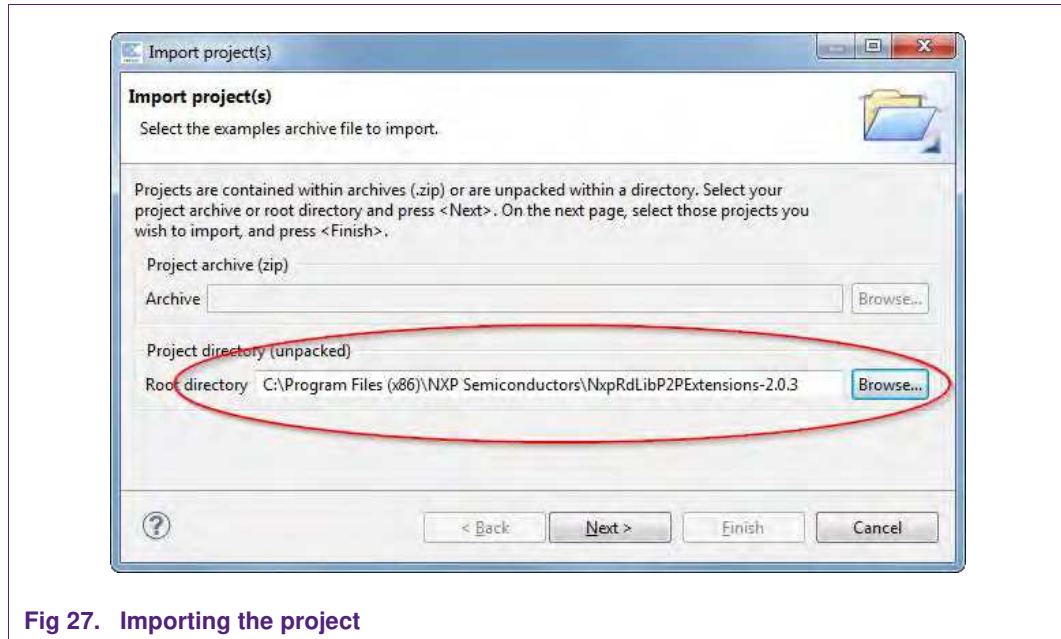


Fig 27. Importing the project

After the import there are two new projects in the workspace - one for the use with the RC663 Blueboard or PREV601 board and one for the use with the PN512 board. Basically these two projects provide the same functionality.

### 5.5.2 SNEP client project

In this project the PREV601 board behaves like SNEP client with performing the Put request functionality only [14][14]. On the other side there must be a device – peer capable of NFC communication and providing the SNEP server service. These conditions are fulfilled by a mobile device with Android platform (4.0 or later).

Here is a short instruction list how to use the SNEP client software:

1. Run the SNEP client project as described in section 4.3.1 with respect to LPC1227 microcontroller restriction and hardware configuration in section 6.4.
2. Hold your NFC device (smartphone) approximately 5 centimeters above antenna of the PREV601 board.
3. Once the NFC detected in the RF field of the transmission of the NDEF message is started. You may be notified by a sound or vibration of your device. The transmission may take several seconds depending on the size of the NDEF message. Hold your NFC device in range of PREV601 board’s field during entire transmission.
4. After transmission completed there should be the transmitted image displayed in the device’s screen.

#### 5.5.2.1 What is going on inside?

The SNEP client sends a hardcoded NDEF message encapsulated in a SNEP message to the NFC device. Thanks to the SNEP client project implements the P2P Reader Library Extension compliance with the LLCP and ISO18092 protocols on the PNEV512-Board’s side are ensured, implicitly a correct SNEP client-server communication.

Execution of the SNEP client software can be summarized in following steps:

1. Hardware initialization

2. Detecting the RF field for an NFC peer of the tag type F.

The software checks the RF field whether there is tag type F capable of performing the P2P communication

3. Once such device is found, the LLC link is activated in compliance with the procedure defined by the NFC forum [15].
4. LLCP socket creation and establishing connection with other peer – SNEP server.
5. Transmission of a given image file to the SNEP server:

The SNEP client sends an initial fragment 128 bytes long. Then it waits for a response from the server. Because in SNEP header it is declared longer SNEP message than one fragment, the server should response with the Continue response. The SNEP client can go on with sending the rest of the SNEP message. As soon as the entire SNEP message has been transmitted, the SNEP client shall receive the SNEP Success response from the mobile device and the transmitted picture should be immediately displayed on the mobile's screen.

The SNEP client software is deeply described in [13] in “Sample code”.

### 5.5.2.2 Choosing the NDEF message

By default the software sends image of the NXP logo as NDEF message. There are more NDEF messages prepared in dedicated header files (see Table 1). Only a single header can be compiled with the SNEP client application. To choose another content of the NDEF message for transmission just follow the instructions below. In case of URI or text message just skip from step 1 to step 4 (steps 2 and 3 are for selection of an image). Concurrently with instructions there are parts of source code to demonstrate choosing of hardcoded PNG image file for transmission.

1. Open for editing the source file *ndef\_message.c* located in folder *src/*.
2. Uncomment the line with content a header file to be transmitted. Let all the other lines commented.

```
/* select the required type of transported data */
// #include <c_tabletxt.h>
#include <c_tablepng.h>
// #include <c_tablepng_NXP_QR.h>
// #include <c_tablejpg_NXP_logo.h>
```

3. In Table 1 in the same row as the chosen file look up two identifiers corresponding to the chosen file.
4. In the array *nmess[]* uncomment the line with the couple of identifiers corresponding to the type of the chosen file. Comment all the other lines.
5. Save the changes and recompile the project.

```
NDEF_messages n_mess[]={ /* type, parameter, string */
//   {NDEF_TYPE_IMAGE,NDEF_IMAGE_JPEG,c_table, sizeof(c_table)},
  {NDEF_TYPE_IMAGE,NDEF_IMAGE_PNG,c_table, sizeof(c_table)},
//   {NDEF_TYPE_IMAGE,NDEF_IMAGE_TIFF,c_table, sizeof(c_table)},
//   {'T', LANG_NO, text1, sizeof(text1)},
//   {'T', LANG_EN, c_table, sizeof(c_table)},
//   {'T', LANG_EN, text1, sizeof(text1)},
```

**Table 1. Table of files prepared for NDEF message transmission**

Identifiers from the last two columns are necessary for choosing the right line from `n_mess[]`.

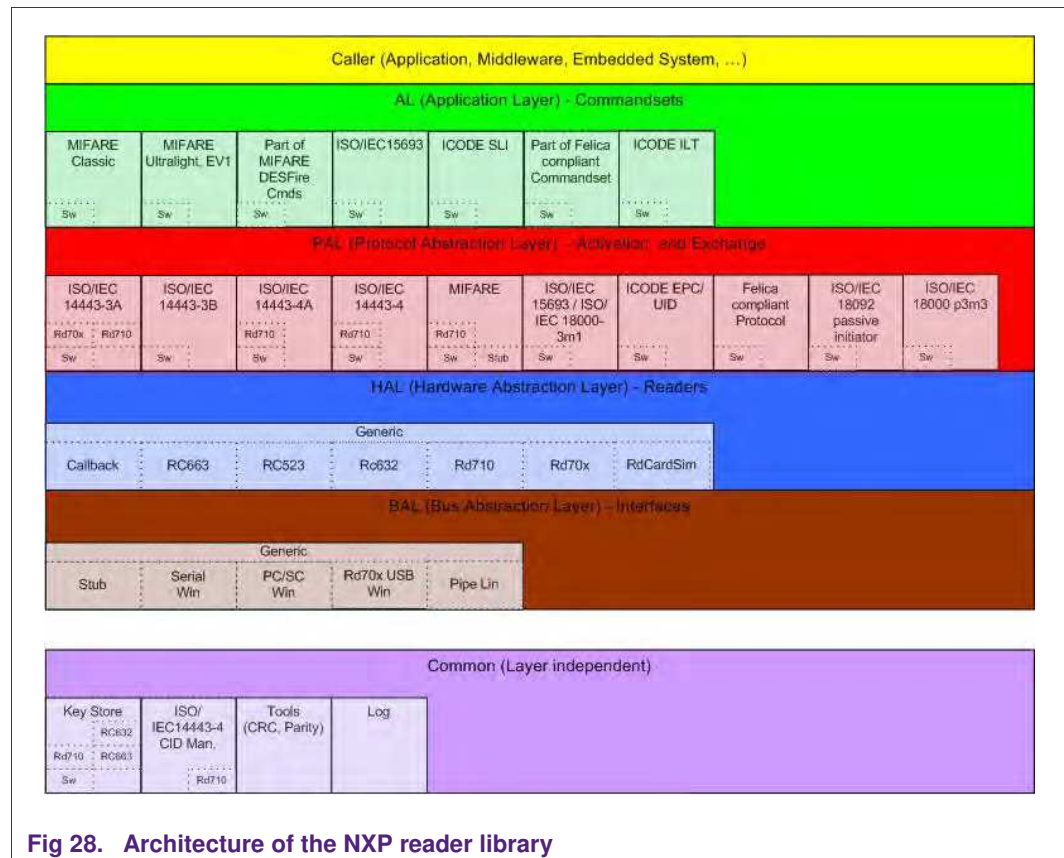
Content	Header Name	NDEF message identifier	File type identifier
PNG image	<code>c_tablepng.h</code>	NDEF_TYPE_IMAGE	NDEF_IMAGE_PNG
QR code of NXP	<code>c_tableQR.h</code>	NDEF_TYPE_IMAGE	NDEF_IMAGE_PNG
Image of NXP logo	<code>c_tablenxp.h</code>	NDEF_TYPE_IMAGE	NDEF_IMAGE_JPEG
Long text message	<code>c_tabletxt.t</code>	'T'	LANG_EN

## 6. Supplementary Notes

### 6.1 Software architecture

The software of the reference reader is based on the NXP reader library [5]. 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 in 4 layers:

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



**Fig 28. Architecture of the NXP reader library**

### 6.1.1 Bus abstraction layer

This layer offers functions to abstract the hardware parts of the LPC1227 microcontroller.

These functions use the specific libraries available for the LPC1227 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 LPC1227 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 RC663.

### 6.1.3 Protocol abstraction layer

Every PAL function is a low level function realizing a single functionality. It is encapsulated in a module. Every function 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,
- ISO/IEC15693,
- FeliCa,
- NFC Initiator

### 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 from the previous three parts into high level functionalities.

For more details on the NXP reader library, the reader is invited to refer to the document RC663 Software Design Guide of the NXPRDLib [\[6\]](#).

## 6.2 Build configuration

This project comprises 2 build configurations:

- 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.

Note, that only in Release Configuration one can flash the software onto the Blueboard and start it automatically, once power has been attached to the board.

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 Debug build (or Release build).

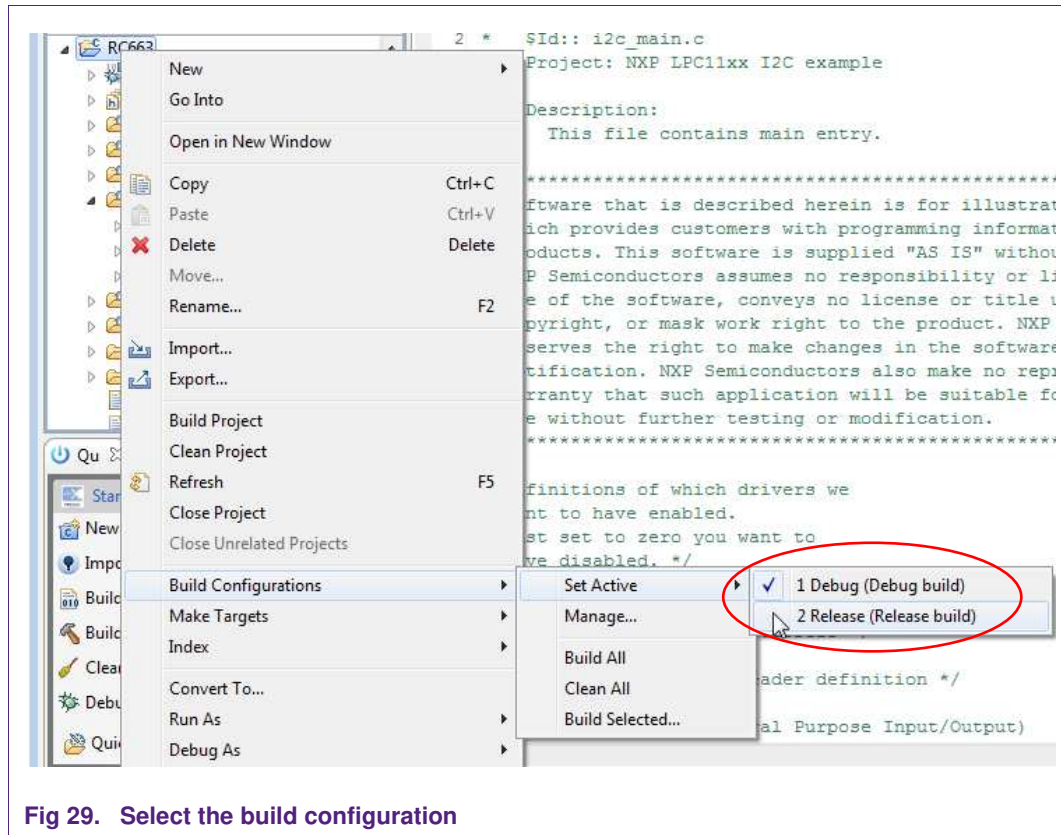


Fig 29. Select the build configuration

### 6.3 Level of compiler optimization

When the code size at the current compiler level overloads the FLASH size of the target board (128K for the ARM-based microcontroller LPC1227), 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 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.

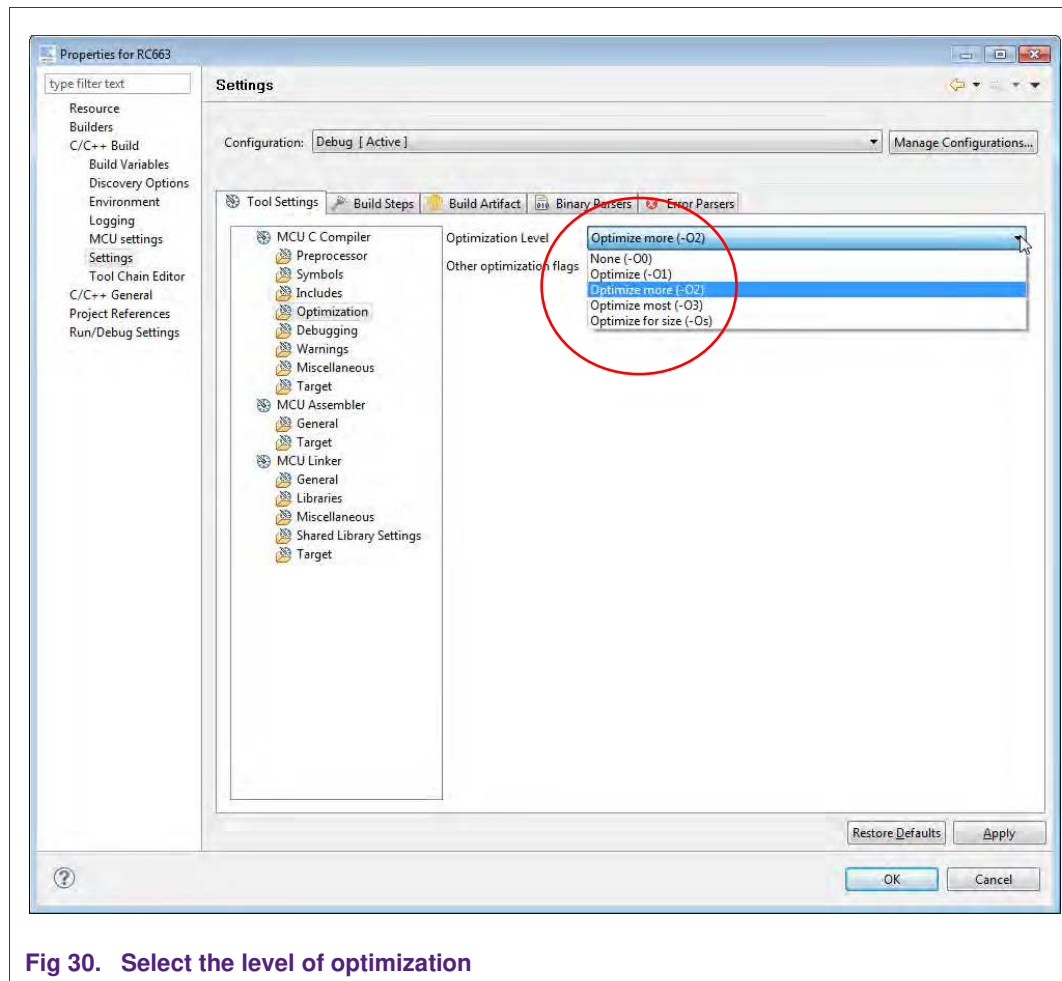


Fig 30. Select the level of optimization

### 6.3.1 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. Since optimization can make such a big difference in code size and performance, it is a good idea to test the project with code optimization on, and plan for a final build that is optimized.