AN11211

Quick Start Up Guide RC663 Blueboard

Rev. 1.8 — 21 July 2014
226218

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	RC663, Blueboard, LPCXpresso, MCU, Code Red, eclipse, LPC1769, LPC1227, NFC Reader Library, CLEV663B
Abstract	This application note is related to the installation procedures of the RC663 Blueboard. It describes the actions to be done to become acquainted with the demo reader.

Revision history

Rev	Date	Description
1.8	20140721	Updates in the description regarding the TUSA board.
1.7	20140519	Removed the note about the version of the LPCXpresso IDE. Some small corrections. Changed the description and pictures of the projects Polling and, Classic Changed the description of the P2P description due to a software update. Removed the description about the projects Ultralight and DESFire. All projects are now based on the NFC Reader Library version 3.010. Therefore all projects have been refactored. Added support for the development board LPCXpresso LPC1769 which is based on an ARM Cortex M3 microcontroller. The NXP Reader Library is now called NFC Reader Library
1.6	20131110	Added a note about the LPCXpresso IDE version in chapter 0
1.5	20130613	Added description about the P2P Snep Client
1.4	20130221	Added description of the P2P project. Added description of the I²C configuration for the Blueboard version 3.0 and above. Added information about the use of the projects in conjunction with the LPC1227 MCU. Added information about the documentation of the NFC Reader Library. Added information about the exemplary project of code size optimization of the NFC Reader Library.
1.3	20120913	Small corrections of the TUSA description
1.2	20120822	Insertion of the description for the 3rd party “Tusa” Board
1.1	20120704	Small text corrections
1.0	20120604	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

1. Introduction

This application note gives a detailed overview of the hardware for working with the RC663 contactless reader - we use the LPCXpresso LPC1769 and the Blueboard (**Chapter** 2) – the installation procedures of the Development Environment (**Chapter** 4.1) and the handling of the reader projects using the NFC Reader Library (**Chapter** 4.2).

Detailed information on the RC663 in connection with the NFC Reader Library can be obtained at [1]. But with the information provided in this document one will be fine to get started.

The projects used in this documentation are:

- Communication with MIFARE Classic → **Chapter** 5
- Polling for Tags in the RF - field → **Chapter** 5.2
- Exemplary Peer to Peer Initiator implementation (**LPC1769 only**) → **Chapter** 5.3

2. Hardware overview of the Demo Reader

The demo reader is made up of 2 separate boards:

- A CLEV663B demo board (called Blueboard) provided by NXP. This board has connectors which are designed to exactly fit the ones of the companion, the LPCXpresso LPC 1227 and the LPCXpresso LPC 1769 development boards.
- A commercial LPCXpresso LPC 1769 development board (**12NC:** 935291912598, **Type:** OM13000+598) which can be provided by NXP or bought directly on the market. See Ref. [11].

Once the two boards are joined via the connectors, the demo reader is ready for use.

2.1 RC663 demo board (Blueboard)

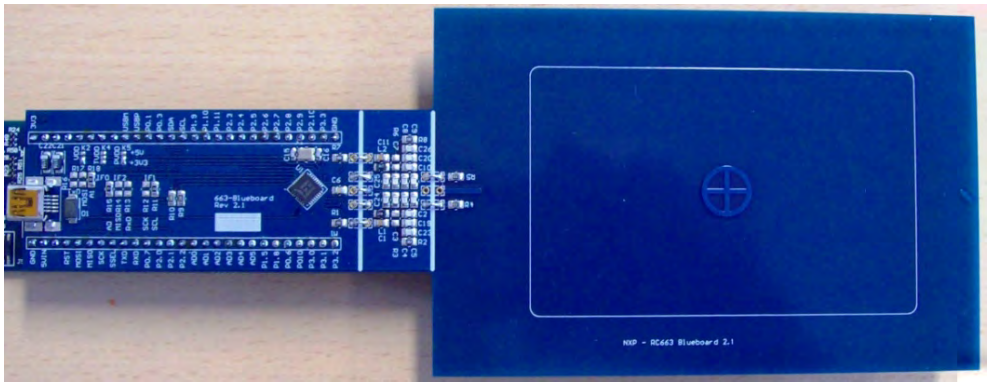


Fig 1. Picture of RC663 demo board

The RC663 demo board embeds the contactless communication transceiver IC RC663 with all its elements needed for transmission: EMC filter, matching network and the antenna. The RC663 supports different kind of contactless communication methods and protocols at 13.56 MHz:

- Reader/Writer mode supporting ISO/IEC14443A/MIFARE,
- Reader/Writer mode supporting ISO/IEC14443B,
- Reader/Writer mode supporting FeliCa scheme,
- Passive initiator mode according to NFCIP-1
- Reader/writer supporting ISO/IEC 15693,
- Refer to the data sheet of this IC [3] for more details

Thanks to the relevant solder bridge, the host link of the RC663 demo board can be configured for:

- Serial UART,
- I^2C,
- SPI

The voltage of the power supply VDD, the pad supply PDD and the transmitter supply can also be configured independently to 3.3 V or 5 V using the solder bridges.

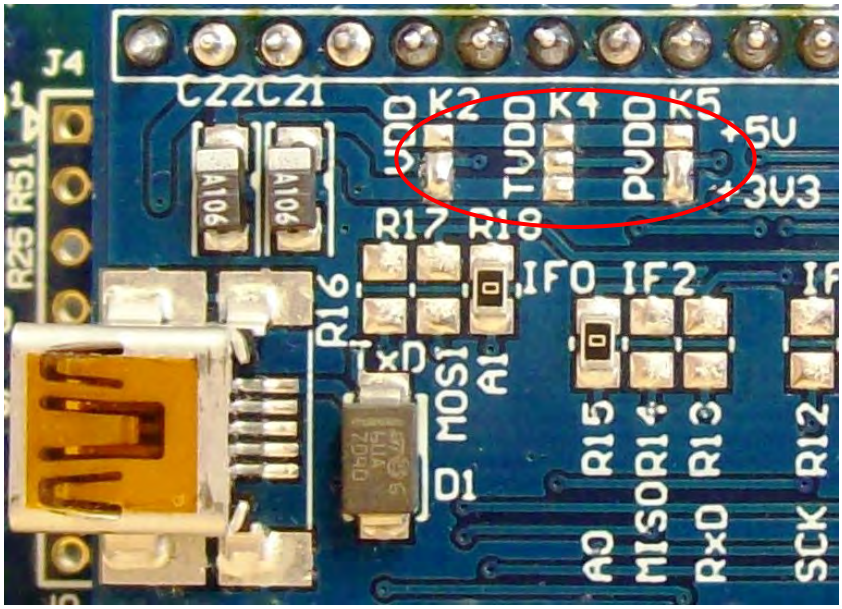


Fig 2. Picture of solder bridges

2.1.1 Derivates of the RC663 demo board (Blueboard)

To meet the interests of the market, we also offer Blueboards with a slightly modified RC663 on it.

The four versions are as follows:

- RC663 Blueboard

  This Blueboard with the RC663 offers the full functionality. It supports the modes described above. In the following text we always relate to the RC663 Blueboard. If not, this will be indicated by a note.

- SLRC610 Blueboard

  This Blueboard with the RC610 works with ISO/IEC 15693 tags only. To use this Blueboard, one will have to manually change one parameter in the NFC Reader Library.

  In the folder ../intfs/ open the file “phhalHw_Rc663_Reg.h” and scroll to the line

```
1    #define PHHAL_HW_RC663_CMD_LOADPROTOCOL         0x0DU
```

  It should be around line 641.

  Change the value from `0x0DU` to `0x03U`.

  Now, one should be able to use the SLRC610 Blueboard.

- MFRC630 Blueboard

  This Blueboard with the MFRC630 works with ISO/IEC 14443A cards only. No modification in the NFC Reader Library is needed.

- MFRC631 Blueboard

  This Blueboard with the MFRC631 works with ISO/IEC 14443A and ISO/IEC 14443B cards only. No modification in the NFC Reader Library is needed.

If one has one of the limited Blueboards, one will have code in some of the example projects that will not work with the very board. There is no need to adapt the code for the

derivate (except for the one described above). If one, for example, executes code for ICODE tags and has a MFRC631 Blueboard, this code will have no effect at all. It will behave as if there was no tag in the field.

2.2 CE certification of the Blueboard

The current version of the Blueboard (v.3.0) is CE (European Conformity) compliant.

2.3 LPCXpresso LPC1769 development board

To work with the provided projects, one will also need an LPCXpresso LPC development board. Such a board is **not included** in the Blueboard hardware package.

The LPC1769 development board integrates an NXP ARM Cortex-M3 microcontroller LPC1769 with 512 Kbytes of Flash memory and 64 Kbytes of RAM. It integrates a lot of hardware parts:

- Serial UART interface,
- SPI controller,
- I^2C controller,
- Serial Wire test/debug interface,
- For detailed information, see LPC1769 product site [4]

The LPCXpresso board contains a JTAG/ SWD debugger called the “LPC-Link” and a target MCU. LPC-Link is equipped with a 10-pin JTAG header and it seamlessly connects to the target via USB (the USB interface and other debug features are provided by NXP’s ARM9 based LPC3154 MCU).

Fig 3. Picture of LPCXpresso LPC1769 development board

2.4 Alternative to the LPCXpresso LPC1769

With the provided code one can use the LPCXpresso **LPC1227** (**12NC:** 935294603598, **Type:** OM13008+598) instead of the LPCXpresso LPC1769 without the need of any adaptation in the code. Compared to the LPCXpresso LPC1769 it has a smaller flash memory of 128KB instead of 512KB and a Cortex M0 processor. For instructions on how to change the project settings to work with the LPC1227 see the description in section 7.2.

2.5 Preparation of the hardware

The first step after unpacking the Blueboard and the LPCXpresso is soldering the connectors onto the boards in order to join them later. In our example we use a multipoint connector as one can see on the pictures below.

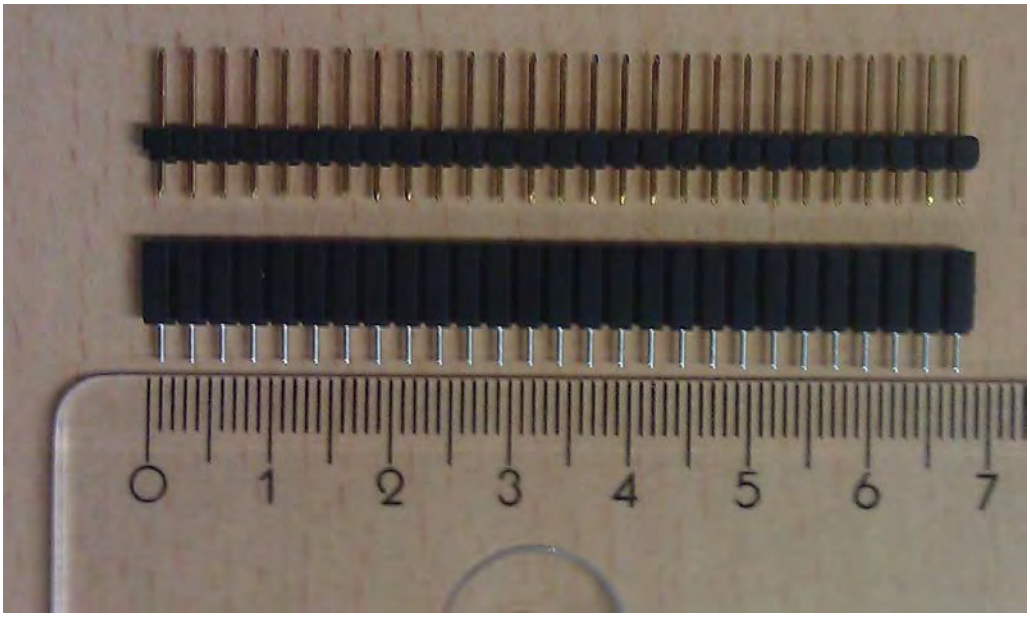


Fig 4. Multipoint Connectors we used

One may buy these connectors at any electronic store. Here are some examples [5].

After soldering the connectors connect the boards as shown on the following figures.

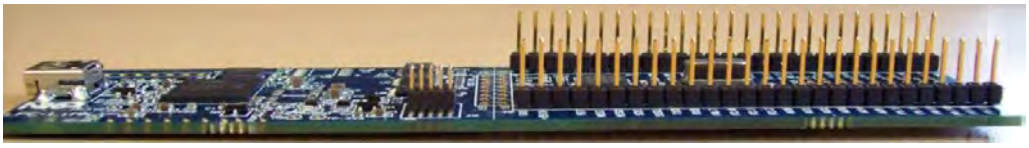


Fig 5. LPCXpresso with the Multipoint Connectors

Now the hardware is ready for use. Please connect the LPCXpresso board to the Blueboard.

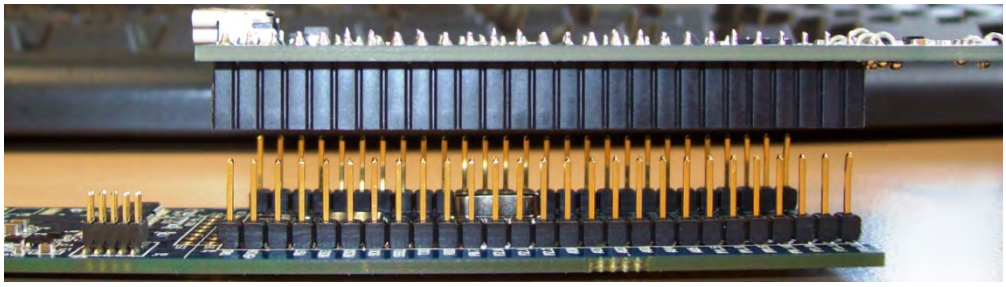


Fig 6. Connect the two boards

Be informed that there is the possibility of arranging the boards vice versa. The pictures below will illustrate how that is meant.

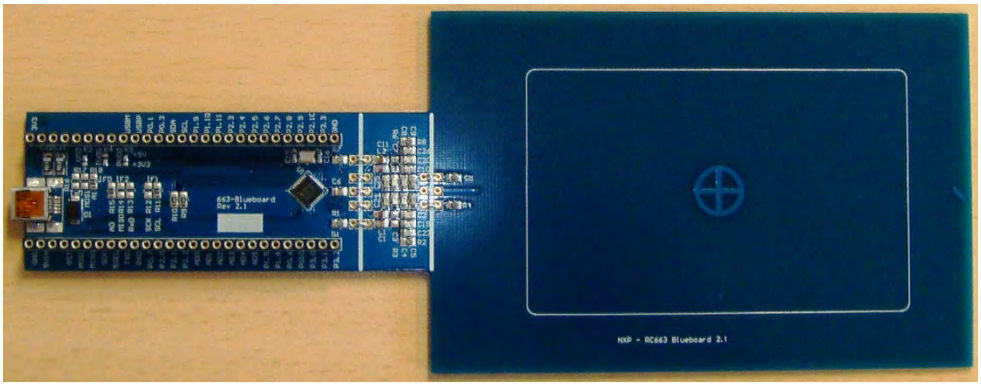


Fig 7. Picture of RC663 demo board with the connectors joined alternatively

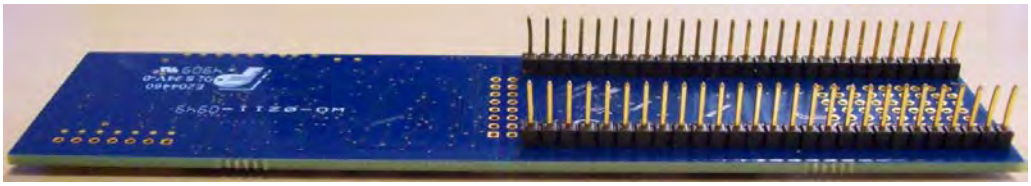


Fig 8. LPCXpresso with the Multipoint Connectors used in the alternative way

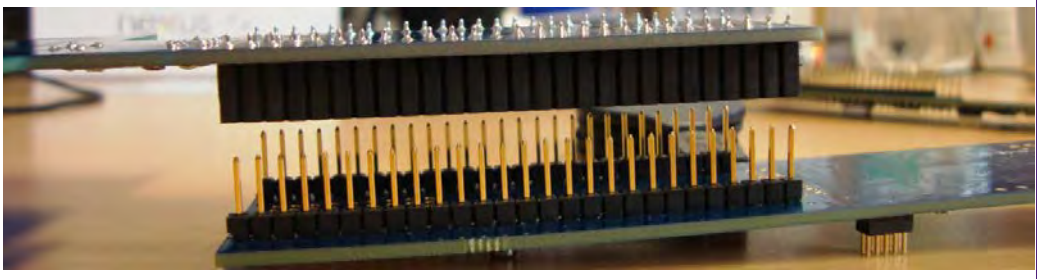


Fig 9. Connect the two boards the alternative way

2.6 Interesting points of measurement

Although the Blueboard is not designed to do extended measurements, there are some points of measurement one might be interested in.

To give some examples, a few of these points of measurement will be described in the following subsections.

2.6.1 RXP - receiver input pin for the received RF signal

RC663 – Pin 12

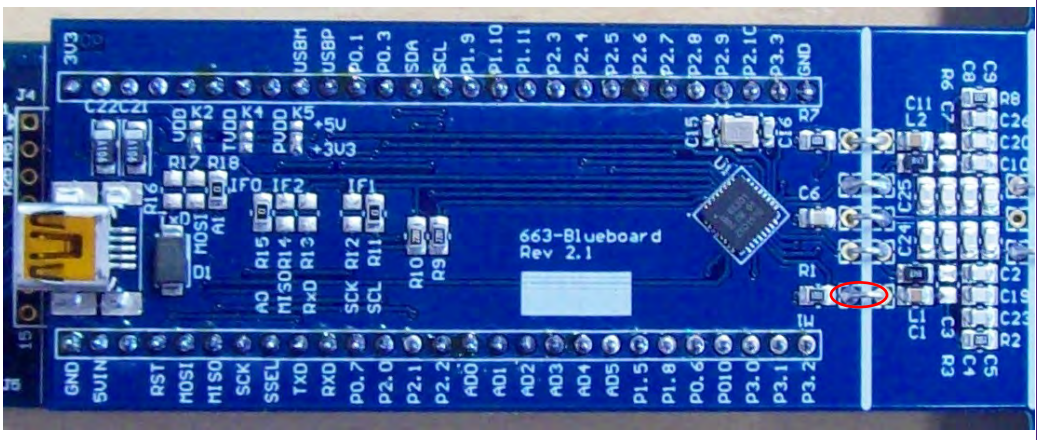


Fig 10. Pin RXP

2.6.2 RXN - receiver input pin for the received RF signal

RC663 – Pin 13

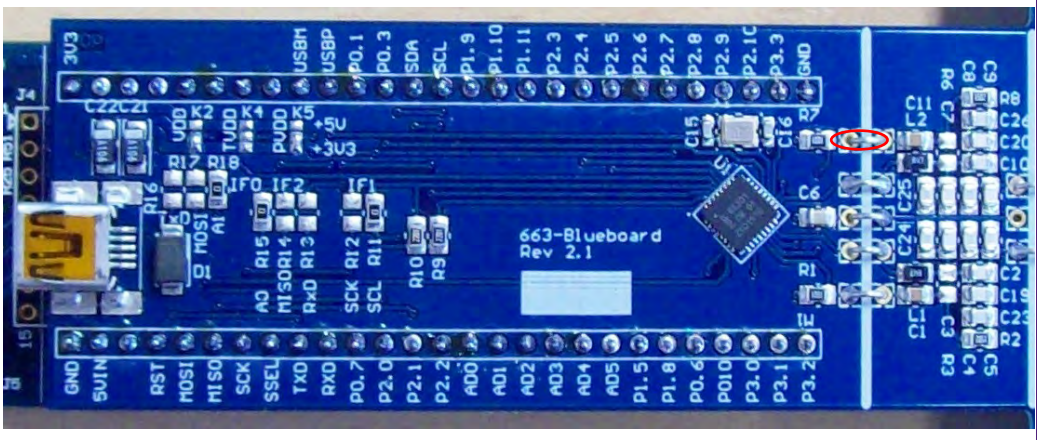


Fig 11. Pin RXN

2.6.3 TVDD - transmitter voltage supply

RC663 – Pin 18

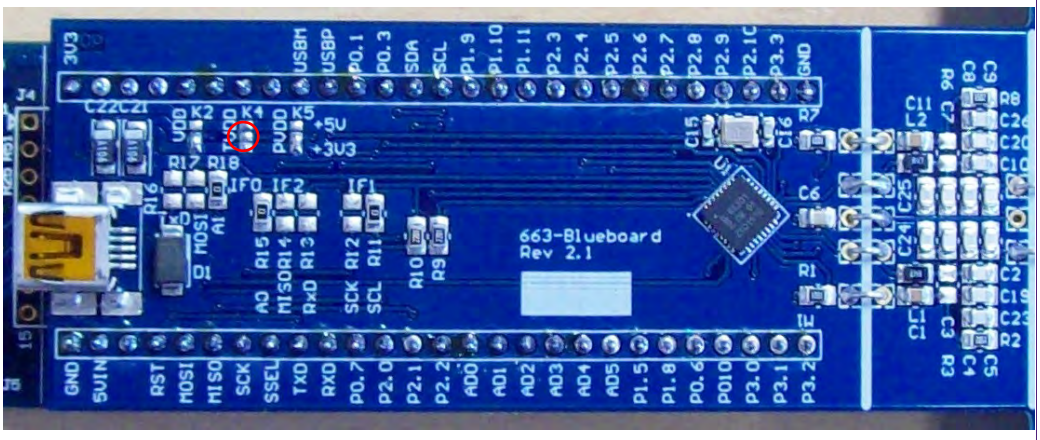


Fig 12. Pin TVDD

2.7 Preparing the Blueboard for the use with SPI or I²C

The Blueboard is generally delivered in I²C configuration, for this reason one only needs to change the configuration if the use of the board in SPI configuration is desired.

From Blueboard version 3.0 on the layout slightly changed. So we provide two different descriptions for changing the interface between I²C to SPI. Since version 3.0 the standard interface switched to the SPI interface.

2.7.1 Blueboard version 2.1 and below

There are exactly six solder bridges to change.

1. Open the bridge at R18
2. Open the bridge at R15
3. Open the bridge at R11
4. Close the bridge at R17
5. Close the bridge at R14
6. Close the bridge at R12

See the picture of the Blueboard in SPI configuration below.

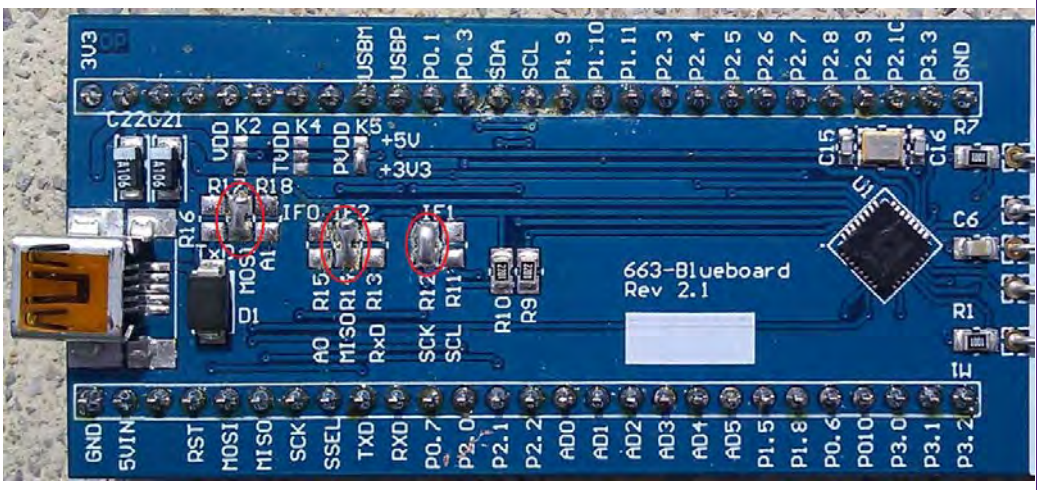


Fig 13. Blueboard in SPI configuration

To use the Blueboard in I²C configuration with the provided software projects, one has to execute two small adaptations in the code, which are described in section 7.6.

2.7.2 Blueboard version 3.0 and above

These boards are delivered in SPI configuration. To change that to I²C, one needs to change six solder bridges.

1. Open the bridge at R15
2. Open the bridge at R21
3. Open the bridge at R26
4. Close the bridge at R16
5. Close the bridge at R20
6. Close the bridge at R29

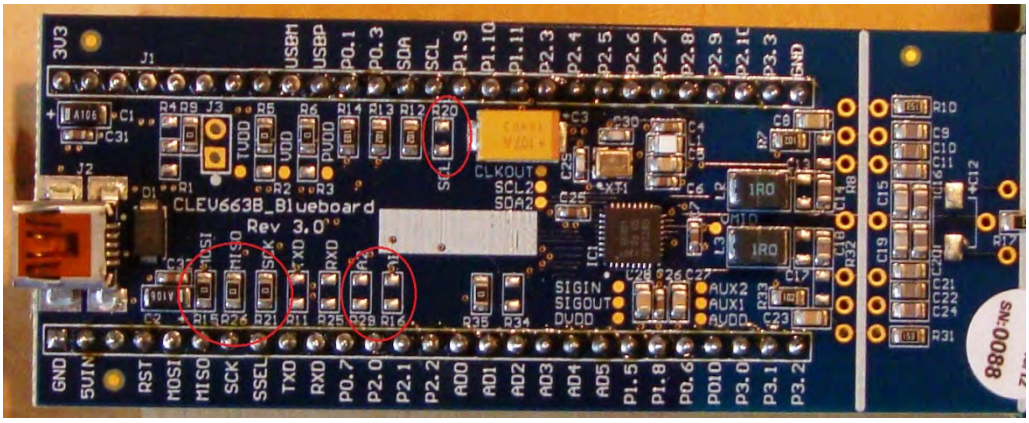


Fig 14. Relevant solder bridges for the SPI and I²C configuration of the Blueboard version 3.0

2.8 Other supported system architectures

The projects described in this guide are also available on Linux. The projects are preconfigured for the use on the Raspberry Pi with the Raspbian image. The SPI interface is used for the communication between the application and the NFC controller. The software and the start guide can be downloaded at the product page of the EXPLORE-NFC [13].

Although this guide only describes the use of the EXPLORE-NFC extension board, it also supports the PNEV512B and the CLEV663B Blueboards. These Blueboards can be used with a special adaptor called BluePi. For information about how to configure the hardware and the software please refer to section 7.8.

3. Installation of the LPCXpresso Board

The guidelines to install the reader are as follows:

- Connect the LPCXpresso Board as well as the Blueboard to a real USB2.0 port of the PC (for speed reasons) using the mini-USB connector. The PC detects and installs the Board automatically. The Blueboard only needs to connected to the PC out of power reasons.
- Once the Board is 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.

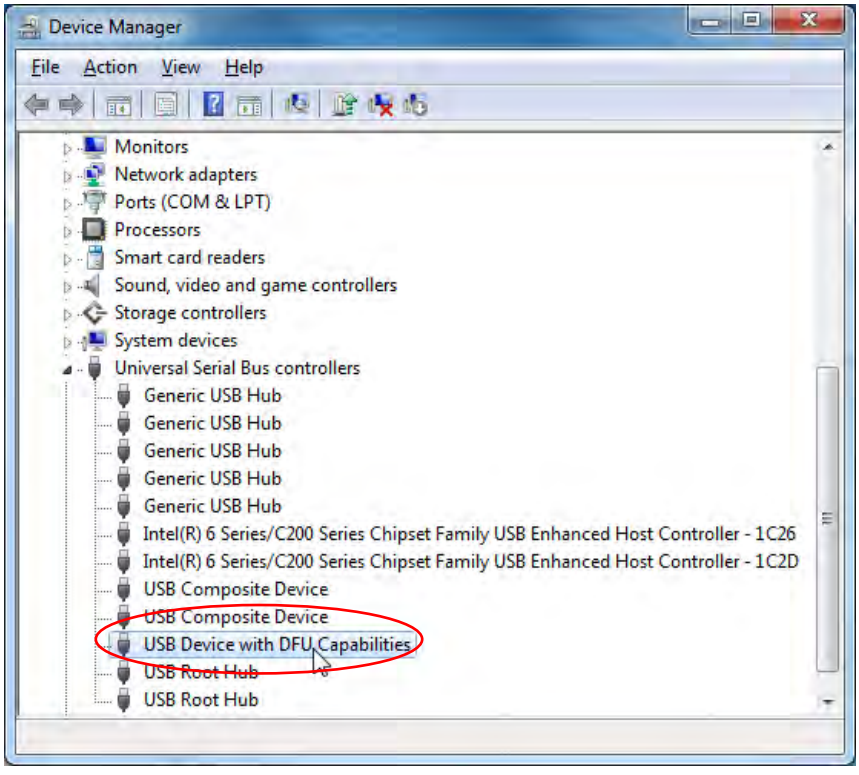


Fig 15. Enumeration of the LPCXpresso Board in Device Manager Window

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 is comprised of a simplified Eclipse-based IDE and low-cost target boards which include an attached JTAG debugger.

This tool can freely be downloaded from the LPCXpresso website [2]. 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.

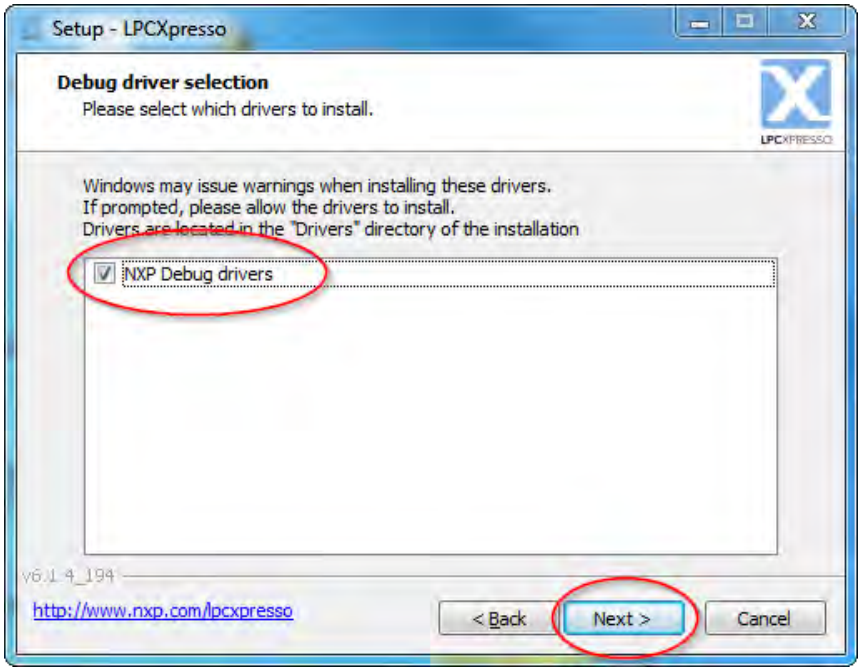


Fig 16. LPCXpresso installation NXP Debug drivers

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

During the installation, the user will be asked if he wants to install some required drivers. The installation of these drivers should be accepted.

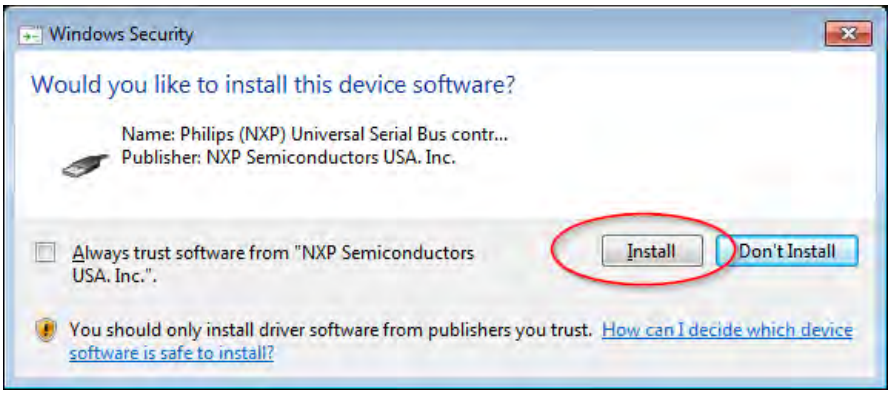


Fig 17. Windows Security dialog

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

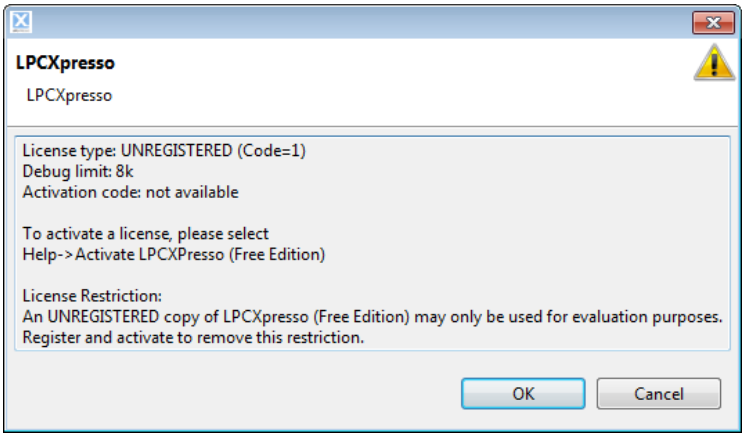


Fig 18. LPCXpresso IDE

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...

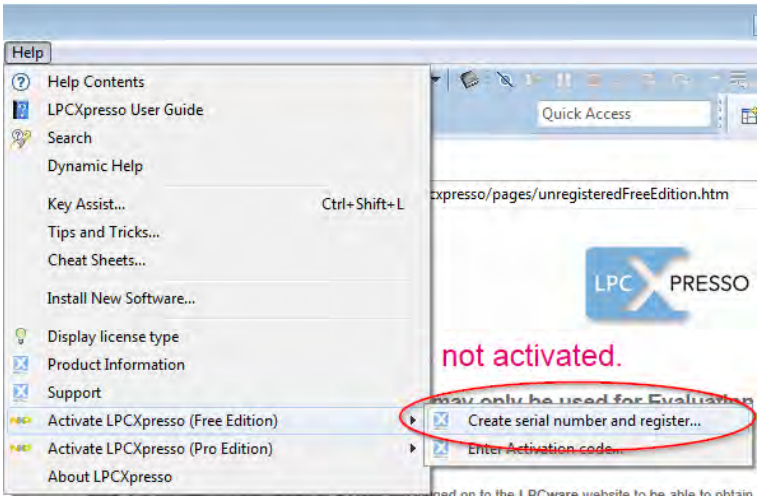


Fig 19. Product activation

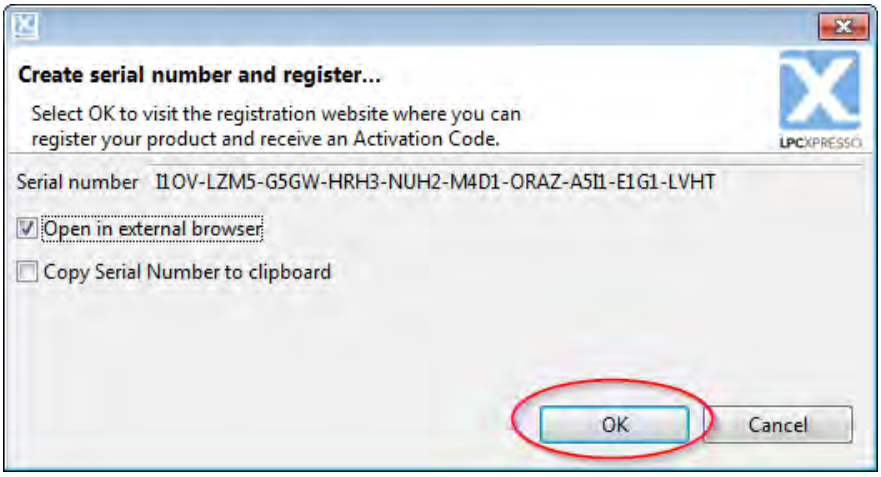


Fig 20. Product activation

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.

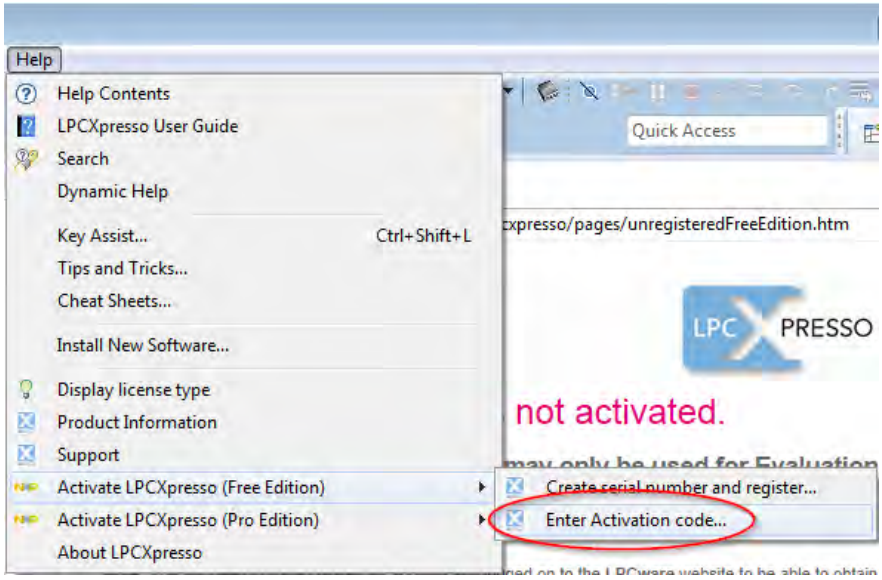


Fig 21. Product activation

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

All demo reader projects are divided into three sub projects.

One project that contains the NFC Reader Library, one project that contains all hardware dependent parts and one project that contains the user application.

The projects “Classic” and “Polling” are distributed in one package that can be imported into the LPCXpresso IDE in one single action. The following example is based on this package.

The sequence of installing the reference reader projects 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.

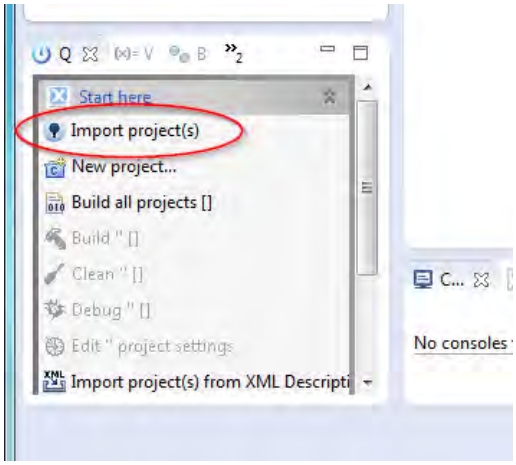


Fig 22. Importing a project into the LPCXpresso IDE

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

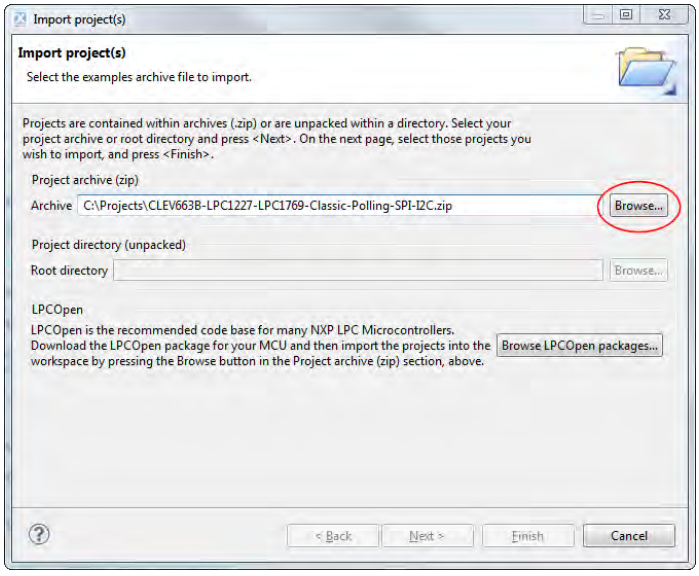


Fig 23. Importing a project into the LPCXpresso IDE

Browse the desired package and click “Next”.

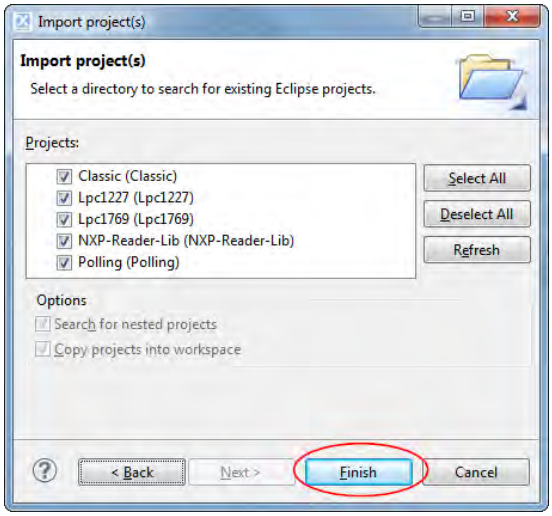


Fig 24. Importing a project into the LPCXpresso IDE

For a working demo project you need to import at least three sub projects. One application project (Classic or Polling), the NFC Reader Library and one MCU library (LPC1227 or LPC1769).

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

Before one can run the project, the LPCXpresso board containing the RC663 Blueboard needs to be connected to the computer. Wait until the adequate drivers have been installed.

4.3 Run the project

Before running the project, please ensure that the LPCXpresso with the Blueboard is connected to the computer. Please also make sure that the correct microcontroller and the correct build configurations are chosen. Information about how to do this can be found in the sections 7.2 and 7.3.

If one is using the projects with the Silica TUSA Board instead of the CLEV663B Blueboard, please find the needed modifications in the code at section 7.9.

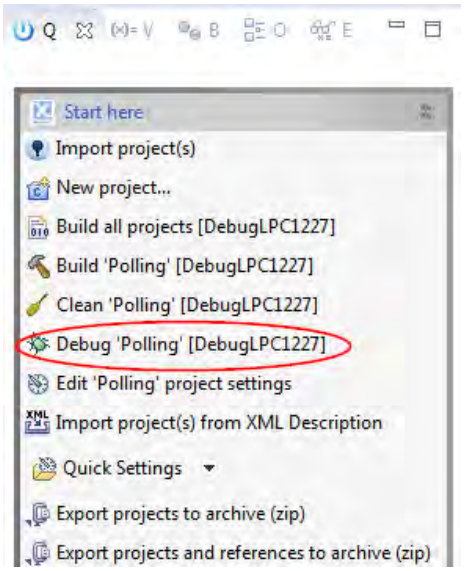


Fig 25. Run the project

Choose the project that contains the user application and click the Debug Button on the left hand side as shown in the example picture. Make sure, the name of the build configuration matches the name of the used microcontroller.

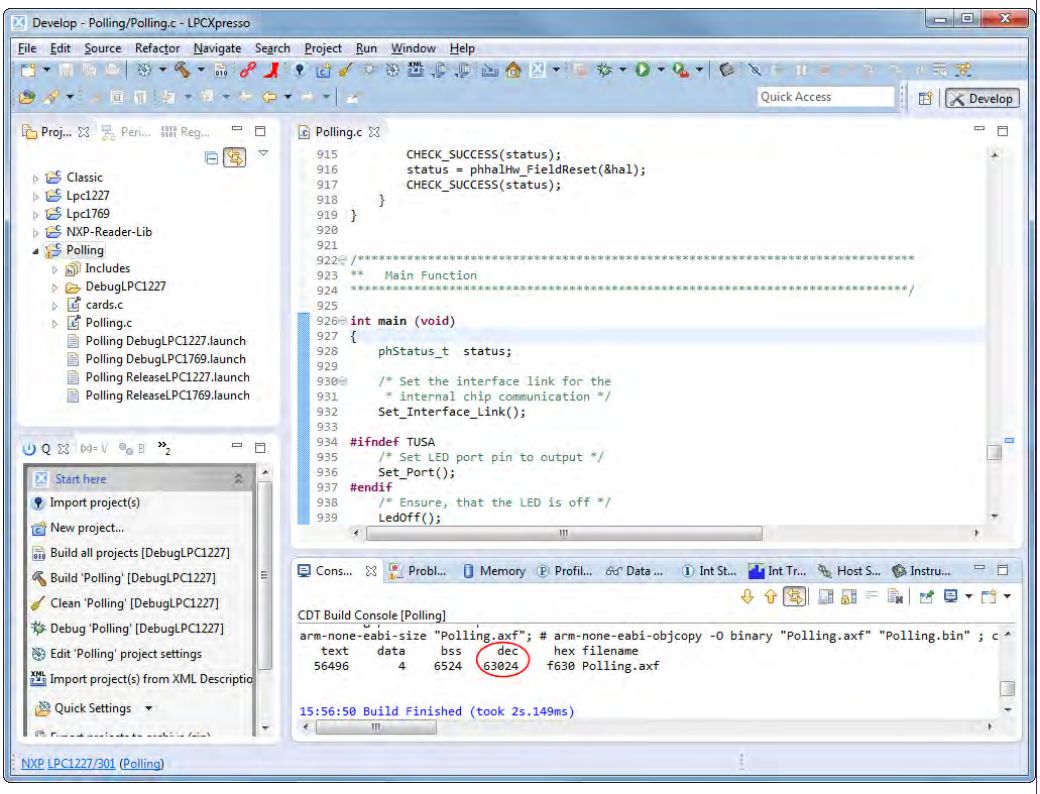


Fig 26. Run the project

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

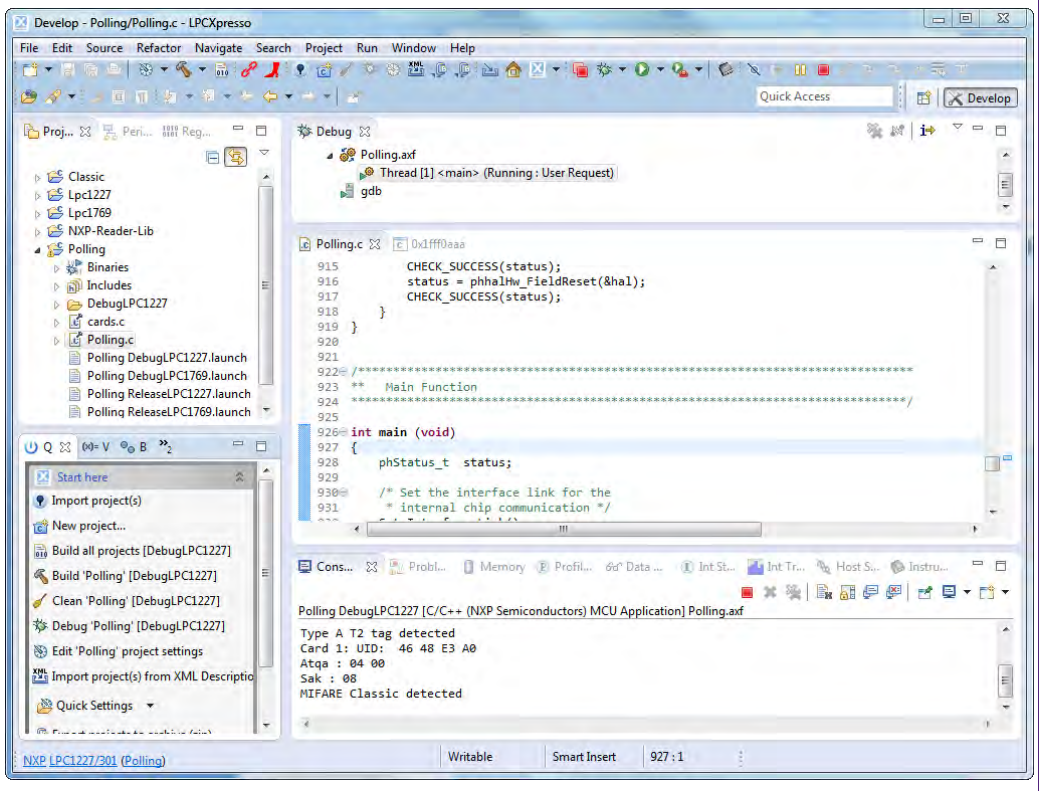


Fig 27. Run the project

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

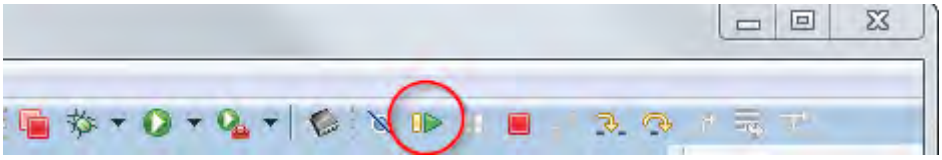


Fig 28. Debugging controls

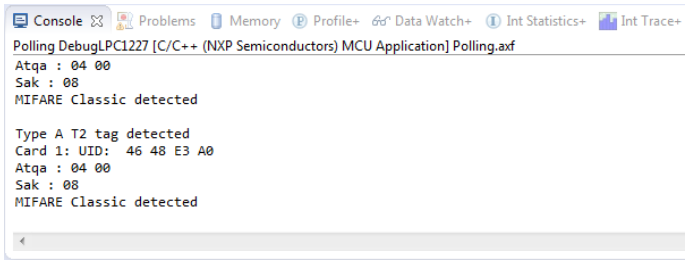


Fig 29. Run the project

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

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 30. Debug Buttons

5. Associated Projects

All example projects are available for download at the product page of the CLEV663B [8] 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 please extract it and run the installer. The installer just copies all you need to get started to your hard disk.

5.1 Communication with MIFARE Classic

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

5.2 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

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.

5.2.1 General card detection

This project makes use of the built in discovery loop to detect the type of cards inside the RF field.

5.3 Peer to Peer Initiator

Based on one example the user should get a basic understanding on how to use the P2P NFC technology.

This project can be imported into the IDE in a single step. The NFC Reader library projects as well as the LPC1769 projects are the same for all projects. So if someone has already one example package imported, he only needs to import the application projects to have everything for P2P in the workspace. The P2P example can be configured to act as a server or as a client.

For a detailed introduction into the P2P functionality please consult the user manual UM10721 - NXP NFC Reader Library User Manual. It can be downloaded at the web site of the CLEV663B demo board [8].

6. Other supported hardware by the projects

It's also possible to use the provided projects with the TUSA Board (Manufacturer: Silica). This board is a 3d party alternative to the CLEV663B Blueboard. It also uses the RC663 reader IC.

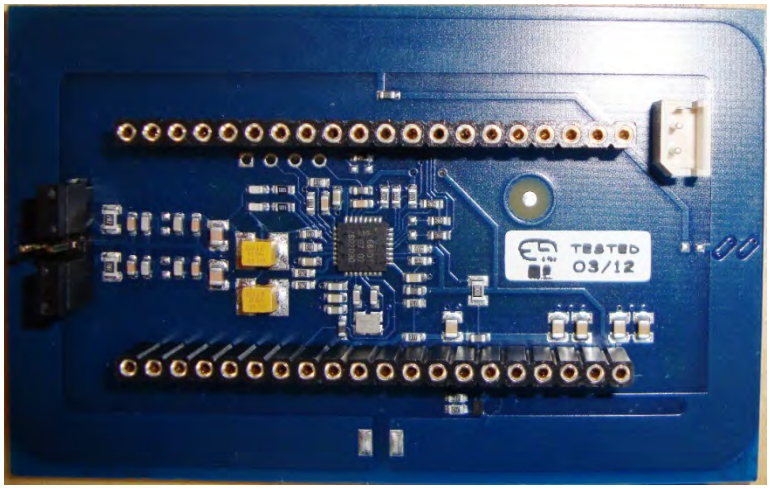


Fig 31. TUSA Board

To get this board to work with the LPC1227 or LPC1769 controller boards you will have to do a small hardware modification. Because the TUSA Board gets its power from the controller board, we need to solder a wire from the Capacitor C28 to the Pin 29 on the LPCXpresso Board. See figure:

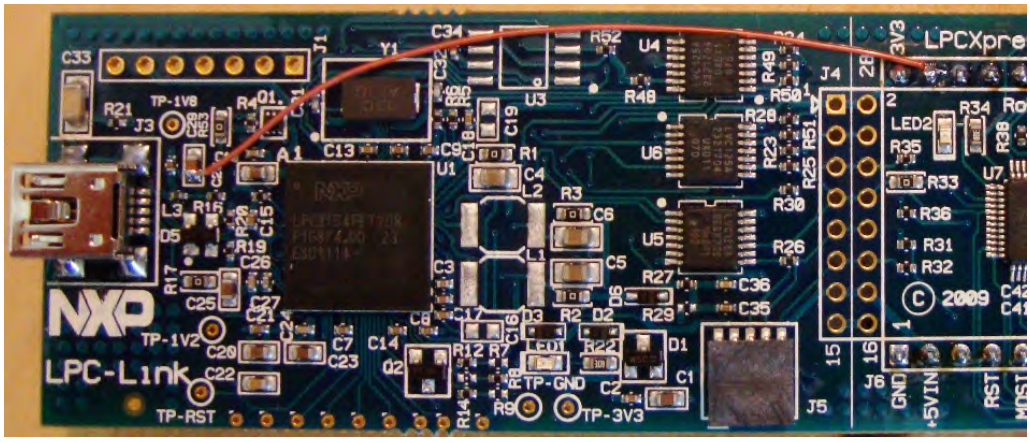


Fig 32. Modification on the LPCXpresso Board

To get a more detailed description of the TUSA Board and the modifications in the LPCXpresso controller board, please visit the product website at Silica [7].

After doing the modifications the hardware is ready and one can put the boards together like shown on the following figure.

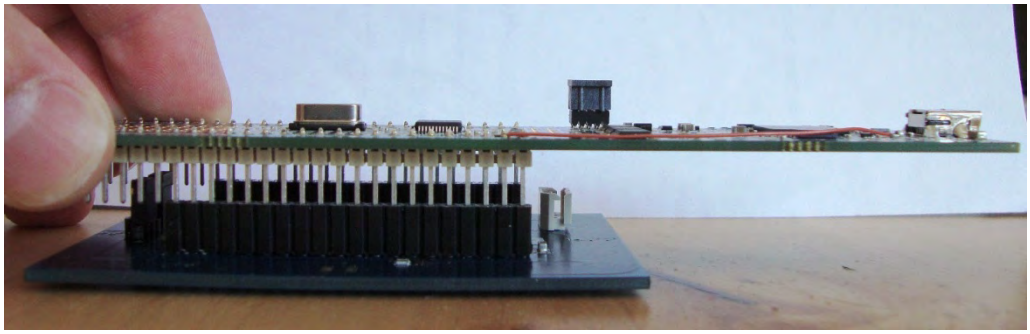


Fig 33. Combine the TUSA Board and the LPCXpresso Board

For information about the needed changes in the projects to work with the TUSA board, please see section 7.9.

7. Supplementary Notes

7.1 Software architecture

The software of the reference reader is based on the NFC Reader Library [6]. 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)

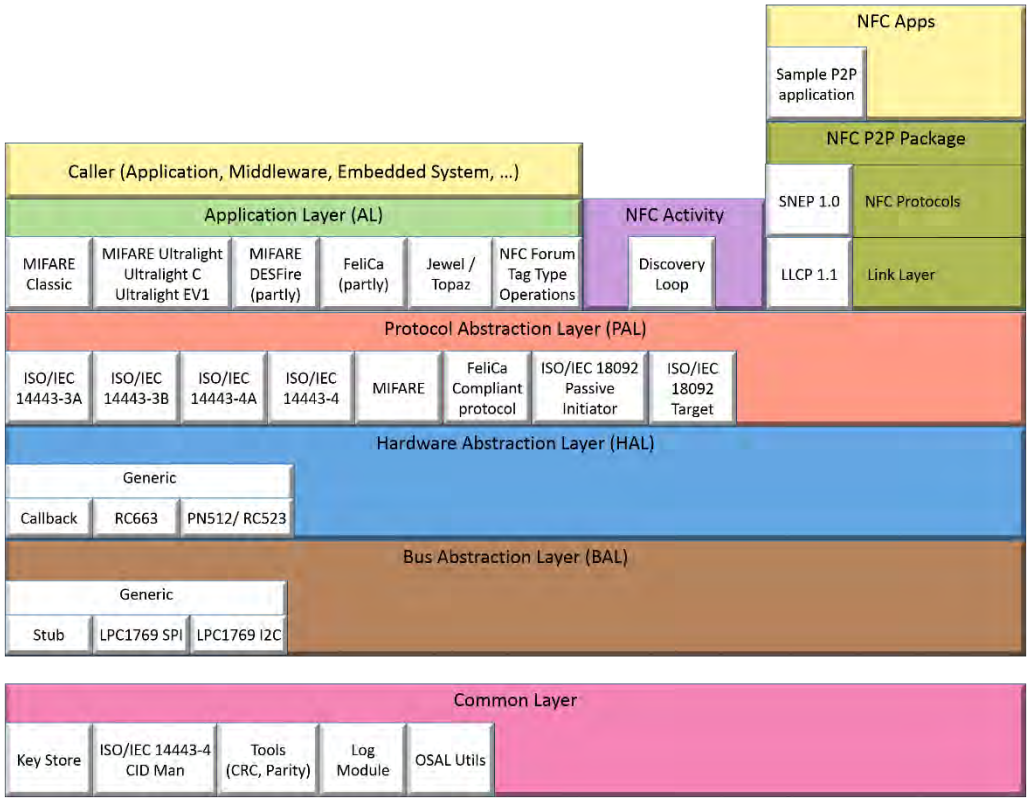


Fig 34. Architecture of the NFC Reader Library

For a detailed description of the NFC Reader Library please refer to the user manual **UM10721 - NXP NFC Reader Library User Manual**.

Documentation of the API can be found in the document **UM10802 - NXP NFC Reader Library API.** Both can be downloaded at the web site of the CLEV663B demo board [8].

7.1.1 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.

7.1.2 Hardware abstraction layer

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

7.1.3 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,
- ISO/IEC15693,
- FeliCa,
- NFC Initiator
- NFC Target

7.1.4 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.

7.2 Build configuration

All the projects mentioned in Chapter 5 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.

  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.

Projects that can run on different MCUs have dedicated build configurations for each MCU. For example the polling project works on the MCUs LPC1227 and LPC1769 and can be configured for release or debug configuration. Therefore this project offers four different build configurations:

- DebugLPC1227
- ReleaseLPC1227
- DebugLPC1769

- ReleaseLPC1769

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 DebugLPC1227 build (or ReleaseLPC1227 build) for LPC 1227.

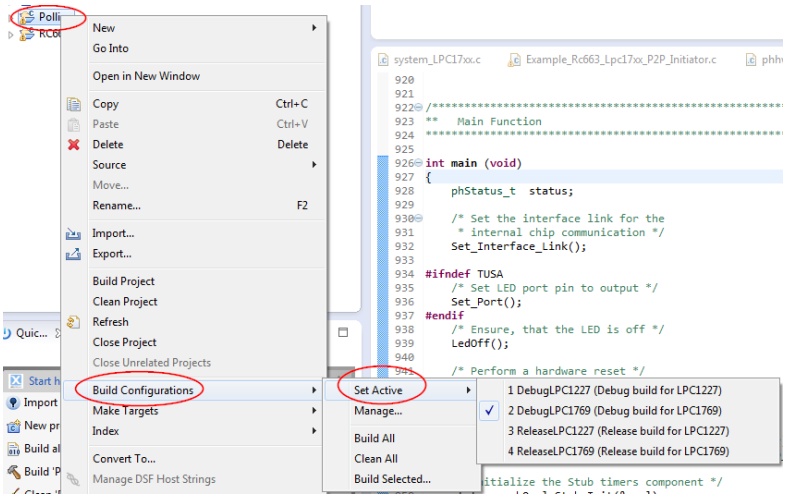


Fig 35. Select the build configuration

Note: When switching from one MCU to another, please take care to also switch to the correct MCU setting within the LPCXpresso IDE. See section 7.3.

7.3 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

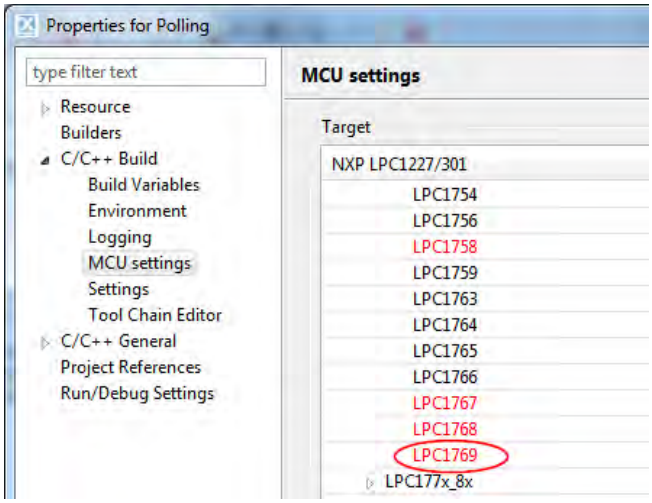


Fig 36. Selection of the LPC1769 MCU

7.4 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 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.

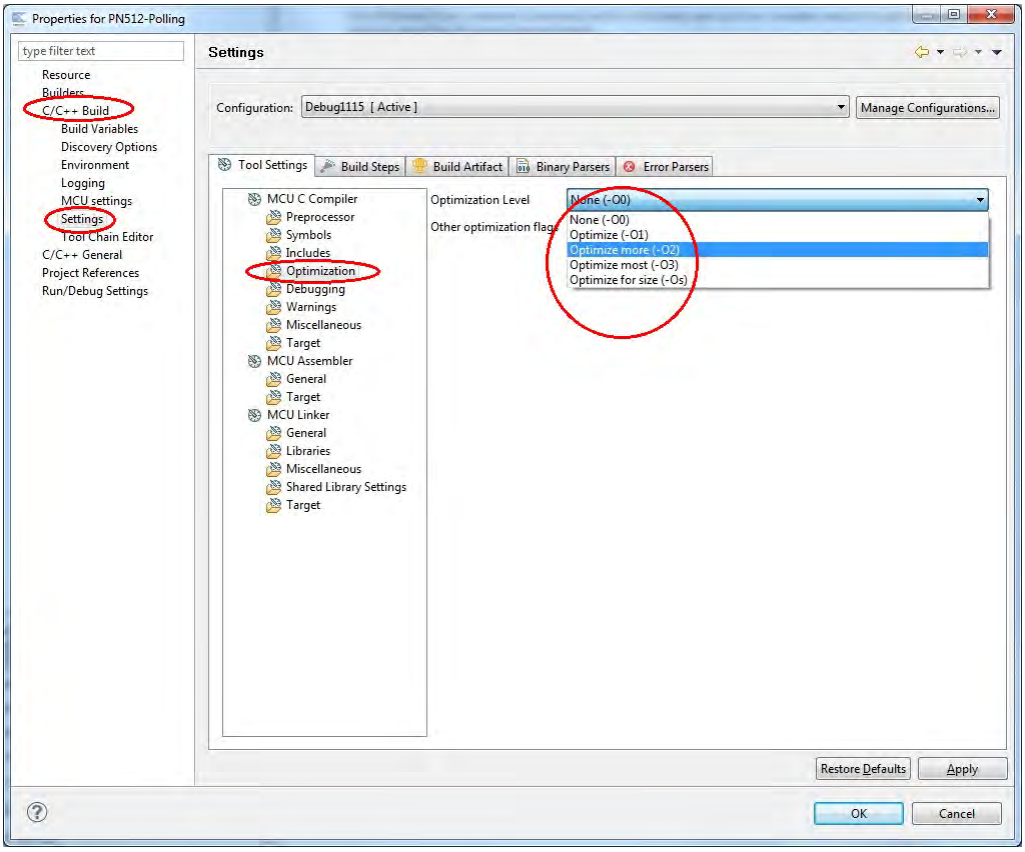


Fig 37. Select the level of optimization

7.4.1 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.

7.5 Optimizing the code size of the NFC Reader Library

A detailed description on how to optimize the code size of the NFC Reader Library for the use with one specially defined reader IC and card type please refer to the attached documents on the product page of the CLEV663B [8]. On that page one can also find an exemplary project for the use of the MIFARE Classic card in conjunction with the CLRC663 reader IC.

7.6 Preparing the projects for the use of the Blueboard in I²C configuration

To use the projects in I²C configuration one has to do some small adaptations in the file phhwConfig.h located in the MCU project LPC1xxx.

1. Open the file phhwConfig.h and

a. Uncomment the line **`#define I2C_USED.`**

b. Comment the line **`#define SPI_USED.`**

7.7 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…

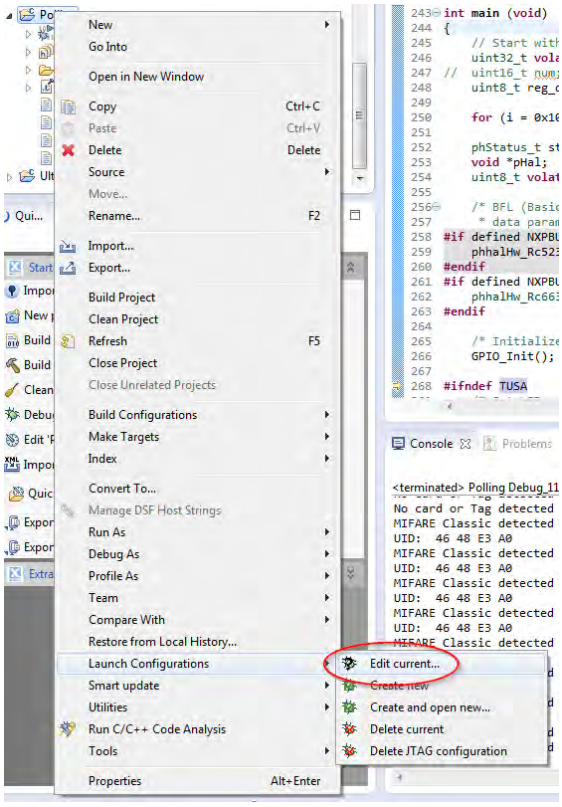
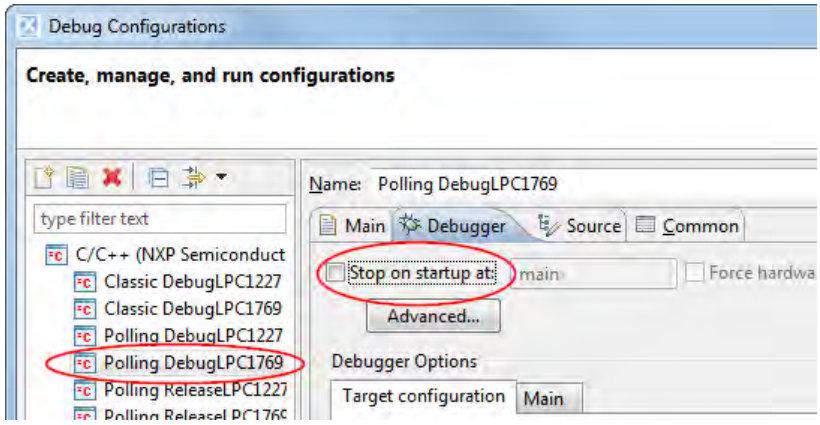


Fig 38. Open Current Launch Configuration

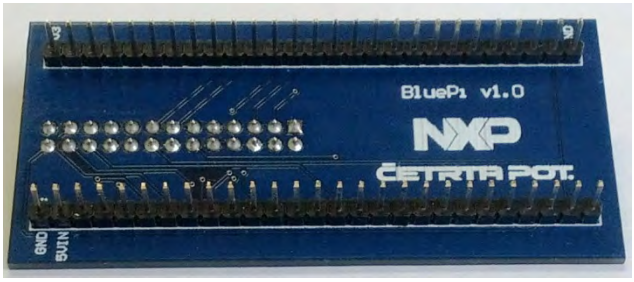
1. Choose the Debug configuration
2. Choose the tab Debugger
3. Uncheck the box near “Stop on startup at:”
4. Click onto Apply and then Close.



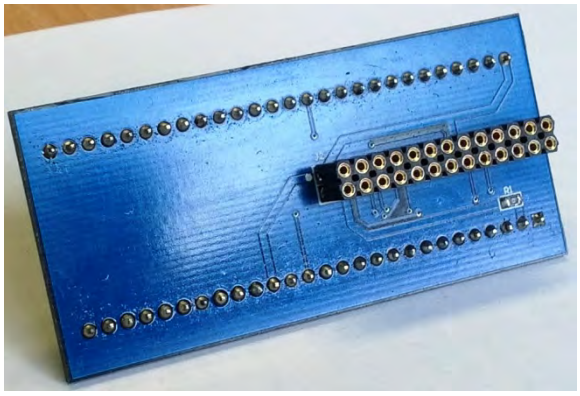
7.8 Using the Blueboard with the Raspberry Pi

Detailed instructions about how to run the NFC Reader Library on the Raspberry Pi can be found in the Start Guide linked at the product page of the EXPLORE-NFC [13]. This sub chapter only describes some information that is not included in the start guide of the EXPLORE-NFC.

7.8.1 Preparing the hardware



a. Connector for the Blueboard



b. Connector for the Raspberry Pi

Fig 39. BluePi adaptor

1. Connect the CLEV663B Blueboard to the BluePi adaptor.
2. Connect the adaptor to the Raspberry Pi
3. Connect the USB interface of the Blueboard to the USB interface of the Raspberry Pi.

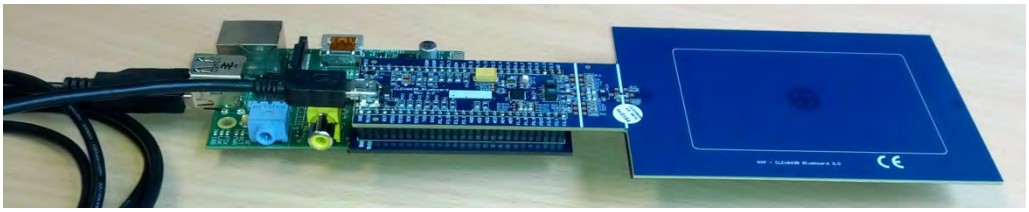


Fig 40. CLEV663B Blueboard connected to the Raspberry Pi

7.8.2 Preparing the software

In order to use the provided software with the CLEV663B there need to be two lines changed within the source code.

1. Open the file ../card_polling/source/NXP-Reader-Library/types/ph_NxpBuild.h
2. Comment #define `NXPBUILD__PHHAL_HW_RC523`
3. Uncomment #define `NXPBUILD__PHHAL_HW_RC663`

7.9 Preparing the projects for the use with the 3d party Tusa Board

It is assumed that the project is currently configured to be used with an LPCXpresso LPC1769 together with an RC663 Blueboard.

Please follow these steps to prepare the software project to get the TUSA board running.

1. From the sub project Lpc1769 open the file phhwConfig.h.
2. Uncomment the definition "`#define TUSA`"
3. Make sure, SPI is activated with the define "`#define SPI_USED`"
4. Clean the project, so that everything gets rebuilt.

If you want to use the TUSA board together with the LPC1227, the reconfiguration of the project works in the same manner.

8. References

[1] **RC663 and NFC Reader Library**
http://www.nxp.com/documents/application_note/AN11021.pdf

[2] **LPCXpresso website**
www.nxp.com/redirect/lpcware.com/lpcxpresso/download

[3] **RC663 data sheet**
http://www.nxp.com/documents/data_sheet/CLRC663.pdf

[4] **LPC176x/5x User manual**
http://www.nxp.com/documents/user_manual/UM10360.pdf

[5] **Multipoint Connectors we used:**
Grid Dimension: 2.54mm, at least 27 pins
http://www.nxp.com/redirect/conrad.at/ce/de/product/741119/STIFTLEISTE-1-X-36-POLIG-VERGOL-RM-254
and
http://www.nxp.com/redirect/conrad.at/ce/de/product/736427/BUCHSENLEISTE-EINREIHIG-36-POLIG-RM254

[6] **Direct link to the NFC Reader Library**
Not yet available

[7] **TUSA Board at the Silica website**
http://www.nxp.com/redirect/silica.com/products/highlight/product/silica-tusa-board.html

[8] **CLEV663B demo board site**
http://www.nxp.com/demoboard/CLEV663B.html

[9] **NXP NFC Reader Library User Manual**
http://www.nxp.com/documents/user_manual/UM10721.pdf

[10] **Technical Specification** – Simple NDEF Exchange Protocol, NFCForum-TS-SNEP_1.0
www.nxp.com/redirect/nfc-forum.org/specs/spec_license

[11] **LPCXpresso LPC1769 development board**
http://www.nxp.com/demoboard/OM13000.html

[12] **EMV** – The table of card types and their matching AIDs are available on
http://www.nxp.com/redirect/en.wikipedia.org/wiki/EMV

[13] **EXPLORE-NFC product page**
http://www.nxp.com/demoboard/PNEV512R.html#documentation

9. Legal information

9.1 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.

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.

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.

9.3 Licenses

Purchase of NXP ICs with ISO/IEC 14443 type B functionality

This NXP Semiconductors IC is ISO/IEC 14443 Type B software enabled and is licensed under Innovatron's Contactless Card patents license for ISO/IEC 14443 B.

The license includes the right to use the IC in systems and/or end-user equipment.

RATP/Innovatron Technology

9.4 Trademarks

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

MIFARE — is a trademark of NXP Semiconductors N.V.

DESFire — is a trademark of NXP Semiconductors N.V..

MIFARE Ultralight — is a trademark of NXP Semiconductors N.V..

MIFARE Plus — is a trademark of NXP Semiconductors N.V.

10. List of figures

11. Contents

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

Date of release: 21 July 2014
226218
Document identifier: AN11211