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

AN11308

Quick Start Up Guide PNEV512B Board

Rev. 2.5 — 26 September 2018
255025

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	PN512, Blueboard, LPCXpresso, MCU, eclipse, LPC1769, LPC11U68, NFC Reader Library, PNEV512B
Abstract	This application note is related to the installation procedures of the PNEV512B Board. It describes the actions to be done to become acquainted with the demo reader

Revision history

Rev	Date	Description
2.5	20180925	Reference in Section 7.1 updated
2.4	20180626	Editorial updates
2.3	20161123	Updated examples descriptions
2.2	20160901	Added LPCXpresso version 8.1.4 requirement Added RTOS options
2.1	20160318	Added description about Ex10 – MIFARE DESFire
2.0	20160125	Added description about LPC11U68 support.
1.9	20151111	Updated all relevant parts in respect to the NFC Reader Library update Removed LPC1227 support
1.8	20141201	Added a note about RAM limitation of LPC1227.
1.7	20140721	Small corrections
1.6	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 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 NFC Reader Library is now called NFC Reader Library
1.5	20140114	Added description for Card Emulation T4T and T2T.
1.4	20131011	Added info about what version of the LPCXpresso IDE to use.
1.3	20130613	Added description about the P2P Snep Client
1.2	20130221	Added description of the P2P project. 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.
1.1	20130108	Red circles of some figures corrected
1.0	20121217	First release

Contact information

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

1. Introduction

This application note gives a detailed overview of the hardware for working with the PN512 contactless reader IC, we use the LPCXpresso LPC 1769 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.

The projects used and explained in this documentation are:

Table 1. Example projects

Example projects delivered with the NFC Reader Library

Example	Description
Example 1 – Basic Discovery Loop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. Default configuration parameters are used.
Example 2 – Advanced Discovery Loop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. All configuration parameters are used and explained.
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.
Example 4 – MIFARE Classic	Explains the usage of MIFARE Classic commands.
Example 5 – ISO1593	Explains the usage of this technology and provides an overview about the most common commands.
Example 6 – EMVCo Loopback	Application used for EMVCo Level 1 PCD certification.
Example 7 – EMVCo Polling	Explains polling for EMVCo payment cards.
Example 8 – HCE T4T	Explains how to emulate a NFC Forum Type 4 Tag supporting read and write operations.
Example 9 – NTAG-I2C	Explains NTAG-I2C specific commands.
Example 10 – MIFARE DESFire	Explains the usage of MIFARE DESFire IC-based cards.
Example 11 – SimplifiedAPI EMVCo	EMVCo loopback application with simplified API, which can be used for EMVCo level 1 digital certification.
Example 12 – SimplifiedAPI ISO	Explains how to use simplified API with different types of cards.

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

2. Hardware overview of the Demo Reader

The demo reader is made up of 2 separate boards:

- A PNEV512B demo board provided by NXP (12NC: 9352 981 99699). This board has connectors which are designed to exactly fit the ones of the companion 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].
  All projects can be used with the LPCXpresso LPC11U68 development board (**12NC:** 935303579598) as well. See Ref. [14].

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

2.1 PNEV512B demo board

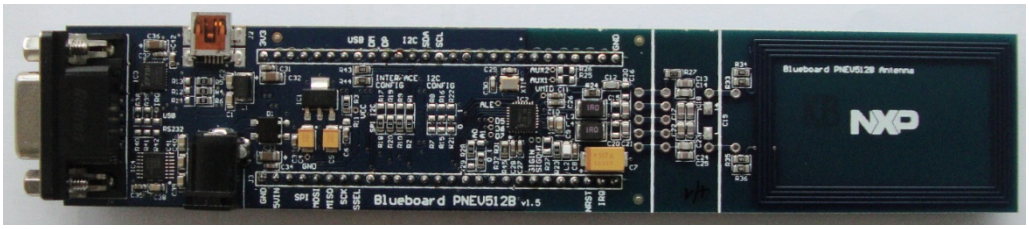

Fig 1. Picture of PNEV512B demo board

The PNEV512B demo board embeds the contactless communication transceiver IC PN512 with all its elements needed for transmission: EMC filter, matching network and the antenna. The PN512 supports different kind of contactless communication methods and protocols at 13.56 MHz:

- Reader/Writer mode supporting ISO/IEC 14443A, MIFARE ICs and FeliCa scheme
- Reader/Writer mode supporting ISO/IEC 14443B
- Card Operation mode supporting ISO/IEC 14443A, MIFARE ICs and FeliCa scheme
- NFCIP-1 mode
- Refer to the data sheet of this IC [2] for more details

Thanks to the relevant solder bridges, the host link of the PNEV512B demo board can be configured for:

- I²C
- SPI
- UART (optional, see Section 2.7)

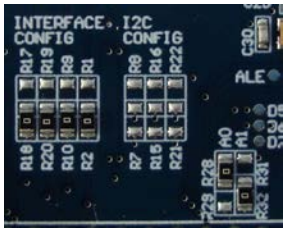

Fig 2. Picture of solder bridges in default configuration

The default interface configuration of the PNEV512B demo board is SPI. The detailed interface configuration is described in section 2.7.

Additional interface and power supply options are described in section 2.8.

2.2 CE certification of the Blueboard

The current version of the demo board (v.1.5) 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 [3]

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 Preparation of the hardware

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

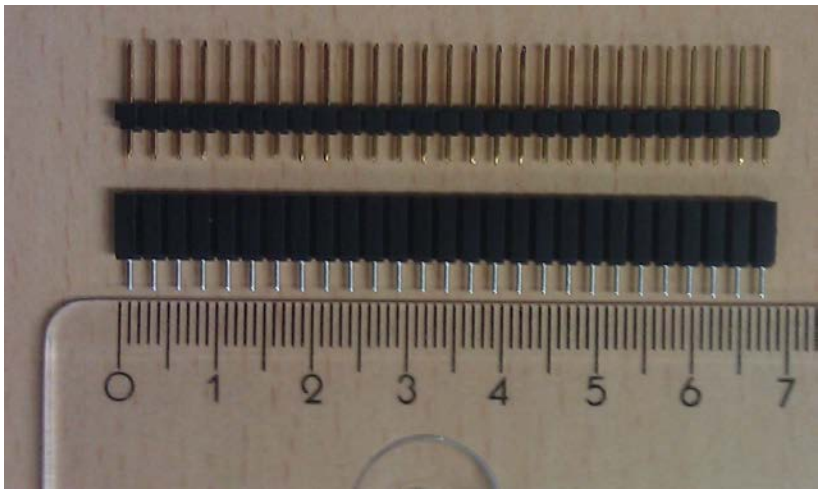

Fig 4. Multipoint Connectors we used

One may buy these connectors at any electronic store. Here are some examples [4].

After soldering the connectors, join the boards as shown on the following figure.

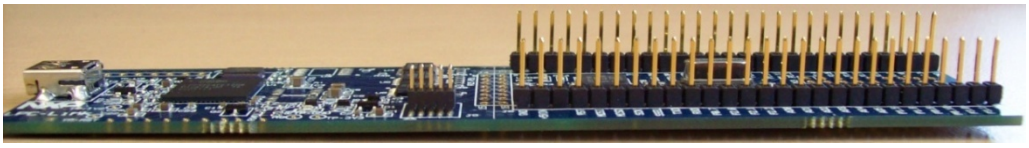

Fig 5. LPCXpresso with the Multipoint Connectors

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

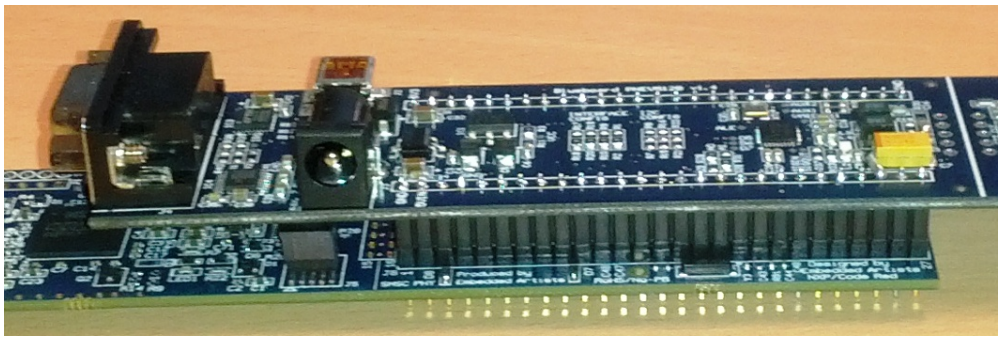

Fig 6. Connect the two boards

2.5 Soldering the interrupt connection

To get the interrupt working, please connect the pins 26 and 27 of the PNEV512B board.

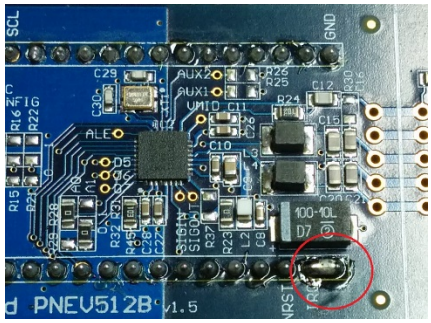

Fig 7. Interrupt connection

2.6 Interesting points of measurement

On the PNEV512B demo board one can find test pads for measurement purposes.

- VCC
- GND
- D5
- D6
- D7
- ALE
- AUX1
- AUX2
- SIGIN
- SIGOUT
- IRQ
- VMID

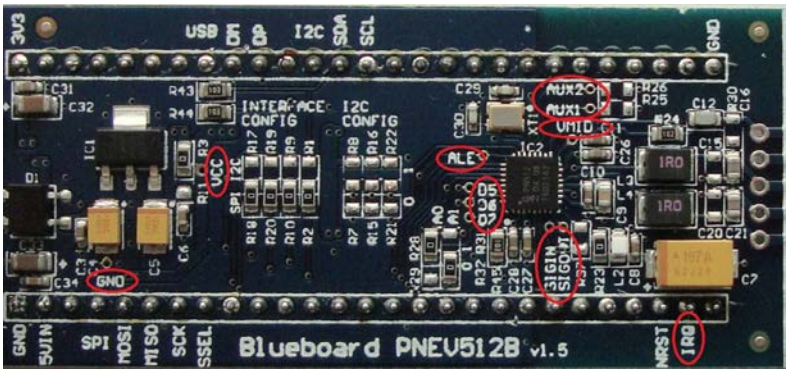

Fig 8. Interesting points of measurement

2.7 Preparing the Blueboard for the use with SPI or I²C

The Blueboard is generally delivered in SPI configuration. To change the interface to I²C the four appropriate 0R0 resistors in the interface config section need to be resoldered to the I²C side of the solder jumpers. Also the two 0R0 resistors at A0 and A1 need to be changed.

Table 2. A0 and A1 interface configuration

Appropriate solder jumpers (0R0 resistors) for interface configuration

Signal	Interface type		
	SPI	I²C	UART(optional)
A0	R28	R29	R29
A1	R32	R31	R32

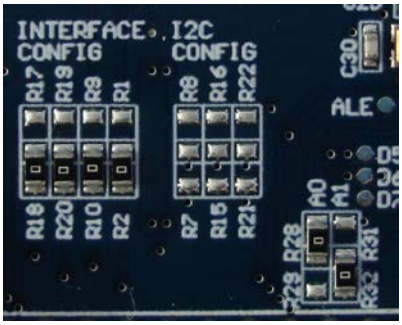

Fig 9. Blueboard in SPI configuration

To use the Blueboard in I²C configuration with the provided software projects, one has to carry out two minor adaptations in the code, which are described in section 7.5.

The I²C-address can be configured either by software or by hardware. To set the I²C-address by hardware the solder jumpers in the I²C config section (see picture above) have to be connected appropriately. R7, R15 and R24 are logically LOW and R8, R16 and R22 logically HIGH.

2.8 Optional interfaces and power supply

The PNEV512B demo board is normally controlled by the LPCXpresso Board. With the optional interfaces and power supply the demo board can be controlled directly by a PC without the LPCXpresso Board.

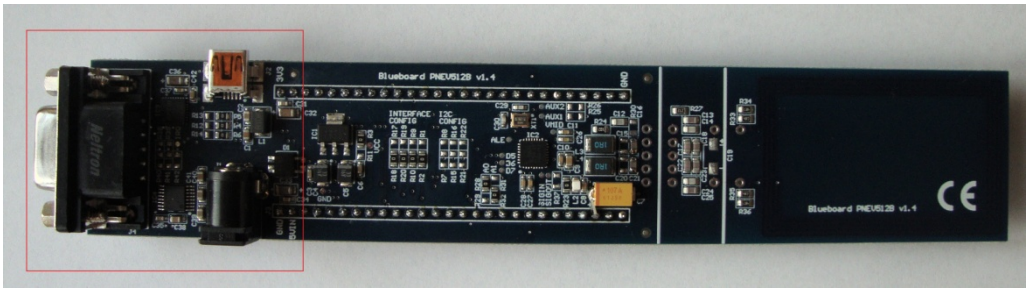

Fig 10. Additional interfaces

To use the additional interfaces the interface selection signals A0 and A1 have to be configured to UART mode (see section 2.7, Table 2).

2.8.1 Configuring the additional interfaces

With the appropriate solder jumpers two different serial interfaces can be selected.

Table 3. Solder Jumpers for selecting the additional interfaces

Interface type	Resistors
USB	R38, R39
RS232	R40, R41, R42

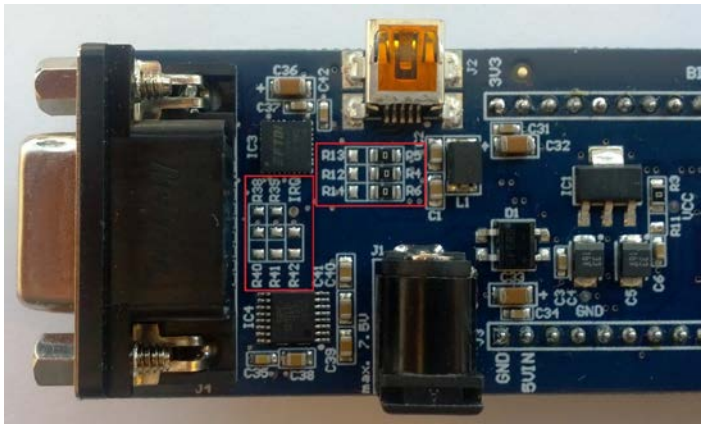

Fig 11. Solder jumpers for additional interface configuration

In delivery default configuration the USB-connector of the PNEV512B demo board is directly routed to the USB-pins of the LPCXpresso Board µC in order to use the USB connector as an additional USB connector of the LPCXpresso Board.

For using the USB interface in UART mode the following solder jumper configuration is needed:

Table 4. Solder Jumpers for USB connector configuration

Connection type	Resistors
LPCXpresso-USB	R4, R5, R6
UART-USB	R12, R13, R14

2.8.2 Configuring the power supply

When using the PNEV512 demo board without the LPCXpresso Board an additional 5VDC power supply is needed. The onboard voltage regulator provides the 3.3VDC supply voltage VCC.

Table 5. VCC power supply configuration

Power supply	Resistors
LPCXpresso Board	R3
External 5VDC	R11

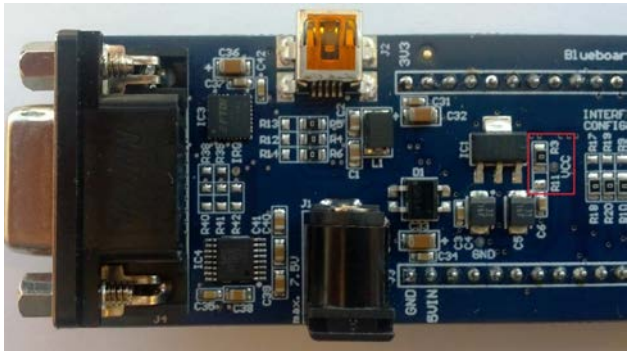

Fig 12. Solder jumper for VCC power supply configuration

2.9 Additional supported development boards

All example projects described in this document are preconfigured for the LPCXpresso LPC11U68 development board as well. The LPC11U68 is a Cortex-M0+ microcontroller with 256kB flash memory.

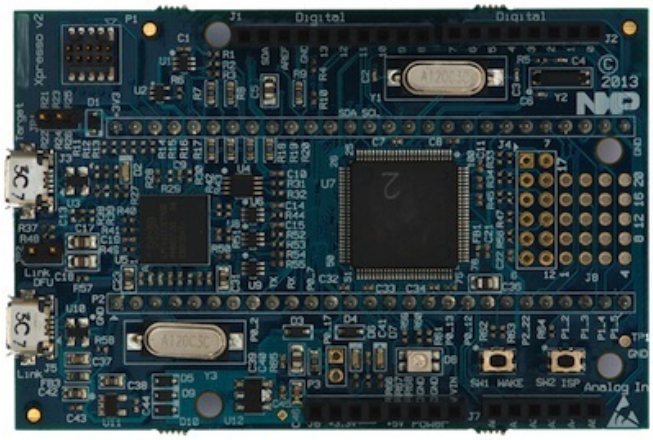

Fig 13. LPCXpresso LPC11U68 development board

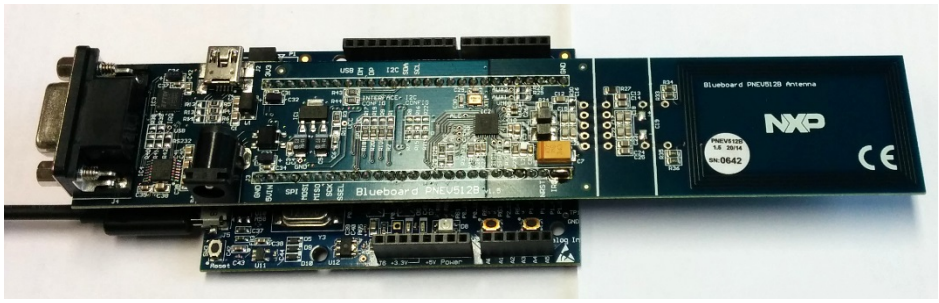

Fig 14. CLEV663B connected to LPCXpresso LPC11U68 development board

2.10 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 [12].

Although this guide only describes the use of the EXPLORE-NFC extension board, it also supports the PNEV512B Blueboard. The Blueboard can be used with a special adaptor called BluePi. For information about how to configure the hardware and the software please refer to section 7.7.

3. Installation of the LPCXpresso Board

The guidelines for installing the reader are as follows:

- Connect the LPCXpresso Board 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.

  Once the Board has been installed, open the Device Manager of the PC to check that the installation was successful. The item “USB Device with DFU Capabilities” is being displayed.

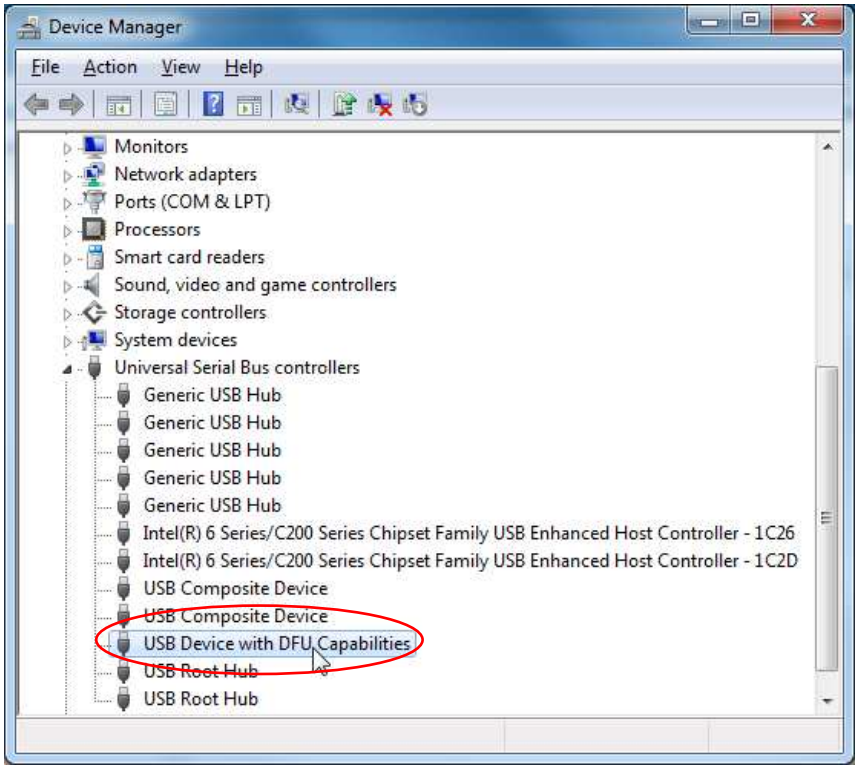

Fig 15. Enumeration of the LPCXpresso Board in Device Manager Window

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 from 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 [1].

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 side effects. Be sure to download LPCXpresso IDE version 8.1.4 or higher. The installation starts after double-clicking the installer file.

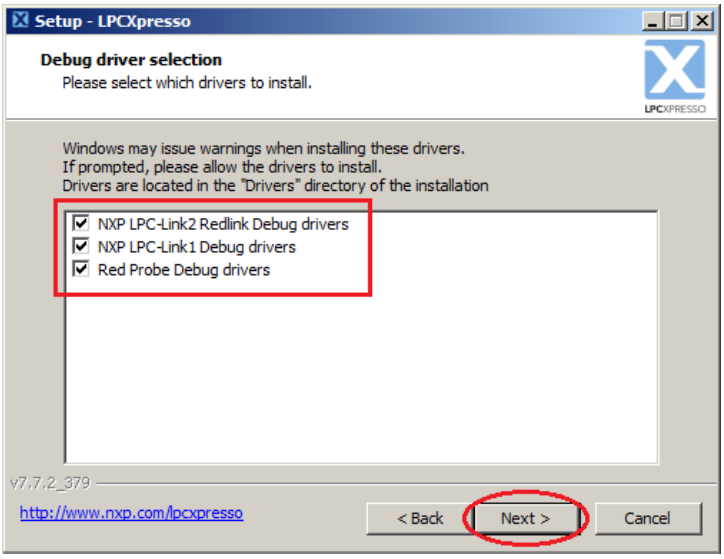

Fig 16. LPCXpresso installation NXP Debug drivers

Make sure, the checkboxes for installing the NXP Debug drivers are 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.

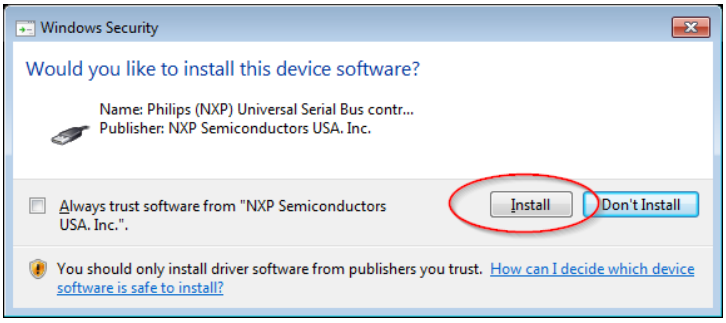

Fig 17. Windows Security dialog

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

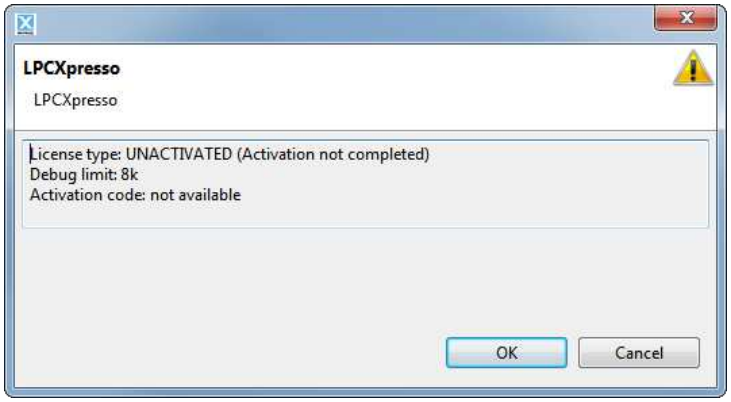

Fig 18. LPCXpresso IDE

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 can be done at the LPCware website. The Link is shown in the menu, Help → Activate → Create Serial number and register...

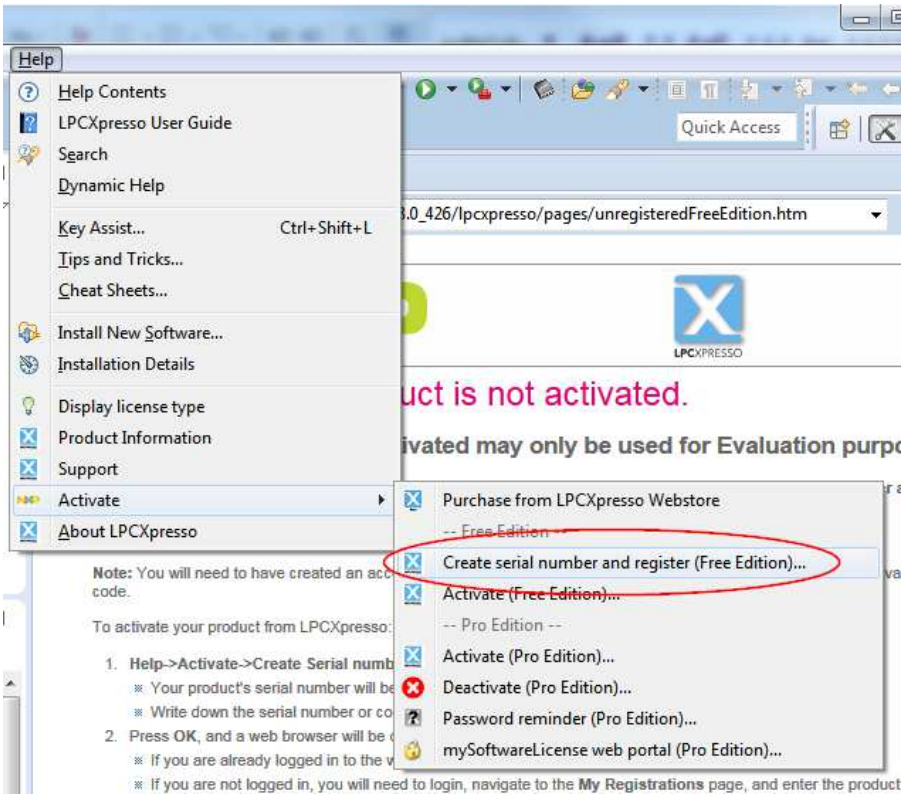

Fig 19. Product activation

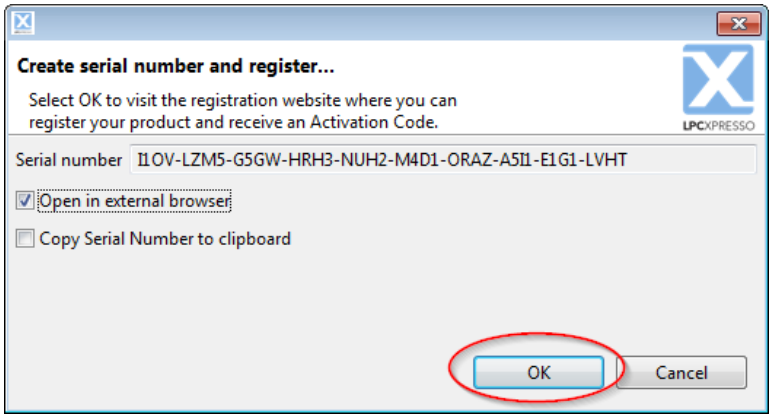

Fig 20. Product activation

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

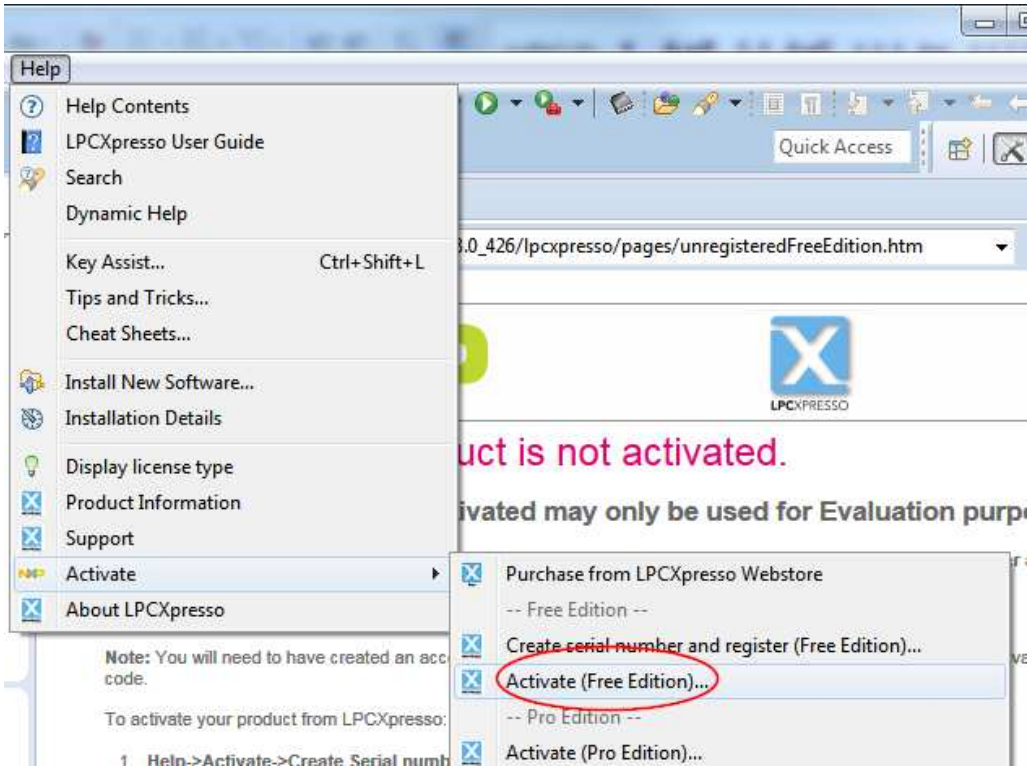

Fig 21. Product activation

Once the activation code arrives please open the activation window by pointing to Help → Activate → Activate (Free Edition), 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 the FreeRTOS operating system, one project that contains all hardware dependent parts and one project that contains the example application.

All projects are distributed in one zip 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.

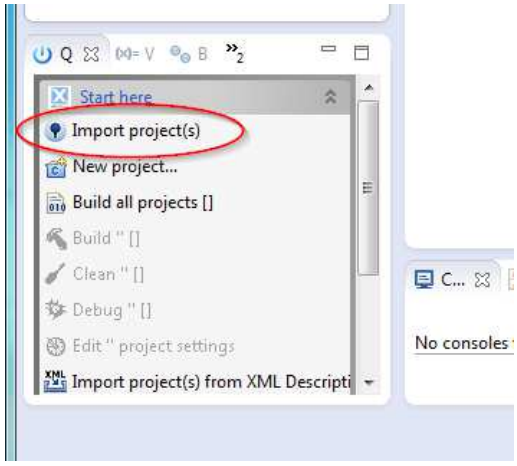

Fig 22. Importing a project into the LPCXpresso IDE

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

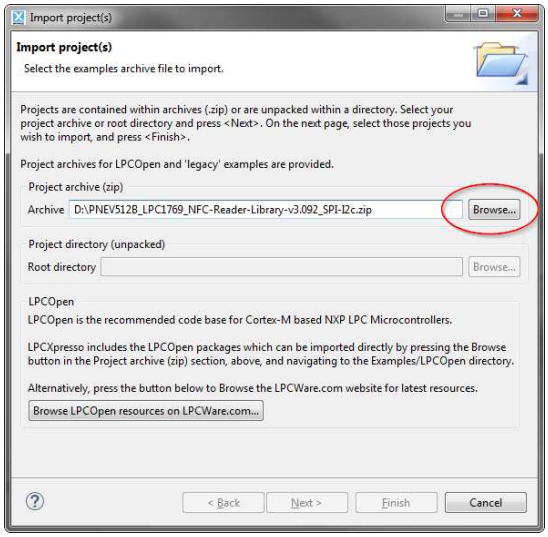

Fig 23. Importing a project into the LPCXpresso IDE

Browse the desired package and click “Next”.

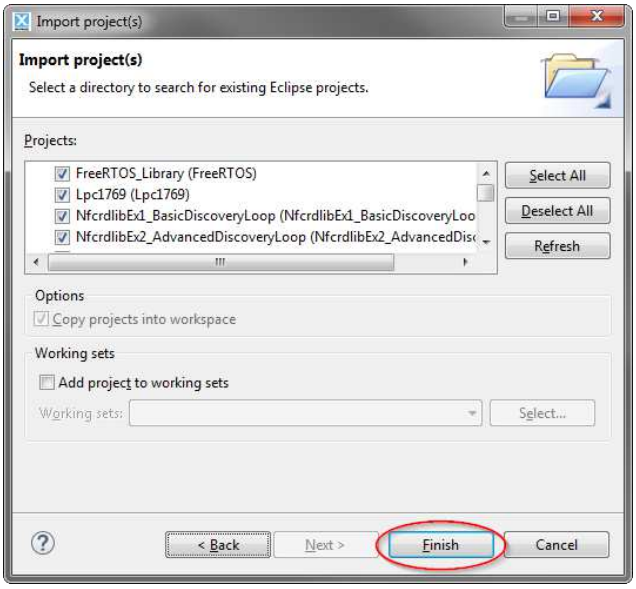

Fig 24. Importing a project into the LPCXpresso IDE

For a working demo project, you need to import at least 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 PN512 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.

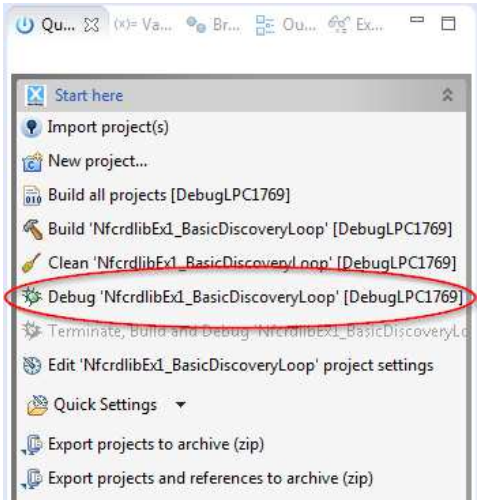

Fig 25. Run the project

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 as well as the selected MCU matches the name of the used microcontroller. See sections and 7.3 and 7.4 for further information.

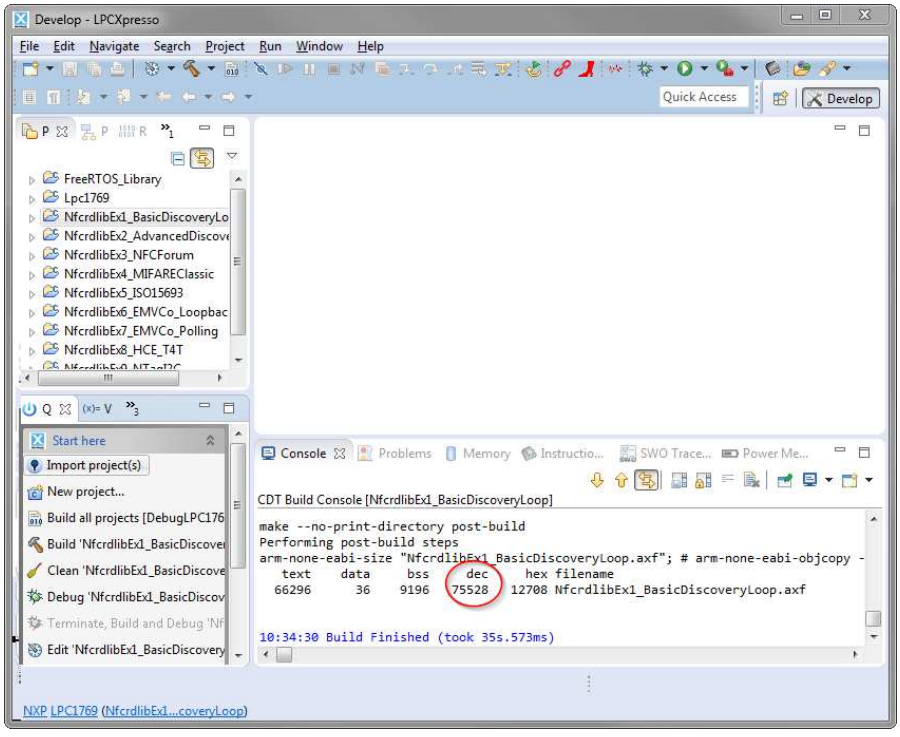

Fig 26. After the build process one can see the size of the image in the console window.

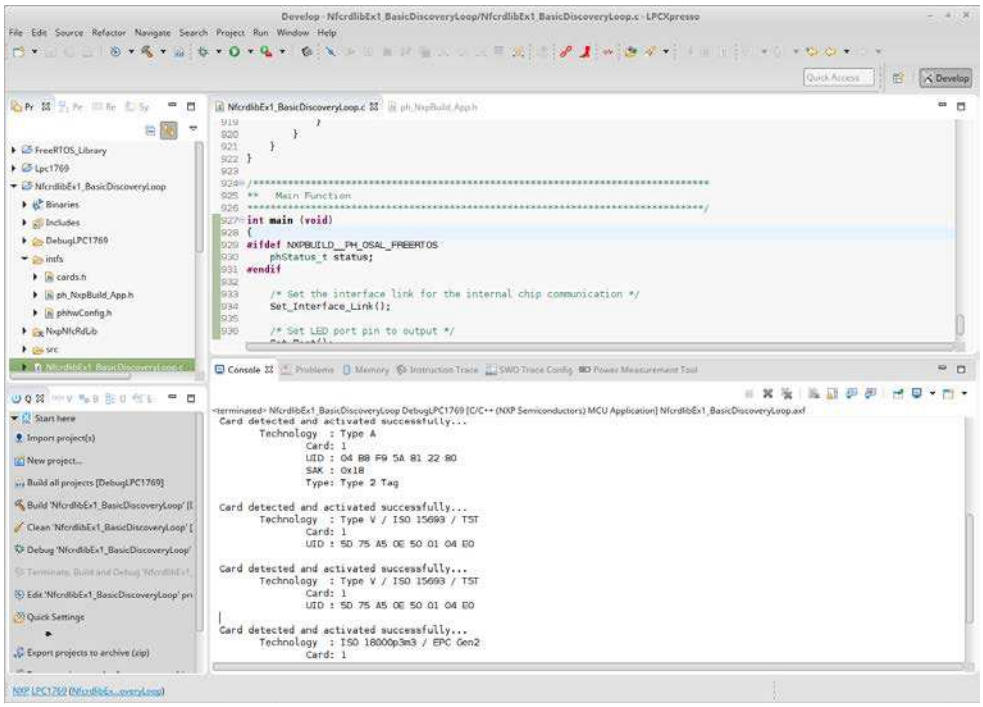

Fig 27. Run the project

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

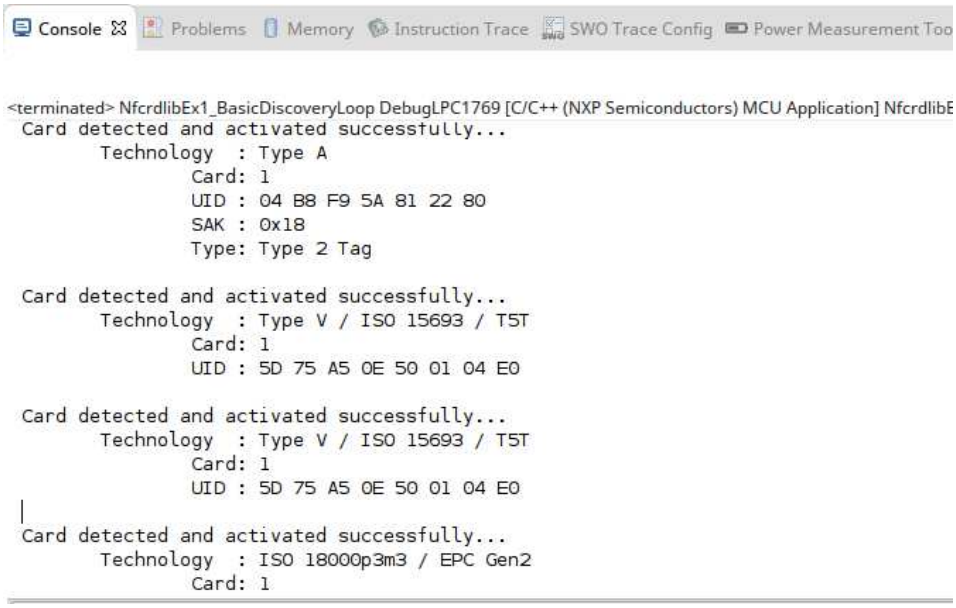

Fig 29. Run the project

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. 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

6. Associated Projects

All example projects are available for download at the product page of the PNEV512B [7] in the documents section and are being distributed in one single file.

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

By default, the projects are preconfigured to be run on the LPCXpresso LPC1769 development board. For instructions about how to run the projects on the LPCXpresso LPC11U68 development board, refer to chapter 7.2 and 7.3, 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
```

6.1 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 product-based 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.

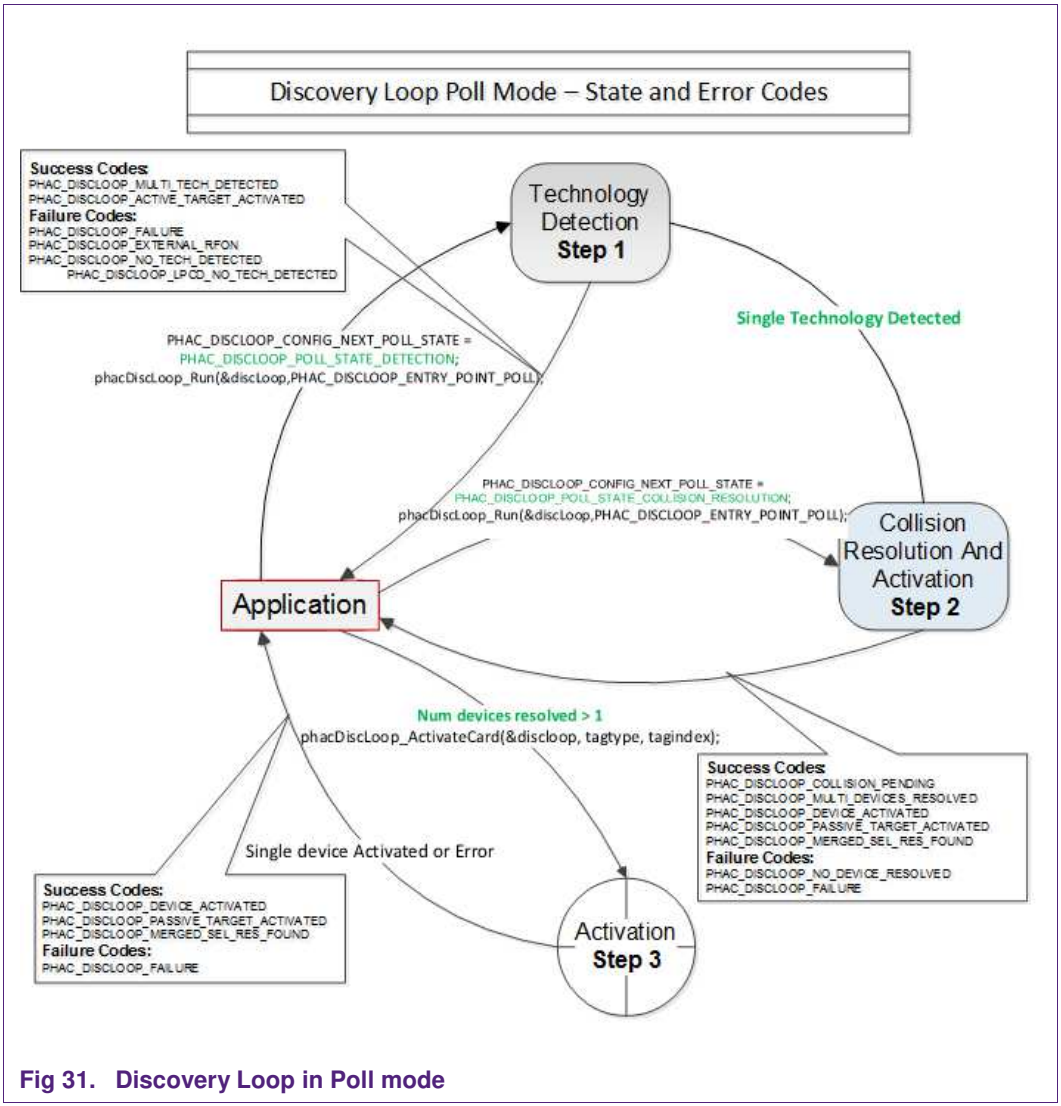

Fig 31. Discovery Loop in Poll mode

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 6. Supported technologies

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

6.2 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.

6.3 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.

6.4 Example 4 – MIFARE Classic

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

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

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

6.5 Example 5 - ISO15693

ISO15693 feature is not supported by PN512, therefore it this example is not available for PNEN512B bluebord.

6.6 Example 6 – EMVCo Loopback

The EMVCo Loopback example is a LoopBack application which is used to perform EMVCo 2.5 level 1 digital compliance validation. Example can be used as such without any changes to be used for the EMVCo certification.

6.7 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.

6.8 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).

6.9 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 product page of the same [13].

6.10 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.

6.11 Example 11 – SimplifiedAPI EMVCo

This example is similar to the "EMVCo Loopback" example, as it can be used to perform EMVCo 2.5 level 1 digital compliance validation.

The different between both examples is in NFCReaderLibrary initialization, where this example is using simplified reader library initialization process. Simplified approach, after library initialization, is using only three commands: