AN11658

NXP-NCI NullOS integration example

Rev. 1.2 — 6 July 2015
323112

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	NXP-NCI, NullOS, NFC, example, LPCXpresso
Abstract	This document intents to provide a description of the NXP-NCI_LPCXpressoExample project. This project demonstrates simple integration of NXP NCI NFC Controller without any OS resources required.

Revision history

Rev	Date	Description
1.2	20150706	Support of LPC82x and LPC11Uxx MCU families added. Adding chapter 5.2 for porting to others MCU recommendation
1.1	20150521	Support of LPC11xx MCU family added
1.0	20150310	Released version
0.1	20141015	Initial version of the document

Contact information

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

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. Introduction

The NXP-NCI_LPCXpressoExample project shows how to easily interact with NCI based NXP's NFC Controller in order to provide NFC capability to an embedded system with no OS resources required. Obviously, this example can easily be ported in a system with RTOS support.

This project is based on LPCXpresso platform and runs on NXP LPC122x, LPC11xx, LPC82x or LPC11Uxx microcontroller driving the NFC Controller.

The present example demonstrates NDEF functionalities:

- R/W mode:
    - extract NDEF content from a remote NFC Forum tag
    - write NDEF content to NFC Forum Type 2 tag
    - authenticate/read/write MIFARE classic card
- P2P mode: exchange (in both way) NDEF content with remote P2P device
- Card emulation mode: expose NDEF content to a remote NFC reader

2. HW setup

2.1 LPC122x

To set up the project we use LPC1227 LPCXpresso board (http://www.embeddedartists.com/products/lpcxpresso/lpc1227_xpr.php)

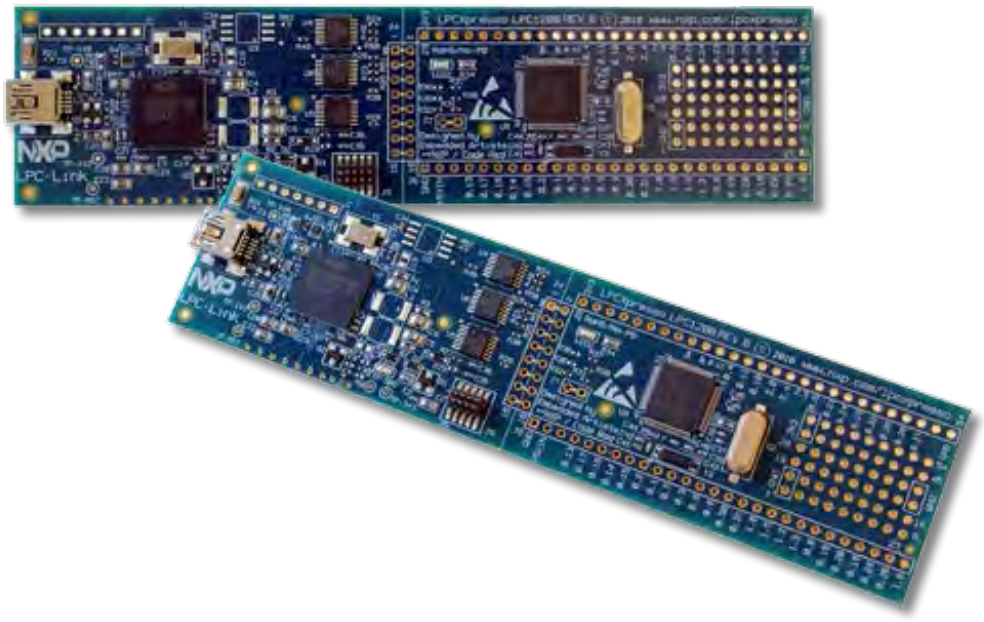

Fig 1. LPC1227 LPCXpresso board

The board must be connected to NFC controller board using the following instructions:

Table 1. HW setup instructions

LPC1227 board pin		NFC controller signal
V_{OUT} 3.3V	<->	V_{BAT}
V_{OUT} 3.3V	<->	$V_{DD(PAD)}$ / P_{VDD}
PIO0.10 / I2C-SCL	<->	I2CSCL
PIO0.11 / I2C-SDA	<->	I2CSDA
PIO0.21	<->	IRQ
PIO0.24	<->	VEN
GND	<->	GND

IRQ and VEN NFC controller signals can eventually connected to other MCU PIOs, in this case the example must be adapted in consequence (see 5.3.2).

2.2 LPC11xx

To set up the project we use LPC1114 LPCXpresso board (http://www.embeddedartists.com/products/lpcxpresso/lpc1114_xpr.php)

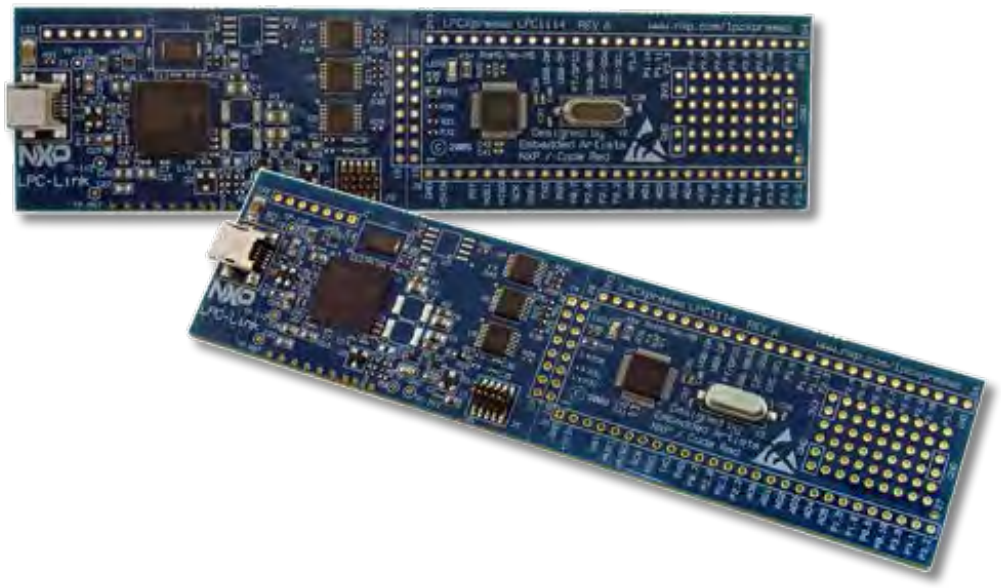

Fig 2. LPC1114 LPCXpresso board

The board must be connected to NFC controller board using the following instructions:

Table 2. HW setup instructions

LPC1114 board pin		NFC controller signal
V_{OUT} 3.3V	<->	V_{BAT}
V_{OUT} 3.3V	<->	$V_{DD(PAD)}$ / P_{VDD}
PIO0.4 / I2C-SCL	<->	I2CSCL
PIO0.5 / I2C-SDA	<->	I2CSDA
PIO3.4 (Note: wrong text in silkscreen, written 2.4)	<->	IRQ
PIO3.5 (Note: wrong text in silkscreen, written 2.5)	<->	VEN
GND	<->	GND

IRQ and VEN NFC controller signals can eventually connected to other MCU PIOs, in this case the example must be adapted in consequence (see 5.3.2).

2.3 LPC82x

To set up the project we use LPC824 LPCXpresso board (http://www.embeddedartists.com/products/lpcxpresso/lpc824_xpr.php).

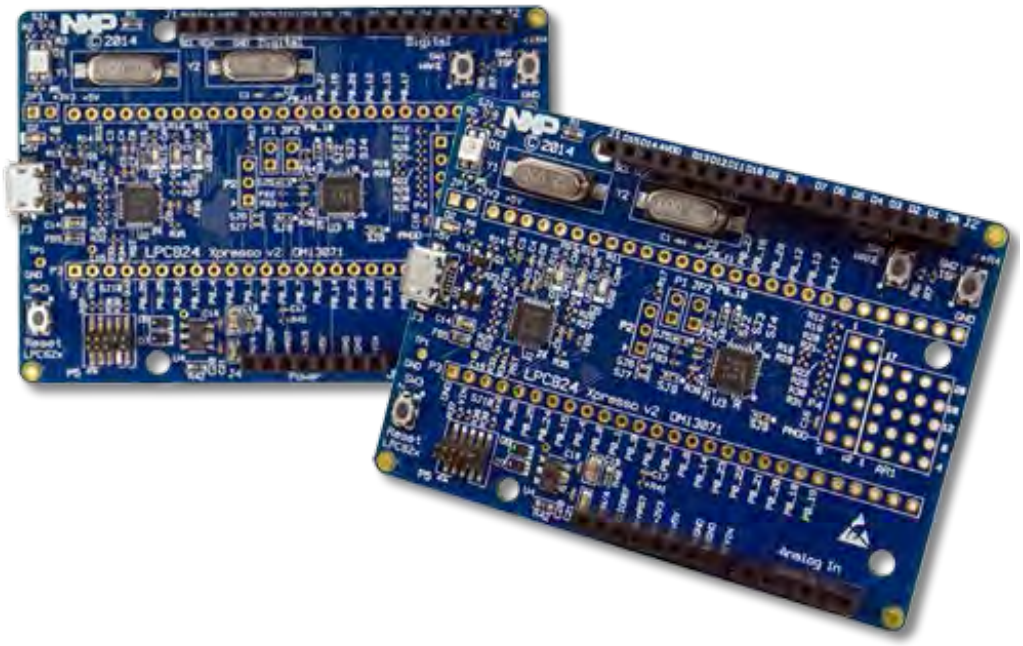

Fig 3. LPC824 LPCXpresso board

The board must be connected to NFC controller board using the following instructions:

Table 3. HW setup instructions

LPC1114 board pin		NFC controller signal
V_{OUT} 3.3V	<->	V_{BAT}
V_{OUT} 3.3V	<->	$V_{DD(PAD)}$ / P_{VDD}
PIO0.10 / I2C0_SCL	<->	I2CSCL
PIO0.11 / I2C0_SDA	<->	I2CSDA
PIO0.13	<->	IRQ
PIO0.17	<->	VEN
GND	<->	GND

IRQ and VEN NFC controller signals can eventually connected to other MCU PIOs, in this case the example must be adapted in consequence (see 5.3.2).

2.4 LPC11Uxx

To set up the project we use LPC11U24 LPCXpresso board (http://www.embeddedartists.com/products/lpcxpresso/lpc11U24_xpr.php)

.

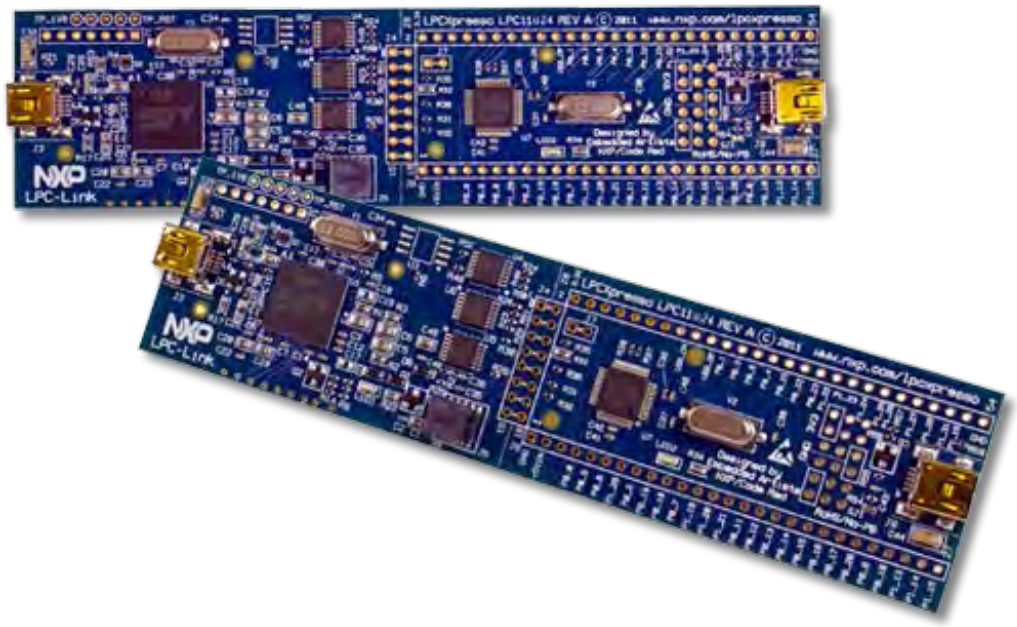

Fig 4. LPC11U24 LPCXpresso board

The board must be connected to NFC controller board using the following instructions:

Table 4. HW setup instructions

LPC1114 board pin		NFC controller signal
V_{OUT} 3.3V	<->	V_{BAT}
V_{OUT} 3.3V	<->	$V_{DD(PAD)}$ / P_{VDD}
PIO0.4 / I2C-SCL	<->	I2CSCL
PIO0.5 / I2C-SDA	<->	I2CSDA
PIO1.23	<->	IRQ
PIO1.24	<->	VEN
GND	<->	GND

IRQ and VEN NFC controller signals can eventually connected to other MCU PIOs, in this case the example must be adapted in consequence (see 5.3.2).

3. SW setup

LPCXpresso IDE can be downloaded from http://www.lpcware.com/lpcxpresso/home.

1. Create an empty workplace in LPCXpresso IDE:

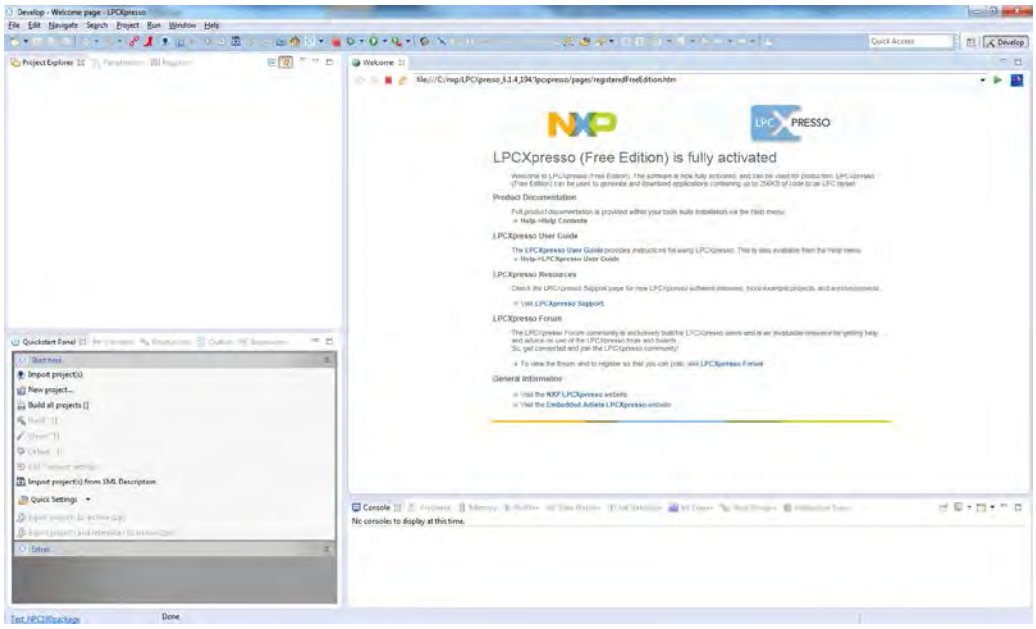

Fig 5. LPCXpresso IDE workplace

2. Import the project from the NXP-NCI_LPCXpressoExample zip file:

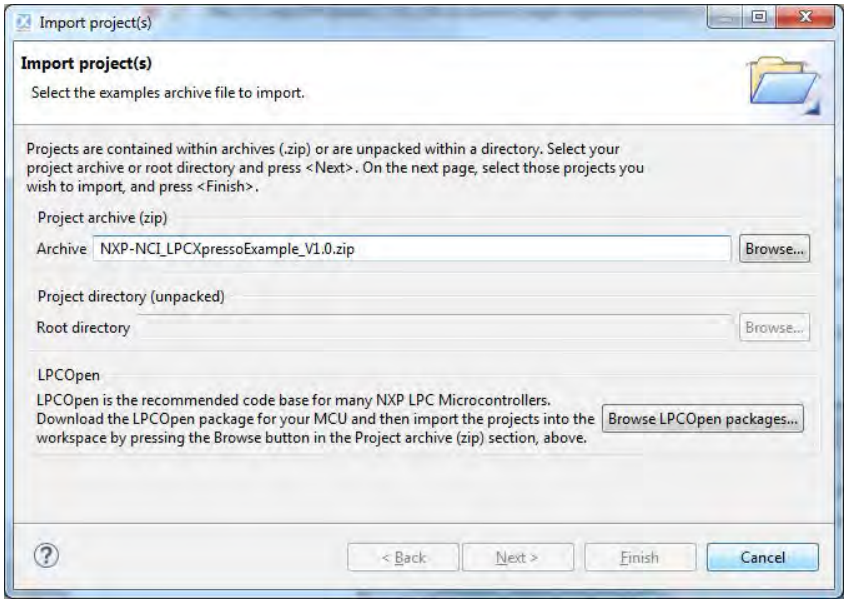

Fig 6. Importing project in LPCXpresso IDE

3. Build the project in « Debug » mode (clicking on the “hammer” icon):

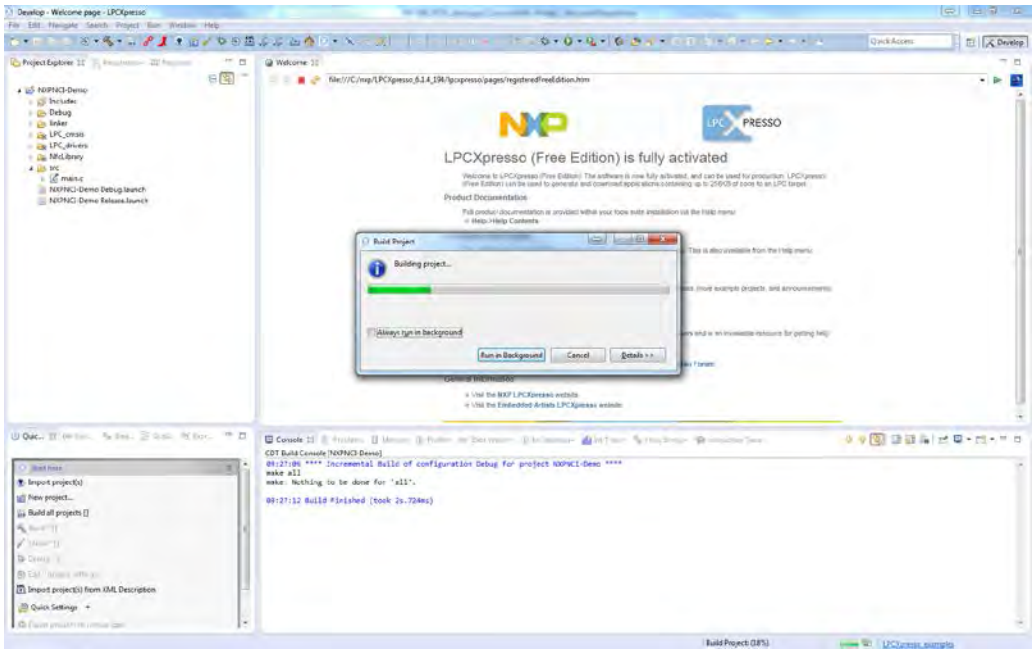

Fig 7. Building project in LPCXpresso IDE

4. Debug the project (clicking on the “bug” icon). This is flashing the LPC122x MCU and sets the execution pointer to the first instruction:

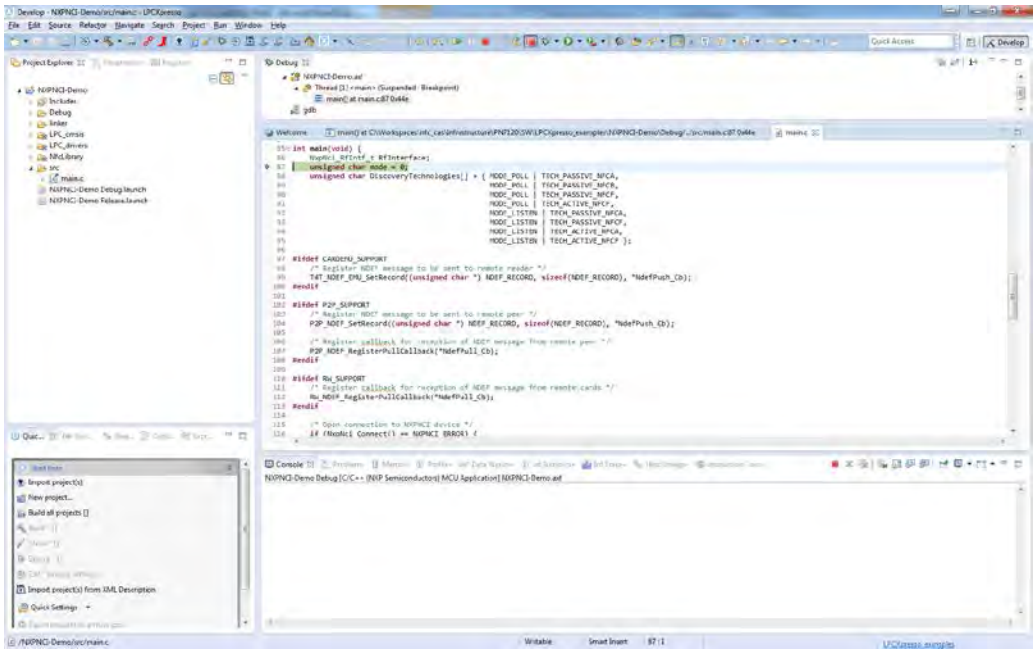

Fig 8. Building project in LPCXpresso IDE

5. Start the execution (clicking on « Resume » button or pressing f8). This launches the discovery and following message is displayed in the « console » window of LPCXpresso:

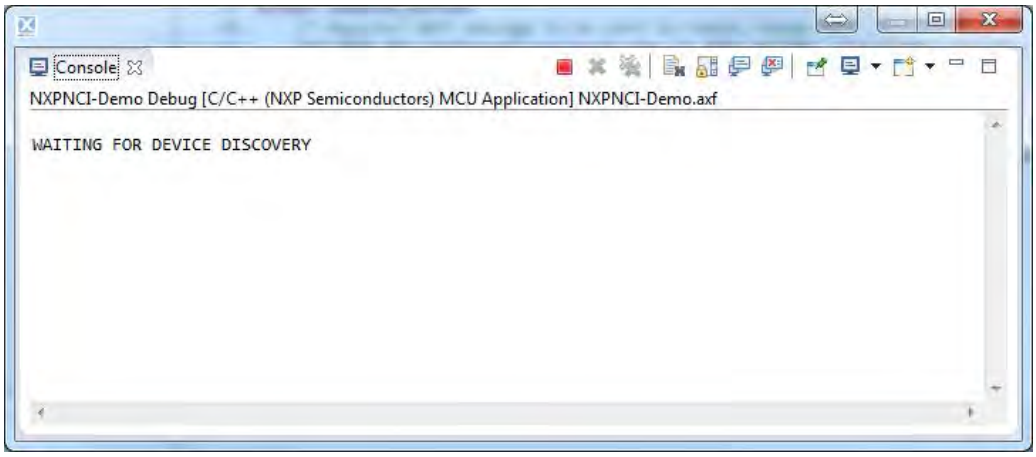

Fig 9. LPCXpresso console window starting project execution

4. Demo

4.1 R/W mode

Bringing a NFC Forum Tag containing NDEF content leads to a message display in the « console » window (here the tag contains Text type NDEF message “NXP Semiconductors”):

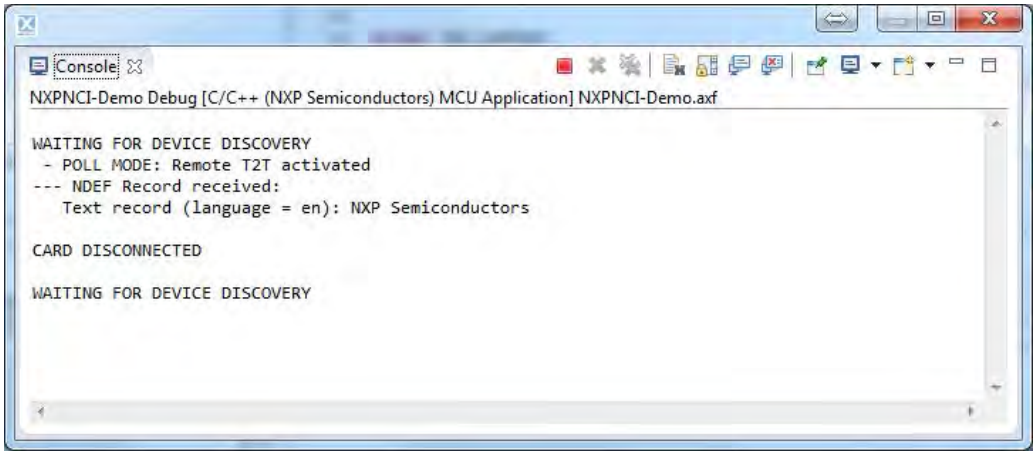

Fig 10. LPCXpresso console window when NDEF tag is read

Bringing a MIFARE classic card (with generic keys for sector 1) gives such result:

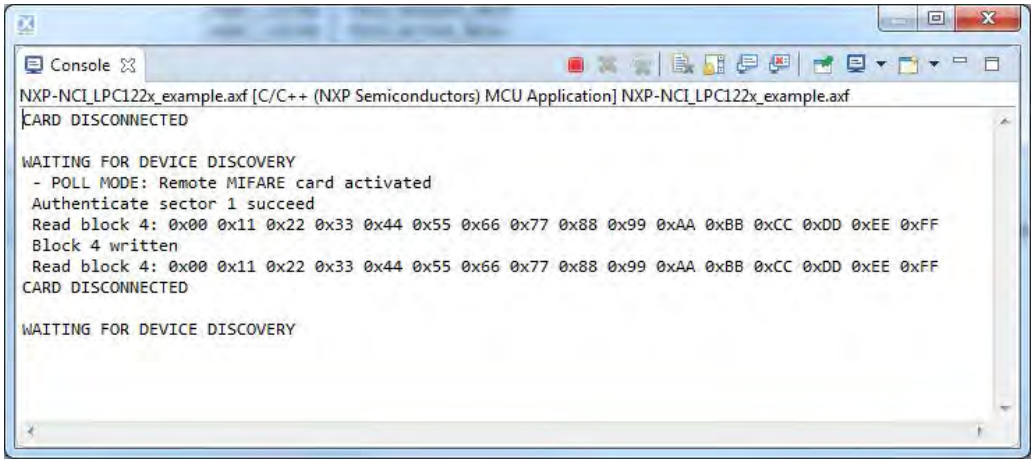

Fig 11. LPCXpresso console window when NDEF tag is read

4.2 P2P mode

Bringing a NFC android phone and « beaming » an URL gives such result:

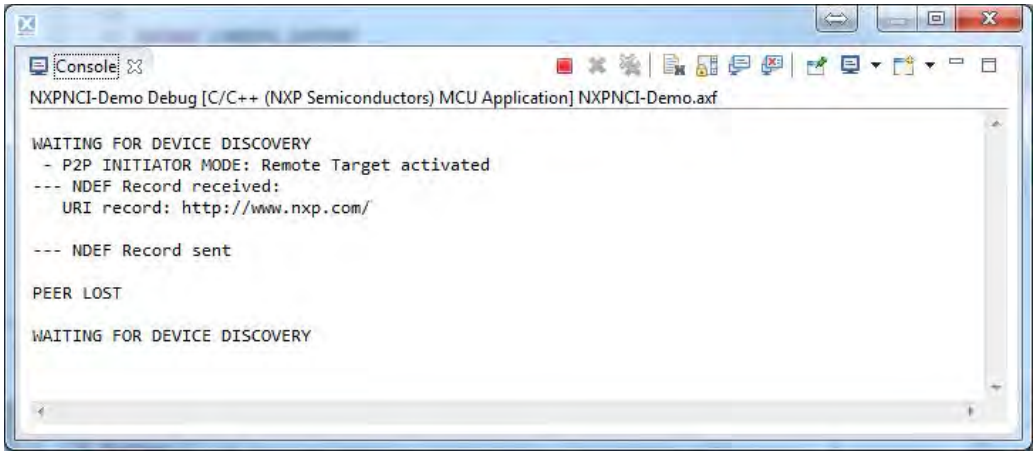

Fig 12. LPCXpresso console window when exchanging data with Android NFC phone

And simultaneously, the phone displays the received NDEF record from the NXP-NCI example project (NDEF Text type « Test » message).

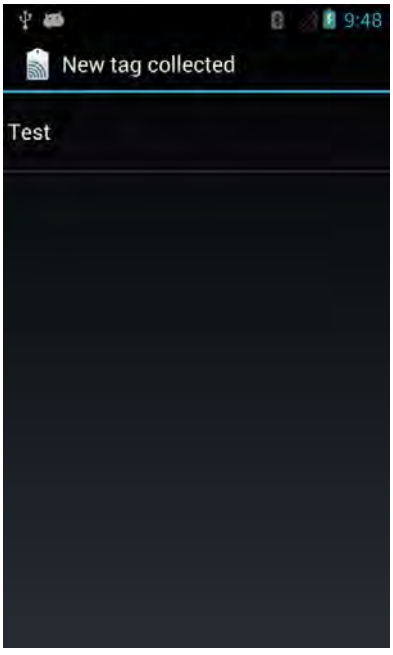

Fig 13. Android phone receiving NDEF message from NXP-NCI example project

4.3 Card Emulation mode

Bringing a NFC reader gives such result:

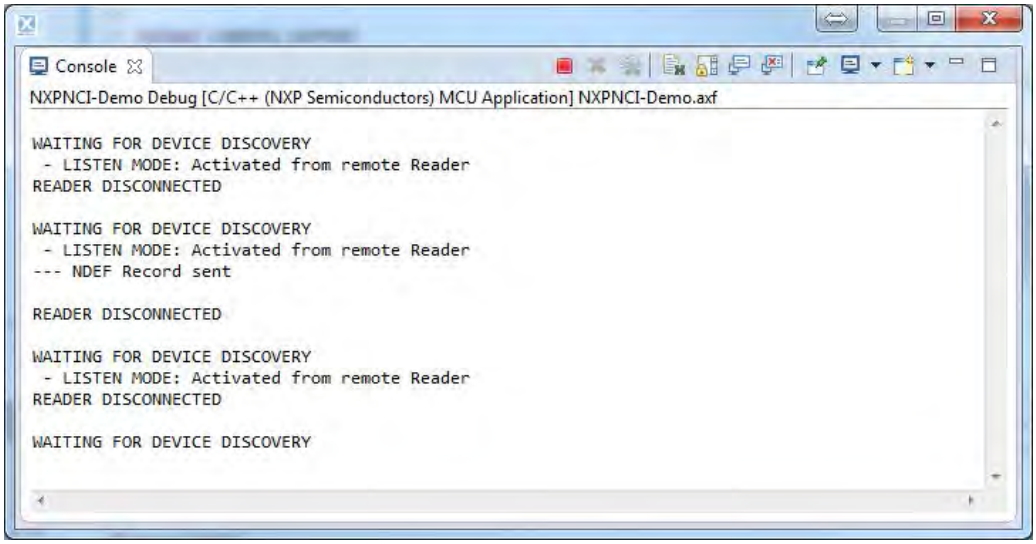

Fig 14. LPCXpresso console window when exchanging data with Android NFC phone

Note: To perform such scenario an Android phone can be used but then P2P mode must be disabled inside the NXP-NCI example project (see following chapter 5.2.1 for details) since otherwise the P2P communication is favored.

5. SW description

5.1 Architecture overview

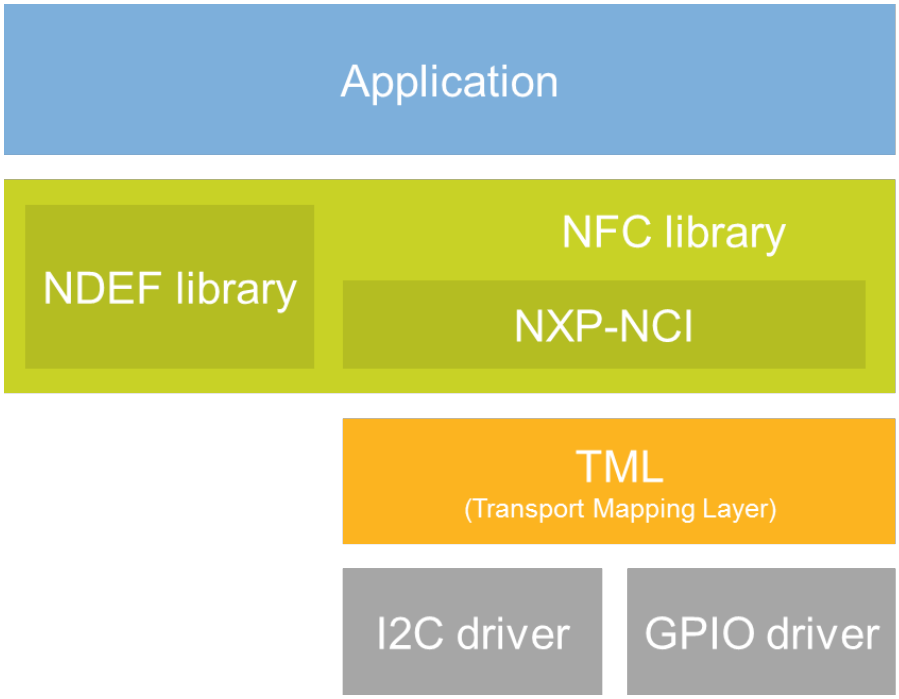

Fig 15. Example SW architecture overview

The Application consists in one “main.c” file. It uses NFC library API to register for NDEF functionalities and manage NXP-NCI processing.

{NXP-NCI} module offers high level NFC API:

- Connection and configuration of the NFC controller
- Start of the NFC discovery
- Wait for NFC discovery
- Process the NFC discovery

{NDEF library} module is composed of independent sub-module:

- {RW_NDEF} implements NDEF extraction from NFC Forum tags (all 4 NFC Forum defined tag types)
- {P2P_NDEF} implements NDEF data exchange with P2P device (over NFC Forum LLCP and SNEP protocols)
- {T4T_NDEF_emu} implements NDEF message exposure through card emulation (NFC Forum Type 4 Tag protocol)

{TML} module brings HW abstraction to NFC library (abstract how the connection to NFC controller IC is managed).

{I2C driver} module provides handle on the I2C communication with NFC Controller IC.

{GPIO driver} module provides handle on the NFC Controller VEN (Hard reset) and IRQ (interface interrupt) pins.

5.2 Stack size

Below is insights about the stack size, compiled on an ARM Cortex M0:

Table 5. Stack size from ARM Cortex M0

Module		Size (in bytes)	
Application		2758	
NDEF library		2732	
	RW_NDEF		112
	RW_NDEF_T1T		437
	RW_NDEF_T2T		325
	RW_NDEF_T3T		337
	RW_NDEF_T4T		646
	P2P_NDEF		473
	T4T_NDEF_emu		402
NXP-NCI		2108	
TML		218	
I2C driver (based on LPC82x implementation)		272	
GPIO driver (based on LPC82x implementation)		158	

The NFC library (NDEF library + NXP-NCI) is about 4840 bytes in its full configuration (RW, P2P and Card Emulation) but could be substantially reduced according to the targeted use case (for instance for T2T RW only support, size would be about 1455 bytes).

5.3 Porting recommendation for other MCUs

The present code example can be easily ported to any other target providing I2C bus master and GPIO capabilities (I2C master can eventually be implemented in SW via GPIO control in case of no HW support is provided by the target). The only modules requiring adaptations are I2C driver and GPIO driver (relates to how the target provides this support), others modules being platform agnostics.

5.4 Example customization

5.4.1 I2C address

NFC Controller I2C address is by default set to 0x28. It can be changed in the file *Drivers/inc/driver_config.h*:

```
#define NXPNCI_I2C_ADDR      0x28U
```

Fig 16. I2C address setting

5.4.2 PIOs assignment

In case a different connection is used than the one described in chapter 2, the file *Drivers/inc/driver_config.h* must reflect PIOs assignment:

```
#define PORT_IRQ              PORT0
#define PORT_VEN              PORT0
#define PIN_IRQ               13   // P0.13
#define PIN_VEN               17   // P0.17
```

Fig 17. I2C address setting

5.4.3 Discovery configuration

The discovery loop can be configured by setting “DiscoveryTechnologies” variable inside the “main” function.

By default all technologies (required for the aimed demonstration) are enabled: Passive NFCA, NFCB and NFCF as well as Active NFCF, in both POLL and LISTEN modes.

Simply adapt the discovery loop by commenting out the related technology you want to remove.

```
unsigned char DiscoveryTechnologies[] = { MODE_POLL | TECH_PASSIVE_NFCA,
                                          MODE_POLL | TECH_PASSIVE_NFCB,
                                          MODE_POLL | TECH_PASSIVE_NFCF,
                                          MODE_POLL | TECH_ACTIVE_NFCF,
                                          MODE_LISTEN | TECH_PASSIVE_NFCA,
                                          MODE_LISTEN | TECH_PASSIVE_NFCF,
                                          MODE_LISTEN | TECH_ACTIVE_NFCA,
                                          MODE_LISTEN | TECH_ACTIVE_NFCF};
```

Fig 18. Discovery configuration variable

5.4.4 Modes compile flags

Three compile flags exists in this SW example allowing to separately disable modes:

- “RW_SUPPORT”
- “P2P_SUPPORT”
- “CARDEMU_SUPPORT”

There are defined by default (all 3 modes supported).
To disable a mode just remove the related definition in the project properties:

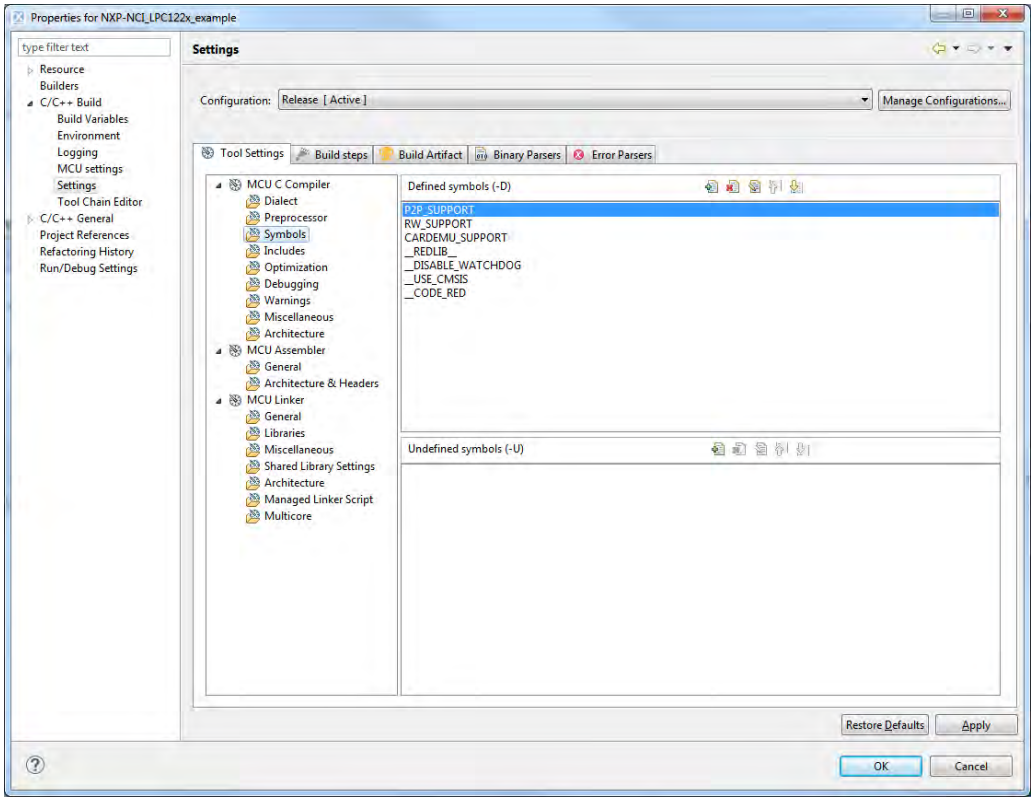

Fig 19. LPCXpresso project properties

6. Abbreviations

Abbr.	Meaning
AN	Application Note
EMU	Emulation (card emulation)
GND	Ground
GPIO	General Purpose Input Output
HW	Hardware
I²C	Inter-Integrated Circuit (serial data bus)
IC	Integrated Circuit
IO	Input / Output
IRQ	Interrupt Request
NDEF	NFC Data Exchange Format
NFC	Near Field Communication
NFCC	NFC Controller
OS	Operating System
P2P	Peer to peer
PCD	Proximity Coupling Device (Contactless reader)
PIO	Programmed Input/Output
PICC	Proximity Integrated Circuit Card (Contactless card)
RF	Radiofrequency
RTOS	Real Time Operating System
RST	Reset
RW	Reader/Writer
SW	Software
T1T	Type 1 Tag (NFC Forum tag types definition)
T2T	Type 2 Tag (NFC Forum tag types definition)
T3T	Type 3 Tag (NFC Forum tag types definition)
T4T	Type 4 Tag (NFC Forum tag types definition)
VEN	V ENable pin (NFCC Hard reset control)

7. Legal information

7.1 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.

7.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

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.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

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.

7.3 Licenses

Purchase of NXP ICs with NFC technology

Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards.

7.4 Trademarks

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

DESFire — is a trademark of NXP B.V.

I²C-bus — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

SmartMX — is a trademark of NXP B.V.

8. List of figures

9. Contents

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

Date of release: 6 July 2015
323112
Document identifier: AN11658