# AN11308

Quick Start Up Guide PNEV512B Board

Rev. 1.8 — 1 December 2014 255018

Application note COMPANY PUBLIC

#### Document information

Info	Content
Keywords	PN512, Blueboard, LPCXpresso, MCU, Code Red, eclipse, LPC1769, LPC1227, 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
1.8	20141201	Added a note about RAM limitation of LPC1227 in section 5.3
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: <u>http://www.nxp.com</u>

For sales office addresses, please send an email to: <a href="mailto:salesaddresses@nxp.com">salesaddresses@nxp.com</a>

AN11308

All information provided in this document is subject to legal disclaimers.

# 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 (**Chapter 5**).

The projects used in this documentation are:

- Communication with MIFARE Classic → Chapter 5.1
- Polling for Tags in the RF field → Chapter <u>5.2</u>
- Using the PN512 in Card Emulation mode (LPC1227 only) → Chapter 6
- Exemplary Peer to Peer implementation → Chapter <u>5.3</u>

Each project can also run on the LPCXpresso LPC1227 board (Chapter 7.3).

Application note COMPANY PUBLIC

# 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 1227 and 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. [12].

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 and FeliCa scheme
- Reader/Writer mode supporting ISO/IEC 14443B
- Card Operation mode supporting ISO/IEC 14443A/MIFARE 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:

- I<sup>2</sup>C
- SPI
- UART (optional, see Section 2.7)



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<sup>2</sup>C 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 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 128 KB instead of 512 KB and a Cortex M0 processor. For instructions on how to change the project settings to work with the LPC1227 see the description in section <u>7.2.</u>

## 2.5 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.



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

#### 2.6 Interesting points of measurement

On the PNEV512B demo board one can find test pads for measurement purposes. D7

- VCC
- GND •

D6

•

•

- D5
- ALE .
  - AUX1
  - AUX2

- SIGIN
- SIGOUT
- IRQ .

•

VMID



## 2.7 Preparing the Blueboard for the use with SPI or $I^2C$

The Blueboard is generally delivered in SPI configuration. To change the interface to I<sup>2</sup>C the four appropriate 0R0 resistors in the interface config section need to be resoldered to the I<sup>2</sup>C side of the solder jumpers. Also the two 0R0 resistors at A0 and A1 need to be changed.

#### Table 1. A0 and A1 interface configuration

Appropriate solder	iumpers (0R0 resiste	ors) for interface cor	nfiguration
Signal	Interface type		
	SPI	I <sup>2</sup> C	UART(optional)
A0	R28	R29	R29
A1	R32	R31	R32



To use the Blueboard in I<sup>2</sup>C 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 I<sup>2</sup>C-address can be configured either by software or by hardware. To set the I<sup>2</sup>Caddress by hardware the solder jumpers in the I<sup>2</sup>C 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.



To use the additional interfaces the interface selection signals A0 and A1 have to be configured to UART mode (see section <u>2.7</u>, <u>Table 1</u>).

#### 2.8.1 Configuring the additional interfaces

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

Table 2. Solde	r Jumpers for selecting the additional interfaces
Interface type	Resistors
USB	R38, R39
RS232	R40, R41, R42



#### Fig 10. 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  $\mu$ C 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 3.         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 4.	VCC power supply confi	guration	
Power su	upply	Resistors	
LPCXpre	sso Board	R3	
External \$	5VDC	R11	



### 2.9 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 7.7.

# 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.

b Monitors	6
Network adapters	
Ports (COM & LPT)	
Processors	
>	
Sound, video and game controllers	
Ge Storage controllers	
🛛 📲 System devices	
🖌 🏺 Universal Serial Bus controllers	Ê
Generic USB Hub	
🛱 Generic USB Hub	
🚽 🗍 Generic USB Hub	
🟺 Generic USB Hub	
🟺 Generic USB Hub	
🔤 🚽 🚽 🚽 🛶 🛶 Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C26	1
🛶 🟺 Intel(R) 6 Series/C200 Series Chipset Family USB Enhanced Host Controller - 1C2D	
🟺 USB Composite Device	
USB Composite Device	
USB Device with DFU Capabilities	
USB Root Hub	-
USB Root Hub	1

# 4. Managing the Demo Reader project with LPCXpresso IDE

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

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

This tool can freely be downloaded from the LPCXpresso website [1]. Before one can download the software, it is necessary to create an account. Creating an account is absolutely free.

## 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.

Debug driver selection		
Please select which drivers to instal	l.	LPCXPRES
Windows may issue warnings when If prompted, please allow the driver Drivers are located in the "Drivers"	installing these drivers. rs to install. directory of the installation	
NXP Debug drivers		
1 4_194		~
ttp://www.nxp.com/lpcxpresso	< Back	Next > Cancel

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.



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

ĺ		
	LPCXpresso	
	LPCXpresso	
	License type: UNREGISTERED (Code=1) Debug limit: 8k	
	Activation code: not available	
	To activate a license, please select Help->Activate LPCXPresso (Free Edition)	
	License Restriction: An UNREGISTERED copy of LPCXpresso (Free Edition) may only be used for evaluation purposes. Register and activate to remove this restriction.	
	OK Cancel	
Fig 15. LPC	Xpresso 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  $\rightarrow$  Product activation  $\rightarrow$  Create Serial number and Activate...

AN11308



Create serial number and register Select OK to visit the registration website where you can	X
register your product and receive an Activation Code.	UPCKPRESSO
Serial number 11OV-LZM5-G5GW-HRH3-NUH2-M4D1-ORAZ-A511-E1G1-LVHT	
Open in external browser	
Copy Serial Number to clipboard	
	Canad
UK	Cancer

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.



Once the activation code arrives please open the activation window by pointing to Help  $\rightarrow$  Product activation  $\rightarrow$  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.



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

Select the examples archive file	e to import.	
Projects are contained within a project archive or root director wish to import, and press <fini< th=""><th>rchives (.zip) or are unpacked within a directory. Select your y and press <next>. On the next page, select those projects y sh&gt;.</next></th><th>rou</th></fini<>	rchives (.zip) or are unpacked within a directory. Select your y and press <next>. On the next page, select those projects y sh&gt;.</next>	rou
Project archive (zip)		-
Archive C:\Projects\PNEV51	2B-LPC1227-LPC1769-Classic-Polling-SPI-I2C.zip	Browse
Project directory (unpacked)		
Root directory		Browse,
LPCOpen LPCOpen is the recommende	d code base for many NXP LPC Microcontrollers.	
Download the LPCOpen pack workspace by pressing the Bri	age for your MCU and then import the projects into the <u>Bro</u> owse button in the Project archive (zip) section, above.	owse LPCOpen packages
0	< Back Next > E	inish Cancel

Browse the desired package and click "Next".



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 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 PNEV512B demo board is connected to the computer.

**Application note** 

COMPANY PUBLIC



Choose the desired project 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 7.2 and 7.3 for further information.

Develop - RC003-P2P-Initiator/Example_RC003_Lpc.	1/xx_P2P_Initiator.c - LPCXpresso	
e Edit Source Refactor Navigate Search F	roject Run Window Help	
•    @ =   @ • % • 🗟 🔗 📕 🤋	- 🖞 🗸 🕫 🎘 🕼 🕼 🕼 🖉 🖉 📲 🚳 🖌 🖸 🗸 🚱 🖧 📲 🌾 🖉 🖉 😵 🖉 🖉 🖉	
·• ≈ • • • • • •   •	Quick Access	🖽 🔀 Devel
Proje 😒 😤 Penp IIII Regis 🗖 🗖	system_LPC17xxc DE Example_Rc663_Lpc17xx_P2P_Initiator.c 23	-
<ul> <li>E Classic</li> <li>E Lpcl227</li> <li>E Lpcl227</li> <li>E Lpcl2769</li> <li>E MVP-Reader-Lib</li> <li>E Duttor</li> </ul>	<pre>1033 status = phOsal_Timer_Wait(&amp;osal, 1, 1000); 1034 CHECK_SUCCESS(status); 1035 status = phnalw_FieldReset(&amp;hal); 1036 CHECK_SUCCESS(status); 1037 } 1038 }</pre>	
<ul> <li>B Includes</li> <li>C DebugLPC1769</li> <li>C cards.c</li> </ul>	10400/**********************************	
Polling.c	1944# int main (void)	
Polling DebugLPC1227.launch	1945 phStatus t status;	6
Polling DebugLPC1/09.launch	1047	
Polling Releasel PC1769 Jaunch	1848 GPID_Init(); 1848 #ifedef TISA	
S RC663-P2P-Initiator	1059 /* Set LED port pin to output */	
-	1051 Set_Port();	
	1052 #endif 1053 /* Ensure, that the LED is off */ 1054 (_edOff():	
Q XX (M+V @ B HE O 6%)E = 🗆	1855 1856⊜ /* Set the interface link for the 1857 * internal chip communication */	
🛛 Start here 🏦 📩	1058 Set_Interface_Link(); 1059	
Import project(s)	1060 /* Perform a hardware reset */	
New project	1961 Reset reader device():	
nen projectili		
Build all projects [DebugLPC1769]	🖳 Console 🙁 🤮 Problems 📋 Memory 🛞 Profile+ 🛷 Data Watch+ 🕕 Int Statistics+ 🎦 Int Trace+ 🔖 Host Strings+ 🚳 In	struction Trace
Build 'Polling' [DebugLPC1769]		
Clean 'Polling' [DebugLPC1769]	CDT Build Console (Pollina)	
Debug 'Polling' [Debug] PC1769]	7017/makeno-print-urrectory post-outin	
	arm-none-eabi-size "Polling.axf": # arm-none-eabi-obicopy -0 binary "Polling.axf" "Polling.bin" : checksu	um -p LPC1769 -d "P
Import project(s) from XML Description	text data bss dec hex filename 55976 4 6352 62332 f37c Polling.axf	
💆 Quick Settings 👻		
Export projects to archive (zip)	12:02:45 Build Finished (took 2s.50ms)	,
(P LPC1769 (Polling)		

#### Fig 23. Run the project

**Application note** 

**COMPANY PUBLIC** 

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

a	Initializing LPC-Link (HII	D)	
-			
-			
			Cancel
		_	1

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

8-8-00	Quick Access million	Develo
Proje 23 V. Prop. III fogs.	Sp Debug 20 ● The Poling Debug(PC1769 [C/C++ (ROP Semiconductors) MCU Application] ● 愛 Poling and	
CMSS     Debug     Debug     Detrores     Detrores	Porter [1] (mains (Running: User Request)     J gdb      di system [196] Tour      Di Europie (Eddi Justice ) prime Config =      Di Polling =      23	• •
Bernstein      B	921 9220 /*** Main Function 924 925 9256 int main (void) 927 {	
s (g cardisc s (g Pollingsc + ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	#28     phtstus_t status;       929     948       931     ** Set the interface link for the       931     ** Internal chip communication */       932     Set_Interface_Link();       933     #indeef TUSA       935     /* Set LED point pln to output */	
😒 Start here 🙃 🍝	936 Set_Port()) 937 #end() 938 /* Ensure, that the LEO is off */ 939 Leedoff();	
New project     Build all projects [Debug[PC1769]	940 941 /* Perform a hardware reset */ 942 / Reset reader device();	
Clean 'Polling' [DebugLPC1769]	Console 22 19 Problems    Memory (P) Profiles do Data Watchs (L) Int Statistics - Taking Int Traces & Host Strings- & Instruction Trace	-
Debug 'Polling' [DebugLPC1769]     Edit 'Polling' project settings	Polling Debug/PC1769 (C/C++ (NXP Semiconductors) MCU Application) Polling.ad	• 13
Import project(s) from XML Description  Quick Settings	Type A T2 tag detected Cond 1: UD: 46 AB E3 A0 Atga : 44 00 Sak : 40	
Deport projects to archive (zip)	NEFARE Classic detected	-
NXP LPC1769 (Polling)		

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

Anny Debug Petzz/(e/e+ (NAP Semiconductors) Neo Appication Fommy.axi Atqa : 04 00 Sak : 08 MIFARE Classic detected
Type A T2 tag detected Card 1: UID: 46 48 E3 A0 Atqa : 04 00 Sak : 08 MIFARE Classic detected
4

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

<u>Run</u> Project <u>W</u> indo	iw <u>H</u> elp			
林 <u>路</u> •	C 🖻 🏊 🖆 🖾 🔾 , ()	. 🖗 🚵 🔹 🔹 🖏 👘	* 15+0+	🖹 🔀 Develop
Debug 😫		· · · · · · · · · · · · · · · · · · ·		· · · · · · · · · · · · · · · · · · ·
PN512-Polling Debu	ug [C/C++ MCU Application]		$\sim$	
MCU GDB Debu	gger (02.11.12 14:06)			
P Thread [1] (F	lunning)			
📓 arm-none-eabi-	gdb (02.11.12 14:05)			
E. CAlleral and	Annual and Annual DAIE10 Dalling	D-L DNIELD D-III (02111214-06)		

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.
P	Step over C/C++ line.
Ð	Step into a function.
	Stop the debugger.
00	Pause execution of the running
	program.
i⇒	Instruction stepping mode
	(disassembly).

Fig 28. Debug Buttons

# 5. Associated Projects

All example projects are available for download at the product page of the PNEV512B [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.

Every example project can either run on an LPCXpresso LPC1769 or on an LPCXpresso LPC1227. Information about how to switch between microcontrollers can be found in section 7.2 and 7.3.

## 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 examples

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

Every P2P example can be configured to act as a server or as a client.

**Note:** None of the SNEP – Server examples can be executed on the LPCXpresso LPC1227 because of its limited RAM.

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 PNEV512B demo board [8].

# 6. Additional projects

Besides examples based on the NFC Reader Library, additionally we provide functionalities that are not yet supported by this library. Components of the following packages cannot be exchanged with packages from the examples based on the NFC Reader Library.

## 6.1 Tag Type 2 and Type 4 Card Emulation

**Note:** This example only works with the LPCXpresso LPC1227 development board.

**Note:** Card Emulation mode is currently not supported by the NFC Reader Library v3.010. Therefore the card emulation project is based on a modified version of the NXP Reader Library v2.1.

This project is being distributed within a different installer file and needs to be downloaded separately from the demo board web site [8].

The PN512 supports 4 different operating modes:

- Reader/Writer mode supporting ISO/IEC 14443A/MIFARE and FeliCa compliant scheme
- Reader/Writer mode supporting ISO/IEC 14443B
- Card Operation mode supporting ISO/IEC 14443A/MIFARE and FeliCa compliant scheme
- NFCIP-1 mode

The card operation mode is passive mode, in which the PN512 does not generate an RF field but acts as a card that modulates the field for communication with the reader. The IC only supports parts of the ISO/IEC 14443-protocols, the ISO/IEC 14443-4 as well as the ISO/IEC 7816-4 commands need to be provided by the Microcontroller.

A specification to store data for any kind of service and application is specified in the NFC Forum and it is called NFC Data Exchange Format. Storing NDEF formatted data inside contactless card products as mapping models as well as the management of NFC forum device as a specific platform such as a NFC Forum Type 4 Tag are defined in [7]. The following project shows an exemplary implementation of a Tag 4 Type Card on the PN512. Therefore one NDEF File and one capability container (CC) file, with ISO file identifier (ISO FID) equal to E103h, are presented to the reader.

### 6.1.1 Configuration of the example project

In order to change some of the possible options before compiling the project, the file "src\nxprdlib\intfs\phCardEmu\_Options.h" should be edited. This file contains toggles to enable/disable the T2T and T4T functionality, memory sizes, as well as the pin numbers in which the PN512 chip is connected to the LPC1227. Further options are also indicated on the file.

Before flashing the modified project it is mandatory to perform a **Mass erase** on the flash memory of the LPC1227. This can be achieved by clicking the **Program Flash** icon in the LPCXpresso IDE.



In the Program Flash wizard the Mass erase feature can be found in the **Erase flash memory** tab. After activating the **Mass erase** algorithm click OK to perform the procedure.

Program target flash: LPC11xx (NXP LPC1227/3)	00
	- Citatan
Options 🐨 Display progress log	Reopen on completion
Reset target on completion	Repeat on completion
🔲 Run flash command and copy to clipboard	Just copy flash command to clipboard
Connection Options Use JTAG interface LPC-Link (HID) Options Speed 250 Flash Driver Flash Driver LPC11_12_13_128K_8K.cfx	+ Browse,
Program flash memory Erase flash memory	
Algorithm	
Mass erase	Erase by page
	OK Cancel

### 6.1.2 Changing the NDEF message

The predefined NDEF message can be changed in the following files:

• T4T: ../src/nxprdlib/comps/phce7816p4T\_Apps/phce7816p4T\_T4T\_Const.c

nain.c	🖻 phce7816p4T_T4T_Const.c 🕱
1	<pre>#include <phce7816p4t_t4t.h></phce7816p4t_t4t.h></pre>
2	
З	#define NORMAL 0
4	#define MAX_SMARTPOSTER 1
5	#define BIG_MIME_IMAGE 2
6	#define MAX_VCARD 3
7	
8	Thefine this to one of the above for NDEF predefined content
9	#define T4T_NDEFFILE_PREDEF NORMAL
10	
11	// This is the data structure returned by the GetData on tag GetVersion
12	const byte ILV_VERSION[] = {
15	PI GEIDAIA ING VERSION, P2 GEIDAIA ING VERSION, (BYTE)0X16, // neader: Version informat
14	(byte) 1, (byte) 5, // year in Astil
10	(byte) 0, (byte) 1, // month in Ascill
17	(byte) 1, (byte) 5, 77 day in Astii
1.8	(byte) , (byte)'N' (byte)'X' (byte)'D'
10	(byte) m, (byte) A, (byte) F,
Fig 3	1. Change NDEF content for T4T

• T2T: ../src/nxprdlib/comps/phceT2TCmdHdl/src/Sw/phceT2T\_Const.c

尾 mair	n.c 🔀 phceT2T_Const.c 🕱
1	<b>≉include</b> "phceT2T_Sw.h"
2	
З	#define NORMAL 0
4	#define MAX_SMARTPOSTER 1
5	Hadina MAY_UKADD 2
7	HUETTIE MAA_VCARD 5
8	<pre>// define this to one of the above for NDEF predefined content</pre>
9	#define T4T NDEFETLE_PREDEF NORMAL
10	
11	<pre>const uint8_t T2T_LOCK_CC_DEF[16] = {</pre>
12	0x00, 0x00, 0x00, 0x00, // internal bytes
14	axaa, axaa, axaa, axaa, // internal bytes axaa axaa axaa axaa axaa axaa // internal bytes AND
15	// static lock by the // static lock by the //
16	// сс
17	<b>0xE1, 0x10, 0x00, 0x00</b> // 3rd byte - data mem
18	};
19	
Fig 32	Change NDEF content for T2T

## 6.1.3 Program flow



The first blocks describe the initialization of the necessary layers and components independent of the card emulation. Depending which tag type is used the appropriate component will then be initialized. If Tag Type 2 and Tag Type 4 are enabled, both are

AN11308

being initialized. After applying the protocol settings, the PN512 waits for a successful activation of the card side. According to the first command (RATS – yes or no) after the card activation the appropriate tag type component starts.

# 7. Supplementary Notes

## 7.1 Software architecture

The software of the reference reader is based on the NFC Reader Library [5]. It intends to be simple, modular, easily readable and quickly portable by all the customers. This philosophy is reflected in its architecture which is divided into 4 layers:

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

										NFC Apps
									Sample P2 application	р 1
				5.0.00					NF	C P2P Package
Ca	ller (Applica	ation, N	liddleware	e, Embedd	ed System	,)	17.000		SNEP 1.0	NFC Protocols
		Appl	lication La	yer (AL)			NF	C Activity		
MIFARE Classic	MIFARE Ult Ultraligh Ultralight	ralight It C EV1	MIFARE DESFire (partly)	FeliCa (partly)	Jewel / Topaz	NFC Forum Tag Type Operations	D	iscovery Loop	LLCP 1.1	Link Layer
				Pr	otocol Abs	traction Laye	r (PAL)	-		
ISO/IEC 14443-3A	ISO/IEC 14443-3B	ISO/IE 14443-/	C ISO/II 4A 14443	EC MIFA	ARE Comp prot	iCa ISO/IEC pliant Pass ocol Initia	18092 live ator	ISO/IEC 18092 Target		
				Har	dware Abs	straction Laye	er (HAL	)		
	Generic									
Callback	RC663	PN512/	RC523							
	-			-	Bus Abstra	ction Layer (	BAL)			
	Generic									
	LPC1769 SP	LPC17	69 I2C							
Stub										
Stub					Com	imon Layer				
Stub										

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 PNEV512B 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

AN11308

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 PN512.

#### 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 section  $\frac{5}{2}$  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

AN11308

• 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 LPC1227.

> }≥ RCO       	New  Go Into Go Into Open in New Window Copy Ctrl+C Paste Ctrl+V Delete Delete Source Move Rename Export Build Project Clean Project Clean Project Close Uncleated Projects Build Configurations	<pre> system_LPC17xx.c Example_Rc663_Lpc17xx_P2P_Initiator.c phh  p20 p22 ** Main Function p23 ** Main Function p24 p25 phStatus_t status; p26 int main (void) p27 { p28 phStatus_t status; p39 /* Set the interface link for the p31 * internal chip communication */ p32 Set_Interface_Link(); p34 #ifindef TUSA p35 /* Set LED port pin to output */ p36 Set_Port(); p37 #endif p38 /* Ensure, that the LED is off */ p39 LedOff(); p39 LedOff(); p30 Ledoff();</pre>
Import     Import     Import	Make Targets  Index	Manage         Image: Classical Control (Classical Control (Classical Classical C
🚮 Build al ≼ Build 'P 👋	Convert To Manage DSF Host Strings	Clean All 4 ReleaseLPC1769 (Release build for LPC1769) Build Selected itialize the Stub timers component */

**Note:** When switching from one MCU to another, please take care to also switch to the correct MCU setting within the LPCXpresso IDE. See chapter 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  $\rightarrow$  choose properties (at the bottom)
- C/C++ build → MCU settings → expand desired LPC1xxx MCU group → choose the correct microcontroller → click OK

type filter text	MCU settings
<ul> <li>Resource Builders</li> <li>C/C++ Build Build Variables Environment Logging MCU settings Settings</li> </ul>	Target
	NXP LPC1227/301
	LPC1754
	LPC1756
	LPC1758
	LPC1759
	LPC1763
Tool Chain Editor	LPC1764
Decised Defense	LPC1765
Project References	LPC1766
Run/Debug Settings	LPC1767
	LPC1768
	LPC1769
	LPC177x 8x

## 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.

**Application note** 

**COMPANY PUBLIC** 



### 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 Preparing the projects for the use of the Blueboard in I<sup>2</sup>C configuration

To use the projects in I<sup>2</sup>C 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.

### 7.6 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:

Right click on the project and choose Launch Configurations 
 → Edit current...

AN11308

AN11308



Fig 38. Open Current Launch Configuration

- 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.

Create, manage, and run con	figurations
type filter text	Name: Polling DebugLPC1769
C/C++ (NXP Semiconduct Classic DebugLPC1227 Classic DebugLPC1269 Classic DebugLPC1769 Classic DebugLPC1277 C Polling DebugLPC1769	Stop on startup at main Force hardwa
C Polling ReleaseLPC1227	Target configuration Main

## 7.7 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.

## 7.7.1 Preparing the hardware



- Connect the PNEV512B Blueboard to the BluePi adaptor.
- Connect the adaptor to the Raspberry Pi



## 8. References

#### [1] LPCXpresso website http://www.lpcware.com/lpcxpresso/download

- [2] PN512 product information and data sheet http://www.nxp.com/products/interface\_and\_connectivity/nfc\_contactless\_reader\_ic s/series/PN512.html
- [3] LPC176x/5x User manual http://www.nxp.com/documents/user\_manual/UM10360.pdf
- [4] Multipoint Connectors we used: Grid Dimension: 2.54mm, at least 27 pins <u>http://www.conrad.at/ce/de/product/741119/STIFTLEISTE</u> and http://www.conrad.at/ce/de/product/736427/BUCHSENLEISTE-EINREIHIG
- [5] **Direct link to the NFC Reader Library** Not yet available
- [6] Using the PNEV512B in Card Emulation mode http://www.nxp.com/demoboard/PNEV512B.html
- [7] TYPE 4 TAG: NFC Forum, Type 4 Tag Operation Specification, Version 1.0, March 13, 2007 http://www.nfc-forum.org/specs
- [8] PNEV512B demo board site http://www.nxp.com/demoboard/PNEV512B.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 http://www.nfc-forum.org/specs/spec\_license
- [11] **EMV** The table of card types and their matching AIDs are available on <u>http://www.en.wikipedia.org/wiki/EMV</u>
- [12] LPCXpresso LPC1769 development board http://www.nxp.com/demoboard/OM13000.html
- [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.

Application note

COMPANY PUBLIC

# 10. List of figures

Picture of PNEV512B demo board4
Picture of solder bridges in default
configuration5
Picture of LPCXpresso LPC1769 development
board6
Multipoint Connectors we used6
LPCXpresso with the Multipoint Connectors7
Connect the two boards7
Interesting points of measurement7
Blueboard in SPI configuration8
Additional interfaces9
Solder jumpers for additional interface
configuration9
Solder jumper for VCC power supply
configuration10
Enumeration of the LPCXpresso Board in
Device Manager Window11
LPCXpresso installation NXP Debug drivers12
Windows Security dialog13
LPCXpresso IDE13
Product activation14
Product activation14
Product activation15
Importing a project into the LPCXpresso IDE16
Importing a project into the LPCXpresso IDE16

Fig 21.	Importing a project into the LPCXpresso IDE	17
Fig 22.	Run the project	18
Fig 23.	Run the project	18
Fig 24.	Run the project	19
Fig 25.	Run the project	19
Fig 26.	Run the project	20
Fig 27.	Stop the project	20
Fig 28.	Debug Buttons	20
Fig 29.	Program Flash icon in LPCXpresso IDE	23
Fig 30.	Program Flash wizard with Erase flash	
	memory tab	23
Fig 31.	Change NDEF content for T4T	24
Fig 32.	Change NDEF content for T2T	24
Fig 33.	Program flow of the Card Emulation	
	application	25
Fig 34.	Architecture of the NFC Reader Library	27
Fig 35.	Select the build configuration	29
Fig 36.	Selection of the LPC1769 MCU	30
Fig 37.	Select the level of optimization	31
Fig 38.	Open Current Launch Configuration	32
Fig 39.	BluePi adaptor	33
Fig 40.	PNEV512B Blueboard connected to the	
	Raspberry Pi	33

Application note COMPANY PUBLIC

# 11. Contents

1.	Introduction3
2.	Hardware overview of the Demo Reader4
2.1	PNEV512B demo board4
2.2	CE certification of the Blueboard5
2.3	LPCXpresso LPC1769 development board5
2.4	Alternative to the LPCXpresso LPC17696
2.5	Preparation of the hardware6
2.6	Interesting points of measurement7
2.7	Preparing the Blueboard for the use with SPI or I <sup>2</sup> C8
2.8	Optional interfaces and power supply8
2.8.1	Configuring the additional interfaces9
2.8.2	Configuring the power supply10
2.9	Other supported system architectures10
3.	Installation of the LPCXpresso Board11
4.	Managing the Demo Reader project with
	LPCXpresso IDE12
4.1	Installation of LPCXpresso IDE12
4.2	Extraction of the demo reader project15
4.3	Run the project17
5.	Associated Projects21
5.1	Communication with MIFARE Classic
5.2	Polling21
5.3	Peer to Peer examples21
6.	Additional projects22
6.1	Tag Type 2 and Type 4 Card Emulation22
6.1.1	Configuration of the example project22
6.1.2	Changing the NDEF message24
6.1.3	Program flow25
7.	Supplementary Notes27
7.1	Software architecture27
7.1.1	Bus abstraction layer27
7.1.2	Hardware abstraction layer
7.1.3	Protocol abstraction layer
7.1.4	Application layer
7.2	Build configuration
7.3	Setting the MCU29
7.4	Level of compiler optimization
7.4.1	Optimization issues
1.5	Preparing the projects for the USe of the
76	Blueboard In PC configuration
0.1	startup
77	startup
1.1 771	Dreparing the hardware
1.1.1	Freparing the haluware

8.	References	34
9.	Legal information	35
9.1	Definitions	35
9.2	Disclaimers	35
9.3	Licenses	35
9.4	Trademarks	35
10.	List of figures	36
11.	Contents	37

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

#### © NXP Semiconductors N.V. 2014.

#### All rights reserved.

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

> Date of release: 1 December 2014 255018 Document identifier: AN11308

# **X-ON Electronics**

Largest Supplier of Electrical and Electronic Components

Click to view similar products for nxp manufacturer:

Other Similar products are found below :

MC13211R2 PCA9518PW,112 LFSTBEB865X MC33399PEFR2 PCA9551PW,112 MC34825EPR2 CBTW28DD14AETJ PCF8583P MC68340AB16E MC8640DTVJ1250HE EVBCRTOUCH MC9S08PT16AVLC MC9S08PT8AVTG MC9S08SH32CTL MCF54415CMJ250 MCIMX6Q-SDB MCIMX6SX-SDB 74ALVC125BQ,115 74HC4050N 74HC4514N MK21FN1M0AVLQ12 MKV30F128VFM10 FRDM-K66F FRDM-KW40Z FRDM-MC-LVBLDC PESD18VF1BSFYL PMF63UNEX PSMN4R0-60YS,115 HEF4028BPN RAPPID-567XFSW MPC565MVR56 MPC574XG-176DS MPC860PCVR66D4 BT137-600E BT139X-600.127 BUK7628-100A118 BUK765R0-100E.118 BZT52H-B9V1.115 BZV85-C3V9.113 BZX79-C47.113 P5020NSE7VNB S12ZVML12EVBLIN SCC2692AC1N40 LPC1785FBD208K LPC2124FBD64/01 LS1020ASN7KQB LS1020AXN7HNB LS1020AXN7KQB LS1043ASE7PQA T1023RDB-PC