# AN11908

NFC Reader Library for FRDM-K82F Board Installation guidelines

Rev. 1.3 — 18 April 2018 407213

Application note COMPANY PUBLIC

#### **Document information**

Info	Content
Keywords	NFC, Reader Library, MCUXpresso, PN5180, CLRC663, CLEV6630B, Kinetis K82, FRDM-K82F
Abstract	This document describes how to use NXP's NFC Reader Library with a NXP Freedom K82F board using MCUXpresso.



#### **Revision history**

Rev	Date	Description
1.3	20180418	Editorial updates
1.2	20170511	MCUXpresso added
1.1	20170105	Example list updated, ICODE support defined, CLEV6630B description updated
1.0	20161110	First release

# **Contact information**

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

AN11908

AN11908

## 1. Introduction

The NFC Reader Library is a feature complete software support library for NXP's NFC Frontend ICs. It is designed to give developers a faster and simpler way to deliver NFC-enabled products. This multi-layer library, written in C, makes it easy to create NFC based applications. See [1] for more details.

The purpose of the present document is to give instructions on how to install the NFC Reader Library with a NXP Freedom K82F board [2], using MCUXpresso [3], together with NXP's NFC Frontends. In this document PNEV5180B and CLEV6630B demo boards will be used.



AN11908

# 2. FRDM-K82F Development Platform

The NXP Freedom development platform is a set of software and hardware tools for evaluation and development. It is ideal for rapid prototyping of microcontroller-based applications. The Freescale Freedom K82 hardware, FRDM-K82F, is a simple yet sophisticated design featuring a Kinetis K series microcontroller built on the ARM® Cortex®-M4 core which features a floating-point unit (FPU).

FRDM-K82F can be used to evaluate the K80, K81, and K82 Kinetis K series devices. The FRDM-K82F board features the MK82FN256VLL15 MCU, which boasts a maximum operation frequency of 150 MHz, 256 KB of flash, a 256 KB RAM, a full-speed USB controller with available crystal-less operation, and analog and digital peripherals. The FRDM-K82F hardware is form-factor compatible with the ArduinoTM R3 pin layout, providing a broad range of expansion board options. The onboard interface includes a six-axis digital accelerometer and magnetometer, an RGB LED, FlexIO header, and sufficient footprint to add an optional Bluetooth module (for use over UART) and RF module (for use over SPI).

Details how to prepare the board and install proper debug driver are described on FRDM-K82F product page [2].

Note:

NXP Reader Library package for MCUXpresso is preconfigured to be used with SEGGER J-Link debugger tool.

# 3. MCUXpresso IDE

The MCUXpresso IDE enables powerful application development for NXP MCUs based on ARM® Cortex®-M cores, including LPC and Kinetis microcontrollers. The MCUXpresso IDE offers advanced editing, compiling and debugging features with the addition of MCU-specific debugging views, code trace and profiling, multicore debugging, and more. Feature-rich IDE optimized for ease-of-use, based on industry standard Eclipse and GCC providing a powerful application development environment, Supports Freedom, Tower, MCUXpresso and your custom development boards with debug probes from NXP, P&E, and SEGGER. Available in full-featured free (code size unlimited) and affordable professional editions (including MCUXpresso IDE email support and advanced trace features).

Details about the MCUXpresso installation and usage are described on the MCUXpresso product page [3].

#### Importing NFC Reader Library to the MCUXpresso 4.

The NFC Reader library, version for MCUXpresso, is delivered in archive file and contains implementation of the NFC Reader Library, FreeRTOS, SDK 2.0 FRDM-K82F platform and SW examples which demonstrates how to use NFC.

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 ICbased contactless card.

Example projects delivered	with the NFC Reader Library
Example	Description
NfcrdlilbEx1_Basic DiscoveryLoop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. Default configuration parameters are used.
NfcrdlilbEx2_AdvancedDi scoveryLoop	Explains how to poll for different technologies (Tag, P2P, HCE), detect and report them. All configuration parameters are used and explained.
NfcrdlilbEx3_ NFCForum	Explains how to configure the NFC Reader Library for different P2P modes such as Active Mode, Target Mode, Initiator Mode and SNEP Client/Server.
NfcrdlilbEx4_MIFARE Classic	Explains the usage of MIFARE Classic card communication commands.
NfcrdlilbEx5_ ISO15693	Explains the usage of this technology and provides an overview about the most common commands.
NfcrdlilbEx7_ EMVCo_Polling	Explains polling for EMVCo payment cards.
NfcrdlilbEx8_ HCE_T4T	Explains how to emulate a NFC Forum Type 4 Tag supporting read and write operations.
NfcrdlilbEx9_ NTagl2C	Explains NTAG-I2C specific commands.
NfcrdlilbEx10_ MIFAREDESFire	Explains the usage of MIFARE DESFire cards. (This example is delivered with the NFC Reader Library version available via NXP DocStore)
NfcrdlibEx11_ISO10373_ PCD	Example is used to perform ISO 10373-6 PCD compliance validation.
Nfcrdlib_SimplifiedAPI EMVCo	EMVCo loopback application with simplified API, which can be used for EMVCo level 1 digital certification.
Nfcrdlib_SimplifiedAPI EMVCo_Analog	Example is used to perform EMVCo2.6(L1) Analog compliance validation.
Nfcrdlib_SimplifiedAPI ISO	Explains how to use simplified API with different types of cards.

Table 1. **Example projects** 

Following guidelines, describing how to prepare the SW environment to run prepared SW examples on FRDM-K82F board.

### 4.1 Import projects to the MCUXpresso

This chapter explains how to use the MCUXpresso to create workspace and work with delivered projects.

To launch the MCUXpresso for creating, building and debugging projects follow guidelines below:

1. Start MCUXpresso IDE.

The "Eclipse Launcher" dialog box appears and prompts you to select a workspace to use. It is recommended to use an empty folder as a new workspace.

Select a directory as workspace	
MCUXpresso IDE uses the workspace directory to	store its preferences and development artifacts.
Workspace: 510\Documents\MCUXpressoIDE_1	0.0.0_344\workspace_502  Browse
Use this as the default and do not ask again	
Recent Workspaces	
	OK Cancel

The IDE starts and displays the Welcome page.

 Import NFC Reader Library and examples projects. Right click in the "Quickstart Panel" window and select "Import project(s) ..." option.



3. Import wizard appears to select import source. Click "Browse" next to "Archive" and browse dialog appears

Select the examples archive file to import.         Projects are contained within archives (zip) or are unpacked within a directory. Select your project archive or root directory and press <next>. On the next page, select those projects you wish to import, and press <finish>.         Project archives for LPCOpen and 'legacy' examples are provided.         Project archive (zip)         Archive         Browse         Project directory (unpacked)         Root directory         LPCOpen         LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.         MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.         Alternatively, press the button below to Browse the nxp.com website for latest resources.         Browse LPCOpen resources on nxp.com</finish></next>	Import project(s) from	m file system	
Projects are contained within archives (zip) or are unpacked within a directory. Select your project archive or root directory and press <next>. On the next page, select those projects you wish to import, and press <finish>. Project archives for LPCOpen and 'legacy' examples are provided. Project archive (zip) Archive Browse Project directory (unpacked) Root directory (unpacked) Root directory Browse LPCOpen LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers. MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com</finish></next>	Select the examples arch	ive file to import.	
Project archives for LPCOpen and 'legacy' examples are provided.         Project archive (zip)         Archive       Browse         Project directory (unpacked)         Root directory       Browse         LPCOpen         LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.         MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.         Alternatively, press the button below to Browse the nxp.com website for latest resources.         Browse LPCOpen resources on nxp.com	Projects are contained wit project archive or root dir wish to import, and press	thin archives (.zip) or are unpacked within a director rectory and press <next>. On the next page, select t <finish>.</finish></next>	y. Select your hose projects you
Project archive (zip) Archive Browse Project directory (unpacked) Root directory Browse LPCOpen LPCOpen LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers. MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com	Project archives for LPCOp	pen and 'legacy' examples are provided.	
Archive       Browse         Project directory (unpacked)       Browse         Root directory       Browse         LPCOpen       Event         LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.       MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.         Alternatively, press the button below to Browse the nxp.com website for latest resources.       Browse LPCOpen resources on nxp.com	Project archive (zip)		
Project directory (unpacked) Root directory Browse LPCOpen LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers. MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com	Archive		Brows
Root directory         Browse           LPCOpen         LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.           MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory.           Alternatively, press the button below to Browse the nxp.com website for latest resources.           Browse LPCOpen resources on nxp.com	Project directory (unpack	ked)	
LPCOpen LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers. MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com	Root directory		Brows
LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers. MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com	LPCOpen		
MCUXpresso IDE includes the LPCOpen packages which can be imported directly by pressing the Browse button in the Project archive (zip) section, above, and navigating to the Examples/LPCOpen directory. Alternatively, press the button below to Browse the nxp.com website for latest resources. Browse LPCOpen resources on nxp.com	LPCOpen is the recomme	ended code base for Cortex-M based NXP LPC Micr	ocontrollers.
	MCUXpresso IDE include button in the Project arch Alternatively, press the bu	is the LPCOpen packages which can be imported di hive (zip) section, above, and navigating to the Exar witton below to Browse the nxp.com website for late rces on nxp.com	rectly by pressing the Brows nples/LPCOpen directory. est resources.

- In this step archive file containing the NFC Reader Library and examples projects needs to be specified. Search for a file name NxpNfcRdLib\_KinetisK82\_v05.02.00.zip in NxpNfcRdLib installation folder.
- 5. Finally click on "Finish" button to complete the import wizard.

AN11908

Import project(s) from file system	3
Import project(s) from file system	7.
Select a directory to search for existing Eclipse projects.	
Projects	
	_
FreeRTOS (RTOS/FreeRTOS/)	
✓ Ipc_board_nxp_lpcxpresso_1769 (Platform/LPCOpen_1769/lpc_board_nxp_lpcx	ך
V Ipc_cnip_1/sx_bx (Platform/LPCOpen_1/b9/Ipc_cnip_1/sx_bx/)	
Nicralib_simplifiedAPI_EMVCo_mcuv (ComplianceApp/Nicralib_simplifiedAPI_EMVCo_mcuv)	
Nfcrdlib SimplifiedAPI ISO mcux (Examples/Nfcrdlib SimplifiedAPI ISO/mcux	
V NfcrdlibEx1_BasicDiscoveryLoop_mcux (Examples/NfcrdlibEx1_BasicDiscovery	
VfcrdlibEx10_MIFAREDESFire_mcux (Examples/NfcrdlibEx10_MIFAREDESFire/r	
VfcrdlibEx11_ISO10373_PCD_mcux (ComplianceApp/NfcrdlibEx11_ISO10373_F	
Options	
Copy projects into workspace	
Working sets	
Add project to working sets	
Working sets: Select	
	4
	_
? < Back Next > Finish Cancel	
art Draigata Stan 2	
ort Projects – Step 3	

6. All projects are listed in the "Project Explorer" window.

	ြာ Project Explorer 🛛 🕞 🧏 🐌 🤝 🗖 🗖
	ApiDocumentation
	FreeRTOS_Library
	» Stordlib_SimplifiedAPI_EMVCo
	» Stordlib_SimplifiedAPI_ISO
	» StordlibEx1_BasicDiscoveryLoop
	» MfcrdlibEx11_ISO10373_PCD
	» MfcrdlibEx2_AdvancedDiscoveryLoop
	» StordlibEx3_NFCForum
	» StordlibEx4_MIFAREClassic
	» StordlibEx5_ISO15693
	» StordlibEx7_EMVCo_Polling
	» StordlibEx8_HCE_T4T
	» StordlibEx9_NTagI2C
	» AxpNfcRdLib
	▷  SDK_2.0_FRDM-K82F
Fig 6. Import Projects -	Step 4

#### 4.2 Install SDK

To build, run and debug projects it is mandatory to install board SDK. The FRDM-K82F SDK is shipped with the NFC Reader Library and examples package.

1. Left click at Installed SDKs view. Pop-up menu appears.

- To install on SDK simp	w drag and drap an SDK	(rin file (folder) into the 'Installed SDKs' u	iouu
Name	Version	Location	iew.
	<ul> <li>Import archive</li> <li>Import folder</li> <li>Unzip archive</li> </ul>	re r	
	<ul><li>Open Location</li><li>Open Default</li></ul>	n t Location	

2. Right click "Import archive" and browse to the NXP NFC Reader Library and examples projects archive file. Import process starts automatically.

🕅 Installed SDKs 🛛 🔲 Properties	s 📮 Console 💈	Problems	Memory	Notruction Trace
🕅 Installed SDKs				
To install an SDK, simply drag and	drop an SDK (z	ip file/folde	r) into the 'In	stalled SDKs' view.
Name	Version	Location		
V # SDK_2.x_FRDM-K82F	2.2.0	👰 <defa< td=""><td>ult Location:</td><td>&gt;/SDK_2.x_FRDM-K8</td></defa<>	ult Location:	>/SDK_2.x_FRDM-K8
Fig 8. Adding SDK – Step 2				

AN11908

## 4.3 Build, Debug and Run Projects

The provided NFC Reader Library project and example projects are pre-configured to be used with FRDM-K82F target board.

#### 4.3.1 Building Projects

To build only selected projects, and all dependency projects, right-click on the project and select "Build Project" option. Default build configuration is debug (DebugFRDMK82F).

File Edit Naviga	te S	earch Project Run	FreeRTOS	Window	Help
i 📑 🗕 🖬 👘   🛞 י	- %	▼ 🗟 🌴 🌴 🔹 🚺	<b>- </b>	) ≈	
陷 Project Ex 🛛	묾 Pe	eripheral 👭 Registe	ers 🐁 Symb	ool V 🗖	
<u></u>				E 🔄	▽
NfcrdlibEx1_	Racic	New		,	1
NicrdibEx10	-	Go Into			
» P NfcrdlibEx2_		Open in New Windo	w		=
» Store NfcrdlibEx3_		Сору		Ctrl+C	
NfcrdlibEx4_	ß	Paste		Ctrl+V	
NicrdlibEx3_	×	Delete		Delete	
NfcrdlibEx8_	ł	Source		,	
» StordlibEx9_	1	Rename		F2	
Ntcrdlib1st1.	2 2	Import			Ł
() Quicks 🛛 🕬=	2	Export			
		Build Project			
		Clean Project			
		Refresh		F5	
Start here		Close Unrelated Proj	jects		
New project		Build Configurations	;	,	
Import SDK e		Build Targets		,	•
Import project		Index		,	•

Alternatively, select Project > Build All to build all projects at once.

#### 4.3.2 Debugging Projects

To debug a project, perform following steps.

1. Launch the IDE and select a project in 'Project explorer".

2. Click Debug menu item in Quickstart panel.



3. Probe discovered dialog appears.

**Application note** 

COMPANY PUBLIC

AN11908

1	probe found. Select the pr	robe to use:			
Av	ailable attached p	robes			
	Name	Serial number/ID	Туре	Manufa	IDE Debug Mode
JLink	J-Link OpenSDA	621000000	USB	SEGGER	All-Stop
Su	pported Probes (tick/untic	k to enable/disable)			
1	MCUXpresso IDE LinkSer	ver (inc. CMSIS-DAP) probe	S		
V	P&E Micro probes				
•	be seen and antions				
Se	earch again				
7 R	Remember my selection (f	or this Launch configuratior	)		
?				OK	Cancel
C				UK	Cancer

4. Debug output in console view appears.

🕅 Installed SDKs 🔲 Pro	operties 📮 Console 🛚 🧏 Problems 🔋 Memory 🚸 Instruction Trace 🎇 SWO Trace Config 📼 Power Measurement Tool
[MCUXpresso Semihostin	ng Telnet console for NfcrdlibEx1_BasicDiscoveryLoop_mcux started on port 2333]
[MCUXpresso Semiho	osting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop_mcux started on port 2333]
SEGGER J-Link GDB	Server V6.14b - Terminal output channel
BasicDiscoveryLoop	b Example:
Fig 12. Console	e output

AN11908

# 5. Connecting FRDM-K82F Board to NFC Frontend board

This chapter provides instructions how to connect FRDM-K82F development board to the host PC and to the NFC Frontend board: PNEV5180B, CLEV6630B to run NFC Reader Library.

### 5.1 Connecting the FRDM-K82F to the PC Host

To flash binaries and debug project it is required to connect host PC and FRDM-K82F board via debug interface.

The debug interface on the MK82FN512VDC12 MCU is a serial wire debug (SWD) port with trace output capability. There are two debug interfaces on the FRDM-K82F: an onboard OpenSDAv2 circuit (J5) and a K82F direct SWD connection via a 10-pin header (J19). To use an external debugger, such as J-Link on J19, you may need to disconnect the OpenSDAv2 SWD circuit from the K82F by removing jumpers J6 and J7.



Connect FRDM-K82F board and host PC via micro USB cable as shown below.

#### Note:

NXP Reader Library package for KDS supports SEGGER J-Link debugger tool, therefore it is required to flash on appropriate debugger binary. Detailed guidelines how to prepare the board with the right debugger are available on FRDM-K82F product web page.

Application note

**COMPANY PUBLIC** 

## 5.2 Connecting the FRDM-K82F to the PNEV5180B

To assure direct access from the FRDM-K82F development board to the PN5180 frontend IC on the PNEV5180B board, the following change has to be performed:

In total six resistors in two groups need to be removed to obtain proper decupling of the LPC1769 MCU from the PNEV5180 board (Fig 14).



To check for the correct PNEV5180B board configuration please refer to the UM10954 - PN5180 SW Quick start guide.

To configure PN5180 NFC Frontend in the NFC Reader Library it is mandatory to define related preprocessor directive (project properties->settings->preprocessor) to PHDRIVER\_FRDM\_K82FPN5180\_BOARD.

Connect both boards as described below.

Table 2.         Pin Connection Table for PNEV5180B		
PIN Function	FRDM-K82F	PNEV5180B
MOSI	J1-13 (PORTA-16)	MOSI
MISO	J1-7 (PORTA-17)	MISO
SCK	J1-15 (PORTA-15)	SCK
SSEL	J1-9 (PORTA-14)	SSEL
BUSY	J1-1 (PORTA-5)	BUSY
RESET	J1-3 (PORTA-13)	PN_RESET
IRQ	J1-11 (PORTC-7)	IRQ
IFSELO	J1-2 (PORTB-16)	-
IFSEL1	J1-4 (PORTB-17)	-
SDA	J4-4 (PORTC-10)	-
SCL	J4-2 (PORTC-11)	-
DWL	J1-5 (PORTA-12)	DWL
AD0	J2-1 (PORTB-20)	-
AD1	J2-3 (PORTB-21)	-
AD2	J2-5 (PORTB-22)	-
GND	J3-12; J3-14; J2-14	GND
+3.3V	J3-4; J3-8	-
+5V	J3-16	-

AN11908



## 5.3 Connecting the FRDM-K82F to the CLEV6630B

To assure direct access from the FRDM-K82F development board to the CL66303 frontend IC on the CLEV6630B board, the following changes are needed:

In total six resistors marked by red squares need to be removed to obtain proper decupling of the LPC1769 MCU from the CLEV6630B board (Fig 16).



To enable RC663 NFC Frontend in the NFC Reader Library it is mandatory to define related preprocessor directive (project properties > settings > preprocessor) to PHDRIVER\_FRDM\_K82FRC663\_BOARD.

Connect both boards as described below.

PIN Function	FRDM-K82F	CLEV6630B
MOSI	J1-13 (PORTA-16)	MOSI
MISO	J1-7 (PORTA-17)	MISO
SCK	J1-15 (PORTA-15)	SCK
SSEL	J1-9 (PORTA-14)	SSEL
BUSY	J1-1 (PORTA-5)	-

#### Table 3. Pin Connection Table for CLEV6630B

PIN Function	FRDM-K82F	CLEV6630B
RESET	J1-3 (PORTA-13)	CLRC_NRST
IRQ	J1-11 (PORTC-7)	IRQ
IFSELO	J1-2 (PORTB-16)	IF0
IFSEL1	J1-4 (PORTB-17)	IF1
GND	J3-12; J3-14; J2-14	GND
+3.3V	J3-4; J3-8	-
+5v	J3-16	-



## 6. References

[1] The NFC Reader Library gives Software support for NFC Frontend solutions.

For more information about it please visit <u>http://www.nxp.com/products/:NFC-READER-LIBRARY.</u>

[2] FRDM-K82F: NXP Freedom Development Platform for Kinetis K82, K81, and K80 MCUs.

For more information about it please visit http://www.nxp.com/products/software-and-tools/run-time-software/kinetis-softwareand-tools/ides-for-kinetis-mcus/freescale-freedom-development-platform-for-kinetisk82-k81-and-k80-mcus:FRDM-K82F

[3] MCUXpresso Integrated Development Environment (IDE).

For more information about it please visit <u>http://www.nxp.com/products/software-and-tools/run-time-software/mcuxpresso-</u> <u>software-and-tools/mcuxpresso-integrated-development-environment-</u> <u>ide:MCUXpresso-IDE</u>

**Application note** 

**COMPANY PUBLIC** 

# 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. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

# 7.4 Trademarks

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

MIFARE— is a trademark of NXP B.V.

MIFARE Classic - is a trademark of NXP B.V.

DESFire — is a trademark of NXP B.V

Kinetis — is a trademark of NXP B.V.

Application note

**COMPANY PUBLIC** 

# 8. List of figures

Fig 1.	FRDM-K82F: NXP Freedom Development
	Platform for Kinetis K82, K81, and K80 MCUs3
Fig 2.	Eclipse Launcher Dialog Box6
Fig 3.	Import Projects - Step 16
Fig 4.	Import Projects – Step 27
Fig 5.	Import Projects - Step 38
Fig 6.	Import Projects - Step 48
Fig 7.	Adding SDK - Step 19
Fig 8.	Adding SDK – Step 29
Fig 9.	Build Project10
Fig 10.	Start debug session11
Fig 11.	Select Launch Configuration12
Fig 12.	Console output12
Fig 13.	Connecting the FRDM-K82F Board via Micro USB 13
Fig 14.	Resistors to be removed marked red14
Fig 15.	Connections between FRDM K82F and
-	PNEV5180B16
Fig 16.	Resistors to be removed marked red17
Fig 17.	Connections between FRDM K82F and
	CLEV6630B18

# 9. List of tables

Table 1.	Example projects5
Table 2.	Pin Connection Table for PNEV5180B15
Table 3.	Pin Connection Table for CLEV6630B17

# 10. Contents

1.	Introduction	.3
2.	FRDM-K82F Development Platform	.4
3.	MCUXpresso IDE	.4
4.	Importing NFC Reader Library to the	
	MCUXpresso	.5
4.1	Import projects to the MCUXpresso	.6
4.2	Install SDK	.9
4.3	Build, Debug and Run Projects	0
4.3.1	Building Projects	0
4.3.2	Debugging Projects	0
5.	Connecting FRDM-K82F Board to NFC	
	Frontend board1	3
5.1	Connecting the FRDM-K82F to the PC Host1	13
5.2	Connecting the FRDM-K82F to the	
	PNEV5180B1	4
5.3	Connecting the FRDM-K82F to the	
	CLEV6630B1	17
6.	References1	9
7.	Legal information	20
7.1	Definitions2	20
7.2	Disclaimers	20
7.3	Licenses	20
7.4	Trademarks2	20
8.	List of figures	21
9.	List of tables	22
10.	Contents	23

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

© NXP B.V. 2018.

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

Date of release: 18 April 2018

All rights reserved.

407213 Document identifier: AN11908