AN11022

CLRC663 Evaluation board quick start guide

Rev. 1.4 — 15 May 2017
205914

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	CLRC663, CLRC663 plus, CLEV6630B, CLRC663 evaluation board, CLRC663 customer board, CLRC663 GUI, GUI, CLRC663 Support Tool, NFC Cockpit
Abstract	This document describes the CLEV6630B (CLRC663 evaluation board), and how to use it. It describes the NFC Cockpit (Version 3.6), which allows an easy basic access to the CLRC663 registers and EEPROM in combination with basic reader functionality.

Revision history

Rev	Date	Description
1.4	20170515	MCUXpresso IDE installation and usage chapter added Software example descriptions added
1.3	20170503	Update with new CLEV6630B V2.0 and NFC Cockpit
1.2	20150114	RC663 Schematic updated
1.1	20120712	Some Figures updated because of quality reasons, Section Licenses updated
1.0	20120216	Initial release

Contact information

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

1. Introduction

This document describes the CLEV6630B 2.0 (CLRC663 evaluation board), which provides an easy evaluation of the features and functions of the CLRC663.

It provides the first steps to operate the board, using the NFC Cockpit (Version 3.6 or higher).

The default antenna is a 65mm x 65mm antenna with some metal layer inside the antenna area. This antenna is not an optimum antenna as such, but intends to demonstrate the performance and register settings of the CLRC663 under typical design constraints like LCD or some metal (e.g. PCB) inside the antenna area.

1.1 CLRC663 registers & EEPROM concept

The CLRC663 uses internal registers to adapt and optimize the functionality and performance for each of the supported protocols and data rates dependent on the connected antenna, matching network and receiver path. It offers an EEPROM, which contains the default settings for all the supported protocols (locked). These settings are loaded into the registers with the LoadProtocol command for each supported protocol and data rate.

The default EEPROM configuration settings are optimized for the generic use, based on the 65mmx65mm antenna of the board CLEV6630B, and cannot be updated by the user as such. Individual settings must be overwritten by the host µC after the LoadProtocol.

Alternatively, customized settings can be used for the major relevant registers in an extra EEPROM area. Then the command LoadReg must be used to copy the customized EEPROM content into the registers.

Some of these settings can or even **must** be adapted towards a new antenna design (e.g. the Rx settings).

Some EEPROM configuration data is independent from the used protocols and defines e.g. the startup behavior of the CLRC663 or the functionality of LowPower Card detection and requires attention as well for optimum performance of the chip.

1.2 CLEV6630B concept

The basic **concept of the CLEV6630B** is to enable the user to perform a quick evaluation of the CLRC663, and also connect his own antenna to the CLRC663 board. In addition, dedicated boards which allow to solder custom matching components are available. The NFC Cockpit can be used to optimize the CLRC663 antenna tuning, to perform the related TX and Rx optimization without touching any source code.

All the relevant PN5180 registers can be modified and fine-tuned using the NFC Cockpit. For the most relevant registers the customized settings can typically be stored in the CLRC663 EEPROM.

The NFC Cockpit also allows a dump of the complete user EEPROM content into an XML file. This file then can be loaded again into the EEPROM. That allows to manage and exchange different user or antenna configurations. In addition, the register settings found to work well using the NFC Cockpit, can be used during user code development as well.

As soon as the register settings for the targeted protocols and data rates are defined, the NFC Reader Library including the HAL can be used to start the development of the user application. Examples illustrate the usage of the library for typical use cases.

The source code examples of the NFC Reader Library can be used to develop an own application directly on the LPC1769 (see Fig 3), or can serve as a starting point for porting the NFC Library to any other microcontroller platform.

2. Hardware

The CLEV6630 V2.0, as shown in Fig 1 and Fig 2, provides a lot of test functions which might not be used for the typical hardware and software evaluation. It can be used as a simple standard reader without modification, it can be used to define and optimize the analog settings for any connected antenna or it can be used to develop and modify any RFID and NFC application based on the NFC Reader Library.

2.1 Hardware introduction

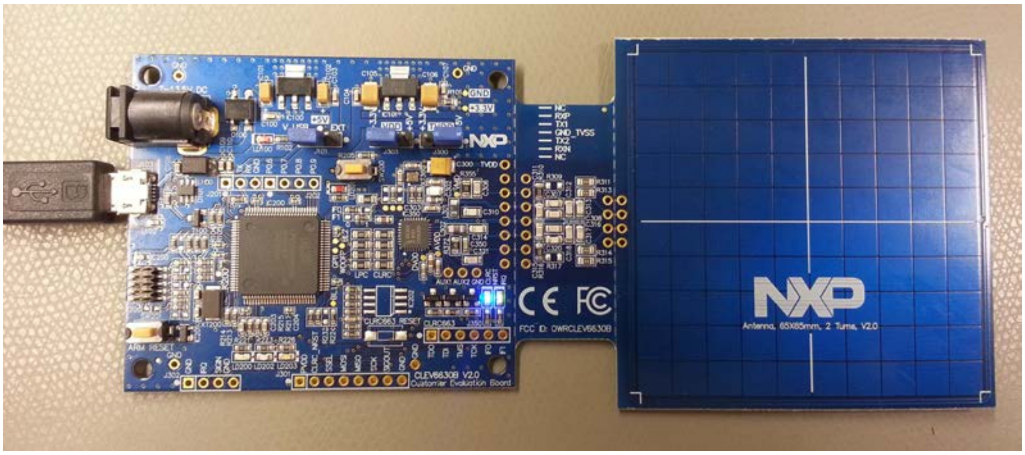
The CLRC663 is supplied with a supply voltage, which can be chosen between: internal and external supply. For the internal supply either 5V or 3.3V can be used. The external power supply can be an AC or DC supply (polarity does not matter) with at least 7.5V, since the board provide a rectifier and LDO to supply the circuit with 5V and 3.3V.

The CLRC663 is connected to an NXP LPC1769 µC via SPI. A specific firmware on the LPC1769 allows to use the CLEV6630B together with the NFC Cockpit.

The connection to the PC is done via USB: USB Micro connectors are supported. The use of the shielded USB cable is required to meet the FCC/CE specifications.

Another connection option allows to connect a LPC-LINK2 board the CLEV6630B by means of a debug cable. This allows the development of custom software or the execution of the NFC Reader Library code including samples.

In case a different host microcontroller shall be used, the SPI interface is available for connection to an external host (the on board LPC1769 is not used in this case).



(1) Version 2.0

Fig 1. CLEV6630B Customer Evaluation Board

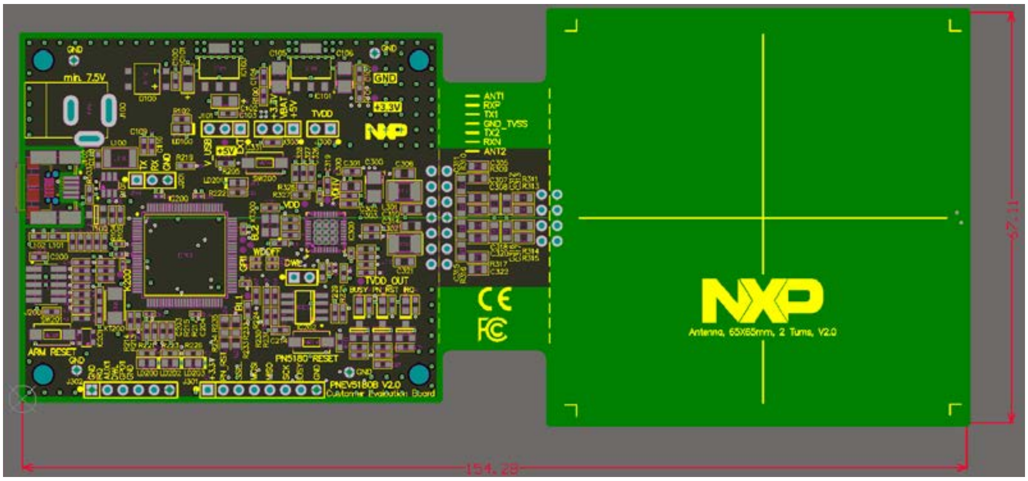


Fig 2. PNEV5180B 2.0 top view

2.2 Schematics

The complete schematics of the CLRC663 evaluation board are shown in the Fig 3, Fig 4, Fig 5, Fig 7, and Fig 8.

2.2.1 LPC1769

The CLEV6630B contains an NXP LPC1769 (see Fig 3).

An LPC Linker can be connected to the LPC1769 via the JTAG interface (see Fig 4).

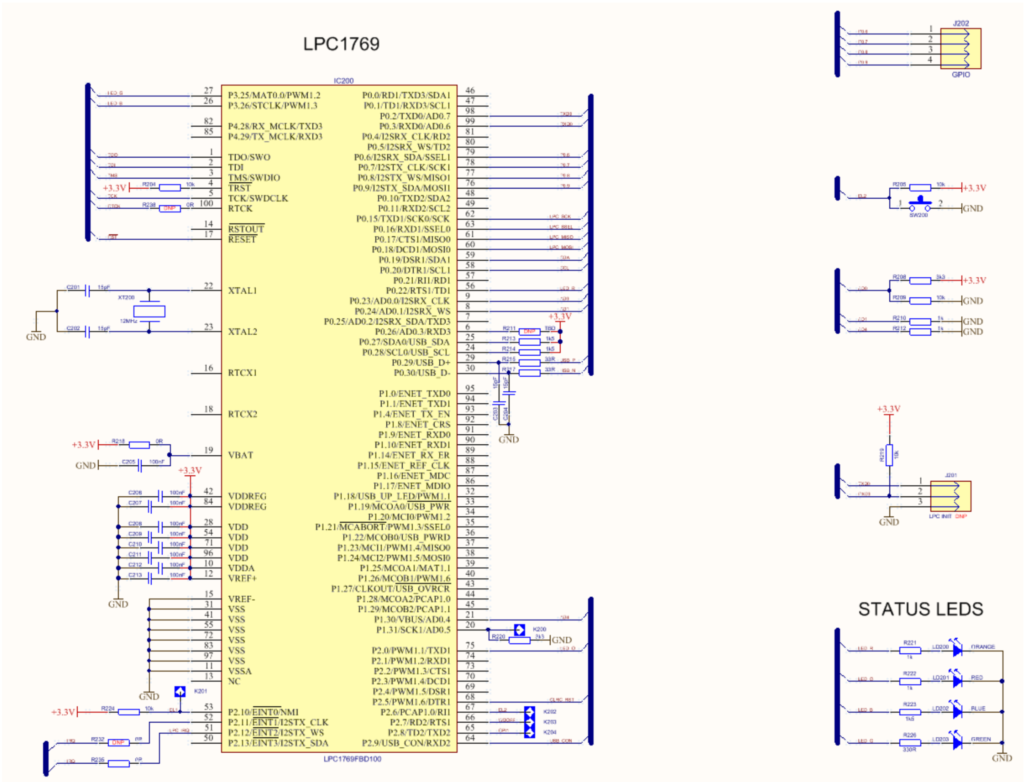
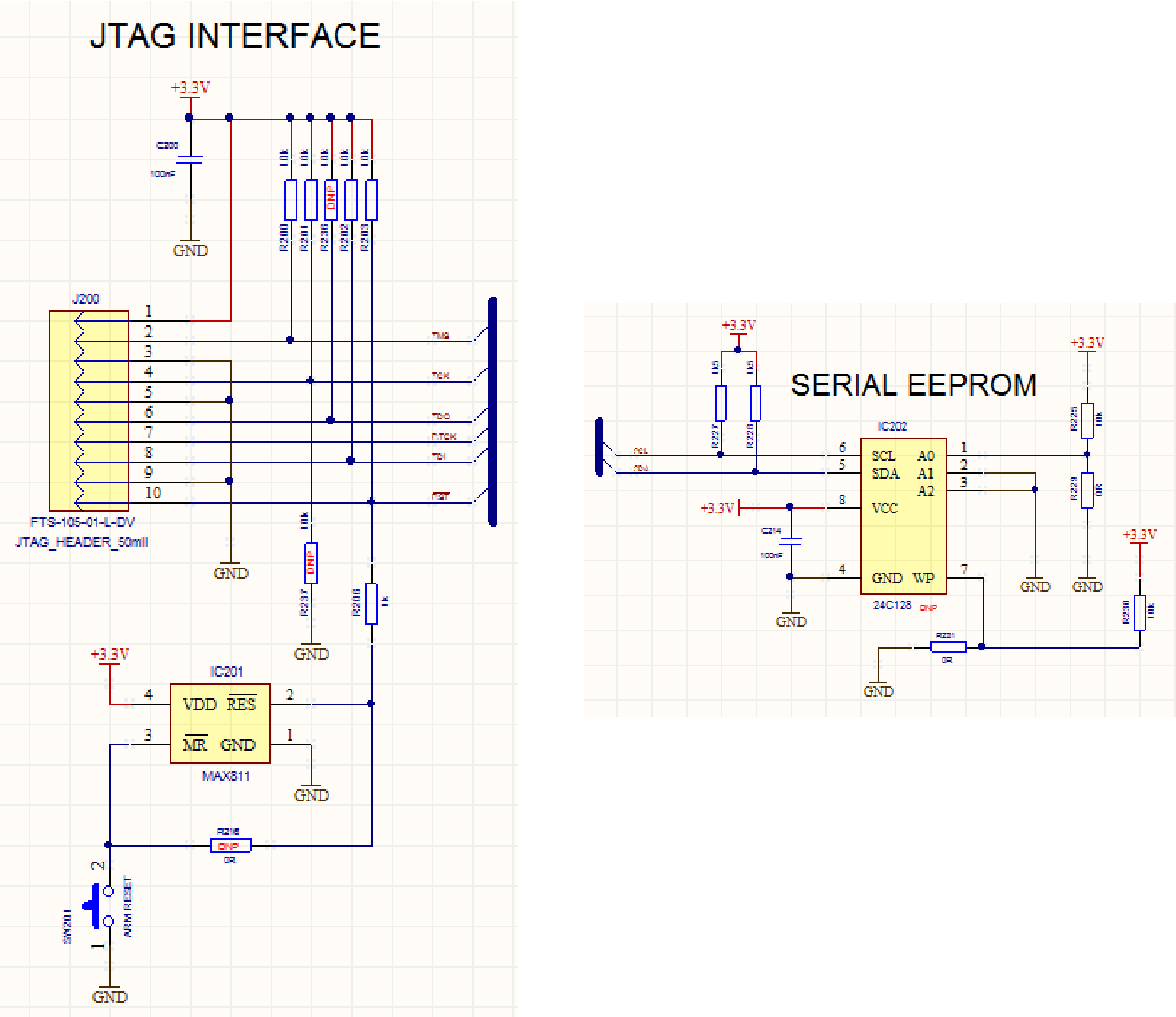


Fig 3. LPC1769



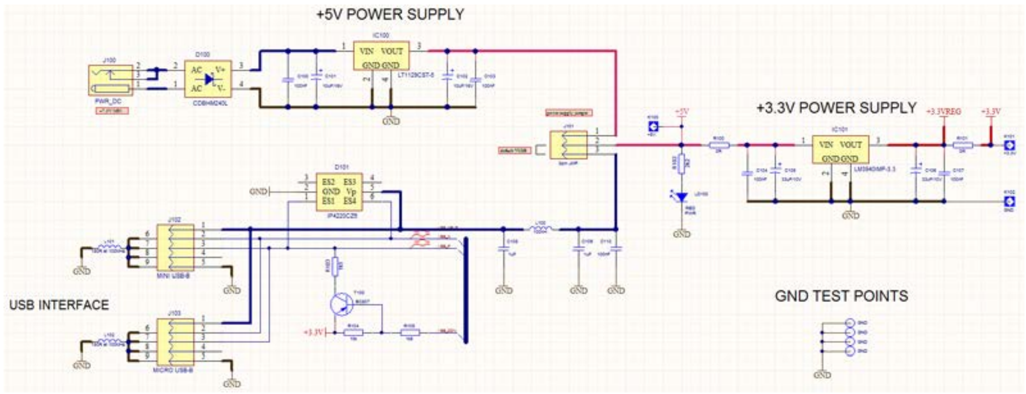
a. JTAG Interface

b. Serial EEPROM not assembled

Fig 4. JTAG interface and Serial EEPROM

2.2.2 Power supply

The default settings use the power supply from the USB connector. For the maximum performance and a better test capability the external power supply should be connected. The AC or DC power input can cover any power supply providing an AC or DC voltage between 7.5 and 12V.

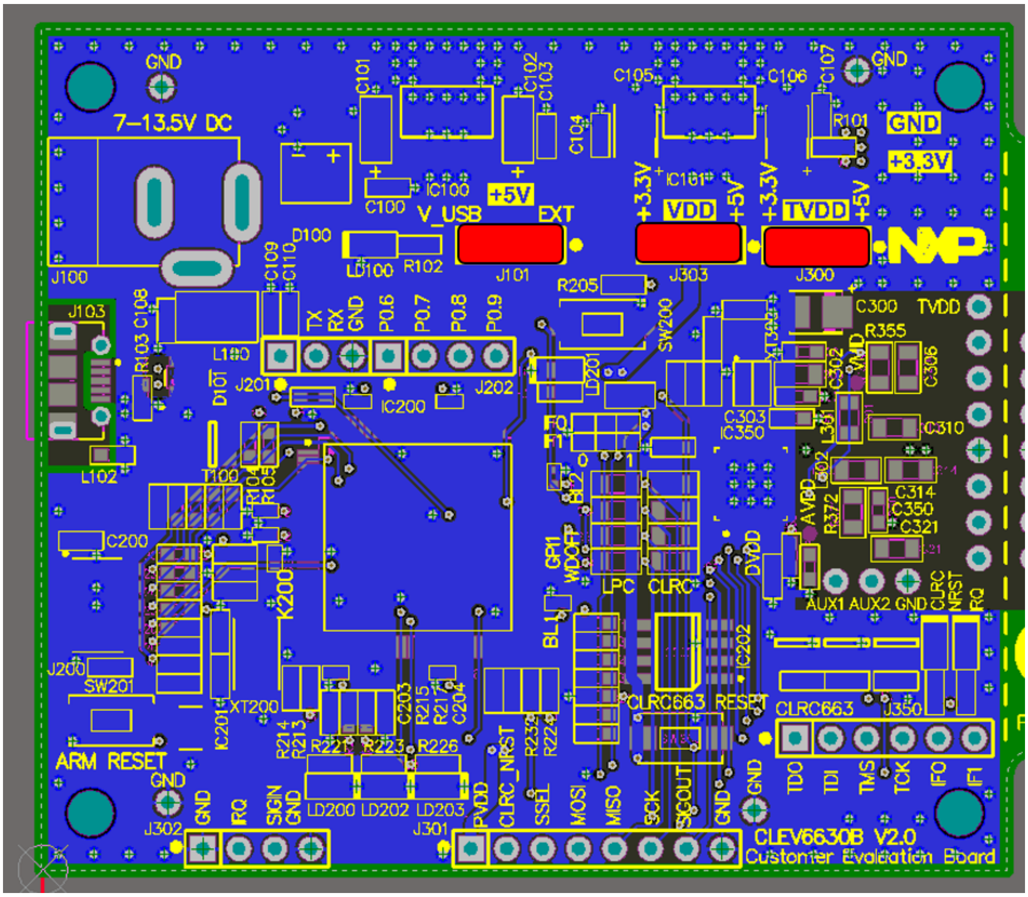


(1) USB supply, external supply & and LPC1769 supply

Fig 5. Power supply

As soon as the board is supplied with power, the red LED LD100 must be on.

The CLRC663 evaluation board provides two LDOs, one for 5V and one for 3.3V. 5V LDO is only be used, if the external power supply is connected and used (J101 default). Using USB power might not give the best RF performance, since the USB voltage level might not be stable 5V.



(1) J101: power selection; J303: VDD selection; J300 TVDD selection

Fig 6. CLEV663B jumpers

Three jumpers can be used to evaluate the different power supply options:

J101: either external or USB power supply (default)

J303: either VBAT = 5V or 3.3V (default)

J300: closed (default) or to measure the ITVDD (bridge with an ampere meter) or to supply the CLRC663 (center pin of J300) with external TVDD from external DC power supply

Note: The best RF performance can be achieved with external power supply.

2.2.3 CLRC663

The CLRC663 is shown in Fig 7.

The clock is based on a 27.12 MHz crystal.

During the antenna tuning and overall hardware design typically the ITVDD must be checked. This can be done with the JP300 (“TVDD”), either using an external power supply or just using an ampere meter instead of the jumper.

The relevant test signals can be derived from the digital test pins at the bottom of the board and the two analog test pins AUX1 and AUX2.

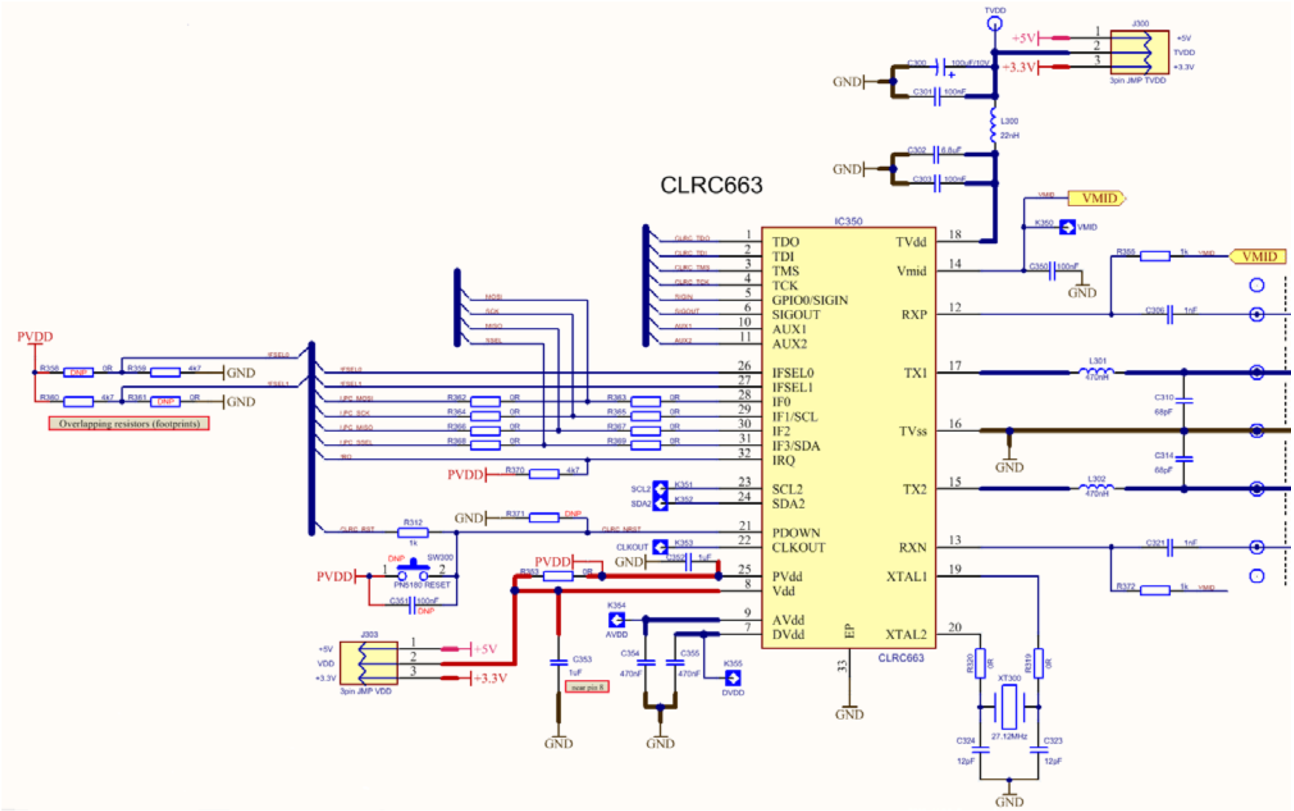
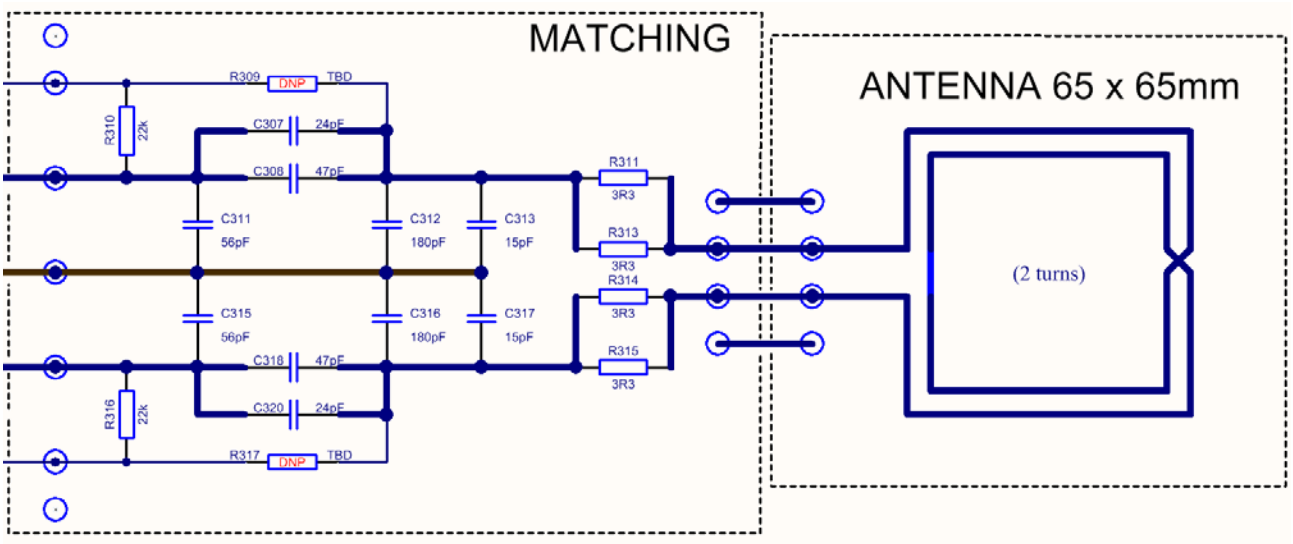


Fig 7. CLEV663B CLRC663

The antenna connection uses the standard tuning circuit. The EMC filter is designed with a cut off frequency of $f_{EMC} \approx 21$ MHz, and the antenna impedance is tuned to $Z \approx 20\Omega$.

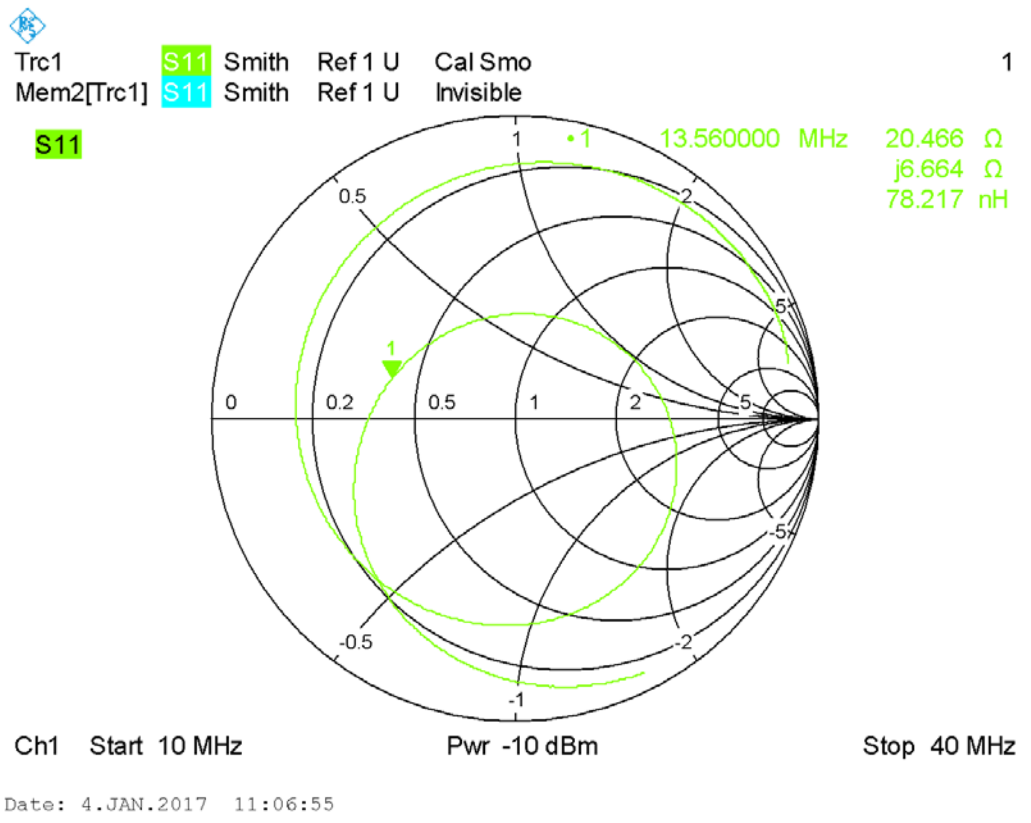


(1) Default tuning

Fig 8. CLEV663B V2.0 antenna circuit

The “asymmetrical” tuning (see Fig 9) is a compromise to provide optimum power transfer and good wave shapes in combination with good loading effects, which automatically reduce the field strength under strong loading conditions.

Note: The CLRC663 *plus* can drive more output power than the CLRC663, so the antenna for the CLRC663 *plus* could be tuned with a lower impedance to increase the field strength. However, the maximum allowed field strength must be taken into account, too.



```
Trc1        S11  Smith   Ref 1 U   Cal Smo                              1
Mem2[Trc1]  S11  Smith   Ref 1 U   Invisible

S11
                         • 1   13.560000  MHz    20.466  Ω
                                                  j6.664  Ω
                                                  78.217  nH

Ch1   Start  10 MHz           Pwr  -10 dBm                 Stop  40 MHz

Date: 4.JAN.2017  11:06:55
```

(1) Typical board from production

Fig 9. CLEV663B V2.0 antenna tuning

2.3 Jumper settings

Three jumpers can be used to evaluate the different power supply options:

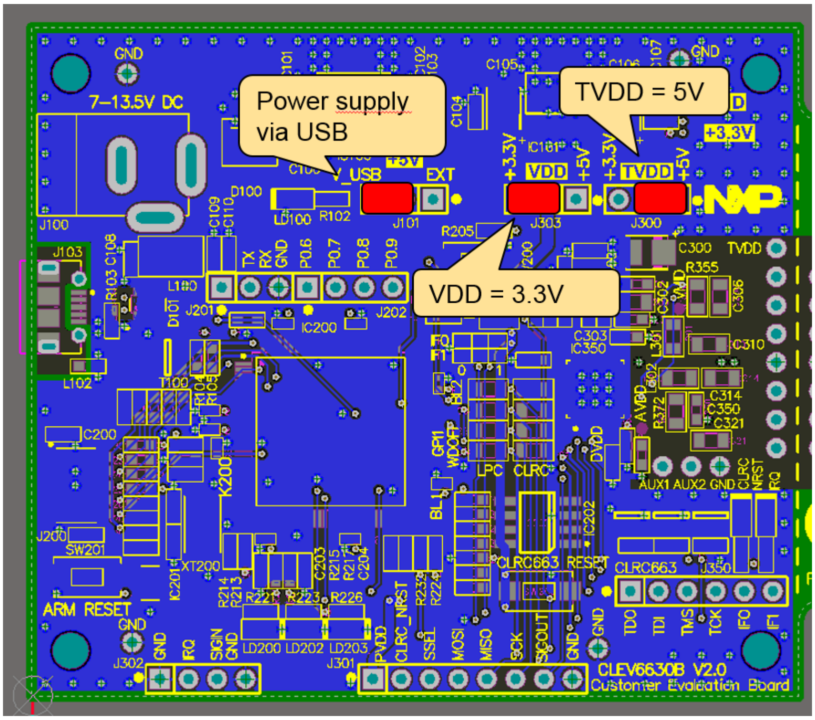
J101: either external or USB power supply (default)

J303: either VBAT = 5V or 3.3V (default)

J300: closed (default) or to measure the ITVDD

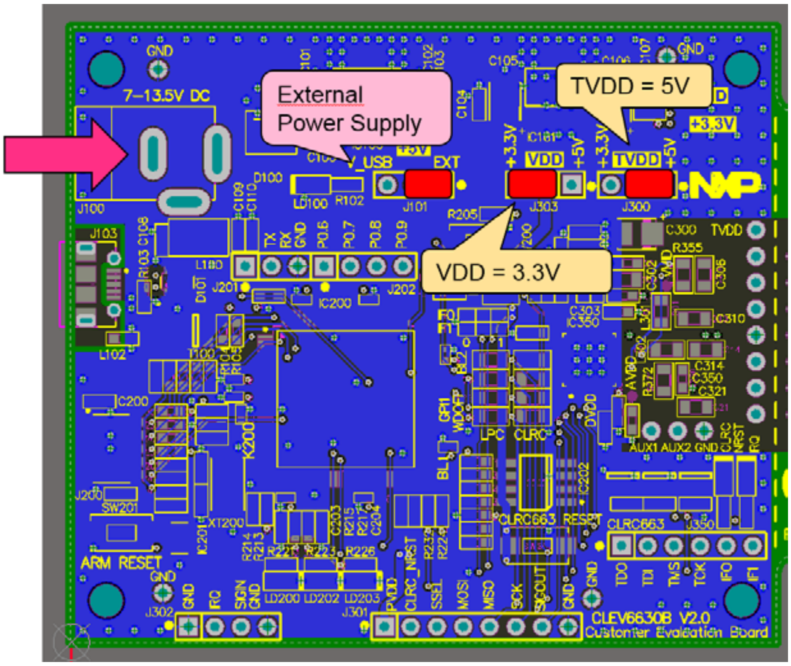
Fig 10 shows the default jumper settings for operation powered via USB.

Fig 11 shows the jumper setting for the operation externally powered.



(1) Default settings, power supply via USB

Fig 10. CLEV663B V2.0 Default jumper settings



(1) External power supply: V ≥ 7V

Fig 11. PNEV5180B 2.0 jumper settings for external power supply

3. Software

The CLEV663B 2.0 evaluation board is delivered with a graphical user interface application (GUI), the NFC Cockpit. The NFC Cockpit can be used to explore the functionality of the CLRC663 and perform RF and antenna design related tests. It allows a direct register access as well as EEPROM read and write access. The NFC Cockpit therefore can be used to configure & test the CLRC663.

3.1 LPC Firmware and Driver

The LPC firmware is installed by default on the CLRC663B and is ready to use. So no LPC firmware installation is required, if the board is only used with the NFC Cockpit.

However, the LPC1769 might be used for software development together with one of the NXP software examples (including the NFC Reader Library). In such case the LPC FW must be re-installed afterwards, if the CLEV663B is supposed to be used together with the NFC Cockpit again. Reason for this is that any software development using the LPCXpresso will erase the default firmware. The use and re-installation of the LPC firmware using the LPCXpresso is described in [5].

In any case the correct PC VCOM driver must be installed, before the NFC Cockpit can be used with the CLEV663B evaluation board. This driver needs to be manually installed, using the “install_vcom.bat” in the subdirectory NFC Cockpit _v xyz \VCOM.

For the first start with the CLEV663B refer to section 4.

3.1.1 LPC Firmware installation

For installation of LPC firmware the LPC link and a LPCXpresso tool is required. For details refer to [5].

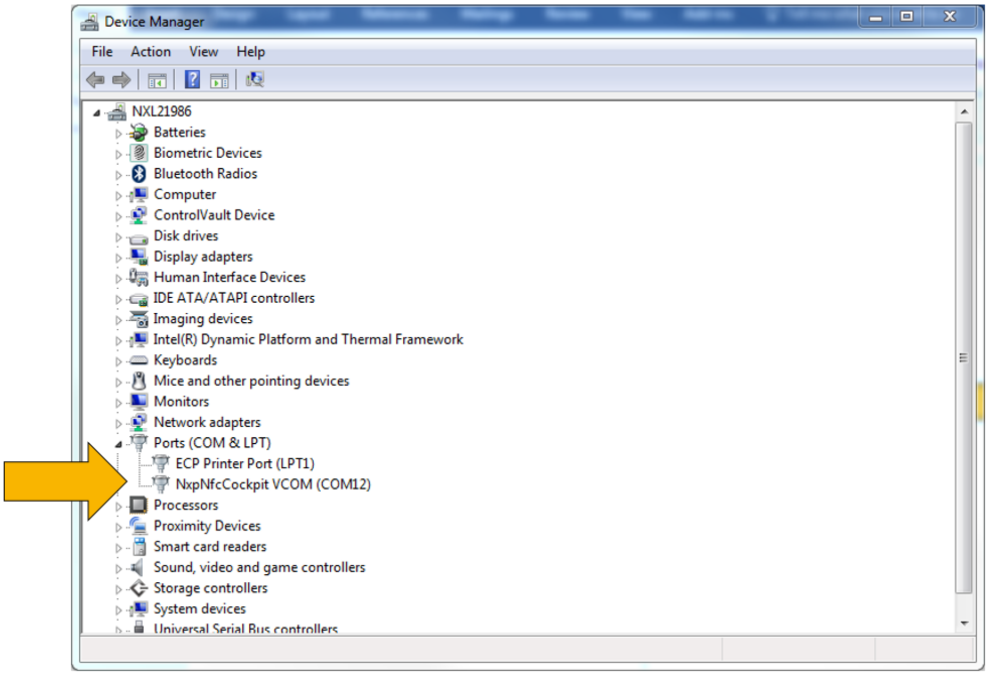
3.1.2 LPC Driver installation

Before the first connection of the CLEV663B (with LPC firmware) to the PC, the driver must be installed with

\Name of the GUI package\VCOM\install_vcom.bat

After successful installation of the driver, the CLEV663B can be connected to the PC and will show up as VCOM device on a COM port, as shown in Fig 12.

<u>Note for possible future NFC Cockpit updates:</u> Please make sure to use latest driver version, otherwise the application might not work correctly. In case of doubt re-install the driver of the corresponding NFC Cockpit package.



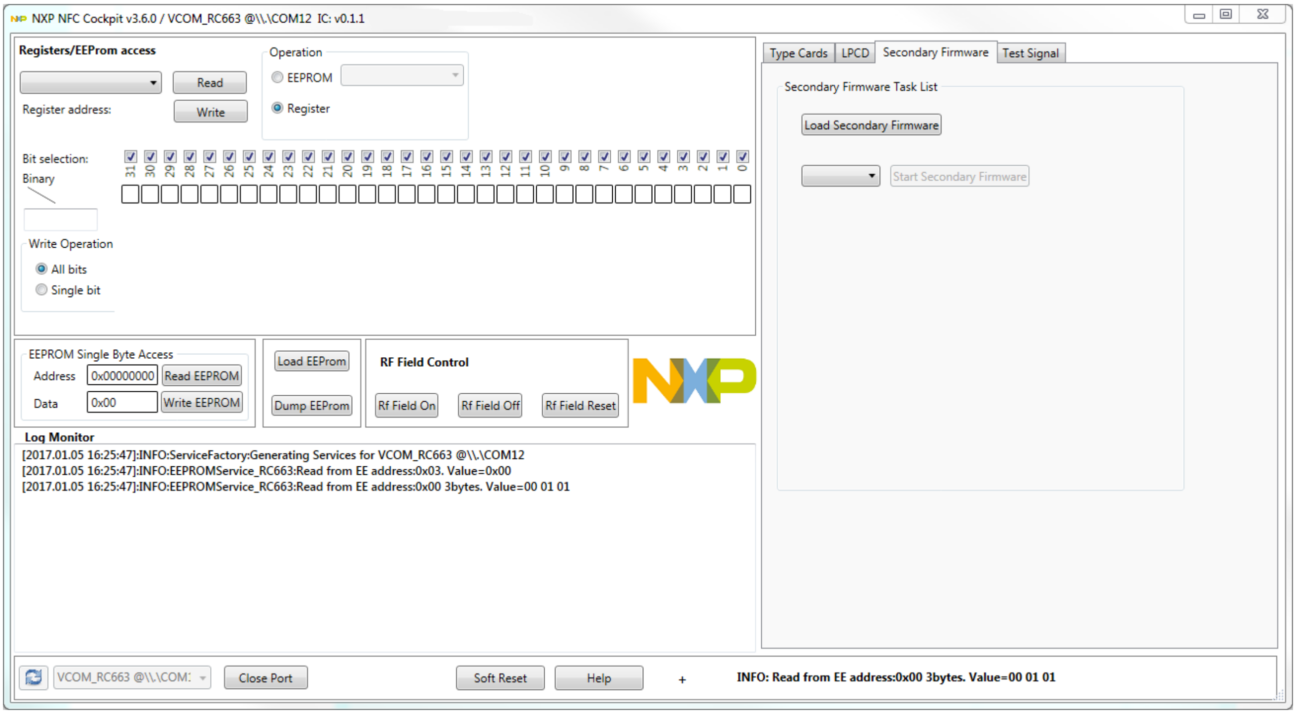
- NXL21986
  - Batteries
  - Biometric Devices
  - Bluetooth Radios
  - Computer
  - ControlVault Device
  - Disk drives
  - Display adapters
  - Human Interface Devices
  - IDE ATA/ATAPI controllers
  - Imaging devices
  - Intel(R) Dynamic Platform and Thermal Framework
  - Keyboards
  - Mice and other pointing devices
  - Monitors
  - Network adapters
  - Ports (COM & LPT)
    - ECP Printer Port (LPT1)
    - NxpNfcCockpit VCOM (COM12)
  - Processors
  - Proximity Devices
  - Smart card readers
  - Sound, video and game controllers
  - Storage controllers
  - System devices
  - Universal Serial Bus controllers

(1) CLEV663B connected to the PC, driver properly installed.

Fig 12. NFC Cockpit VCOM driver

3.2 NFC Cockpit

The NFC Cockpit can be installed and started (see Fig 13).



NXP NFC Cockpit v3.6.0 / VCOM_RC663 @\\.\COM12 IC: v0.1.1

Registers/EEProm access — Operation: ○ EEPROM ● Register

Write Operation: ● All bits ○ Single bit

EEPROM Single Byte Access — Address: 0x00000000 — Data: 0x00

Secondary Firmware Task List

Log Monitor

```
[2017.01.05 16:25:47]:INFO:ServiceFactory:Generating Services for VCOM_RC663 @\\.\COM12
[2017.01.05 16:25:47]:INFO:EEPROMService_RC663:Read from EE address:0x03. Value=0x00
[2017.01.05 16:25:47]:INFO:EEPROMService_RC663:Read from EE address:0x00 3bytes. Value=00 01 01
```

VCOM_RC663 @\\.\COM:

INFO: Read from EE address:0x00 3bytes. Value=00 01 01

(1) Status when starting the NFC Cockpit with connected CLEV663B board

Fig 13. NFC Cockpit Initial view with CLRC663 board

After starting the NFC Cockpit, the communication link between the PC and the CLEV663B (via the LPC VCOM interface) is enabled automatically.

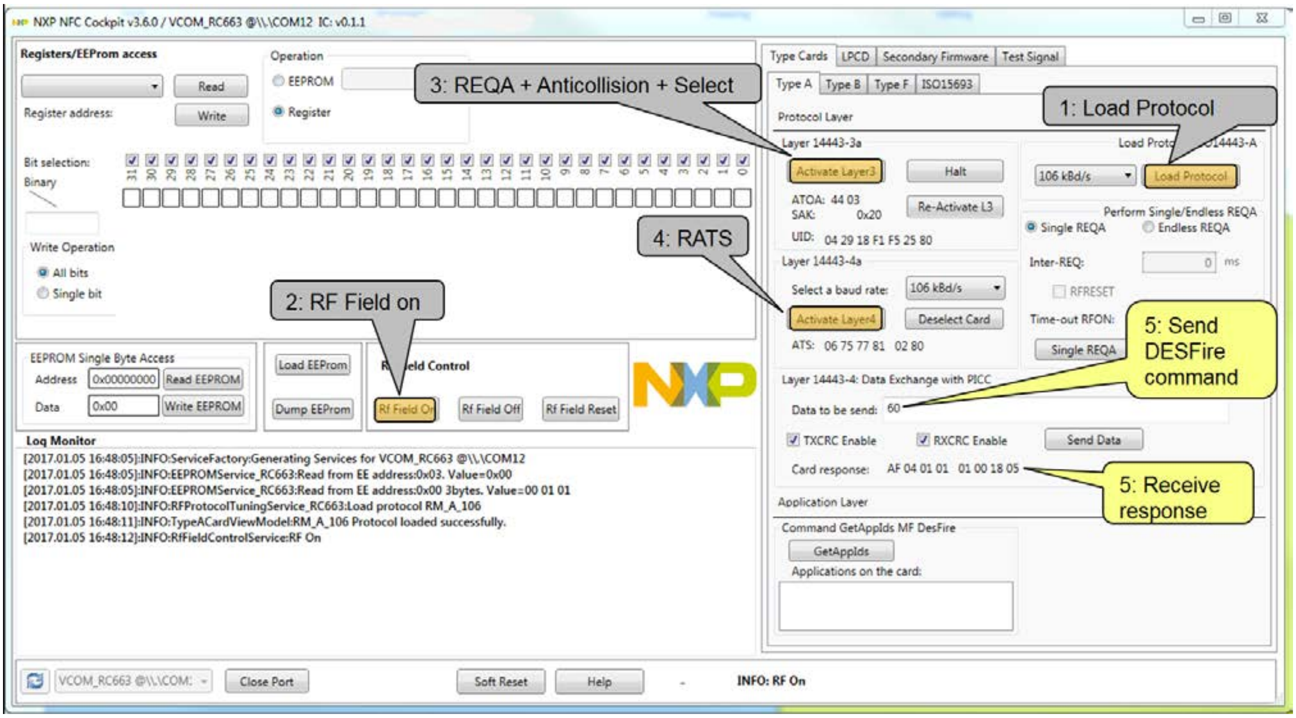
Note: The NFC Cockpit is a development tool, and therefore allows many different kind of operations, even “useless” ones at a first glance. The correct use of the NFC Cockpit is required to operate the CLRC663 properly.

Example: without enabling the RF Field no card can be operated, even though the CLRC663 can be operated.

The Fig 14 shows the activation of a MIFARE DESFire card, using the <Load Protocol> + <Field On> + <Activate Layer3>, followed by <Activate Layer4>. The NFC Cockpit shows the card responses like ATQA, SAK, and ATS.

Afterwards the ISO/IEC 14443-4 protocol can be used to exchange data. The Fig 14 shows the MIFARE DESFire command “Get Application ID” (0x6A), which returns the AIDs.

Note: Make sure that either the CRC is enabled or added manually in the data field.



Address: 0x00000000
Data: 0x00

ATOA: 44 03
SAK: 0x20
UID: 04 29 18 F1 F5 25 80
Select a baud rate: 106 kBd/s
ATS: 06 75 77 81 02 80
Load Protocol 14443-A: 106 kBd/s
Inter-REQ: 0 ms
Time-out RFON:
Data to be send: 60
Card response: AF 04 01 01 01 00 18 05
VCOM_RC663 @\\.\COM:

```
[2017.01.05 16:48:05]:INFO:ServiceFactory:Generating Services for VCOM_RC663 @\\.\COM12
[2017.01.05 16:48:05]:INFO:EEPROMService_RC663:Read from EE address:0x03. Value=0x00
[2017.01.05 16:48:05]:INFO:EEPROMService_RC663:Read from EE address:0x00 3bytes. Value=00 01 01
[2017.01.05 16:48:10]:INFO:RFProtocolTuningService_RC663:Load protocol RM_A_106
[2017.01.05 16:48:11]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 16:48:12]:INFO:RfFieldControlService:RF On
```

INFO: RF On

(2) 0x60 = Get Version command of MIFARE DESFire EV1

Fig 14. NFC Cockpit with CLEV663B: Activation of a MIFARE DESFire EV1 card + Get Version

Similar functionality does exist for ISO/IEC 14443 A and B, for NFC type F and for ISO/IEC 15693 communication.

Be aware that a Load Protocol command must be executed manually before the corresponding protocol settings are loaded from the EEPROM into the registers. So this tab “Type A” can be used to perform

(1) <Load Protocol> (e.g. type A 106)
(2) <Field On>
(3) <Single REQA> (using the EEPROM settings)
(4) Select a TX register, e.g. DRVMODREG, change TXCLOCKMODE
(5) Change some register bits, and write back into RAM
(6) <Single REQA> shows the register changes (probing the field and checking the envelop)

This allows an easy and quick optimization of Tx and Rx parameters. Using the default settings from the EEPROM always resets the relevant registers.

(7) <Load Protocol> (e.g. type A 106)
(8) <Single REQA> (using again the EEPROM settings)

Note: The EEPROM of the CLRC663 is locked for all the LoadProtocol area.

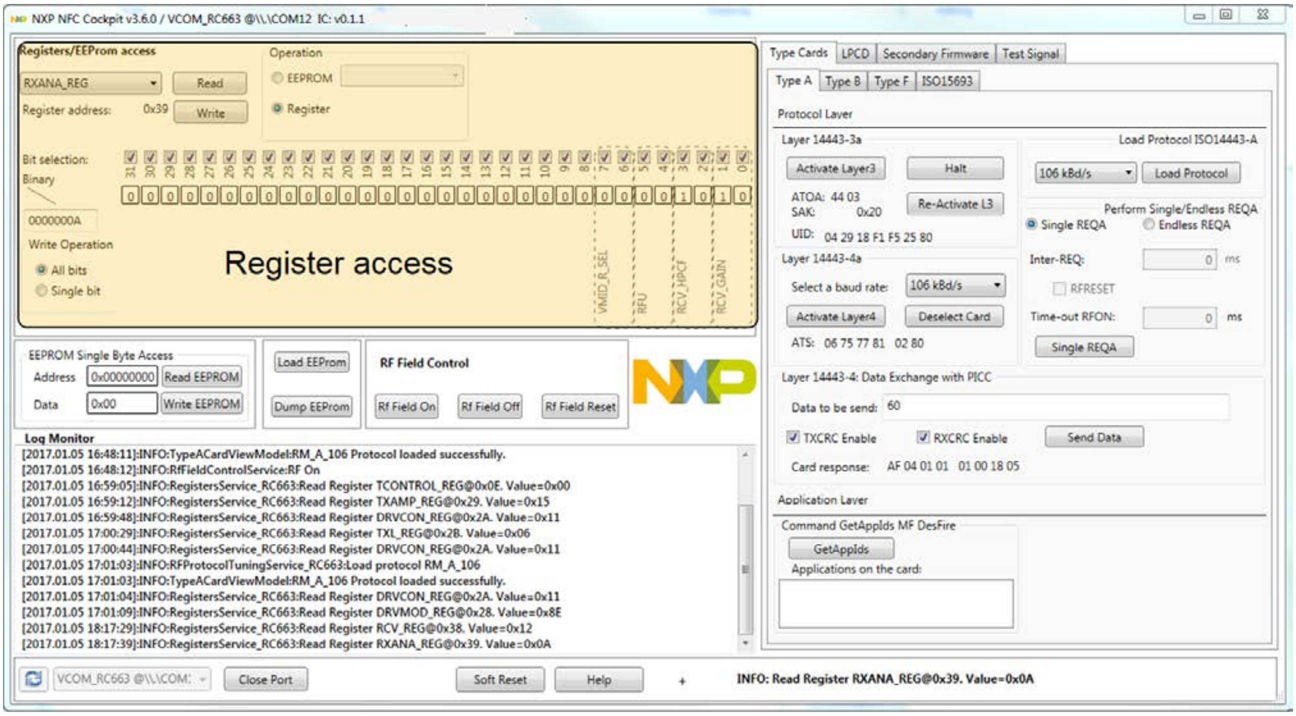
3.2.1 CLRC663 Register access

The NFC Cockpit allows the reading and writing of all the CLRC663 registers (see Fig 15).

Selecting a register reads and shows the hexadecimal content as well as the corresponding bit values. The input allows to change each bit separately as well as writing hexadecimal values. Writing back the value changes the PN5180 register.

A help function automatically shows a short description of the (part of the) registers itself, if the mouse is moved over the names.

Note: Some register content cannot be changed manually ("read only") and some content might be overwritten by the PN5180 firmware.


NXP NFC Cockpit v3.6.0 / VCOM_RC663 @\\.\COM12 IC: v0.1.1

Registers/EEProm access

RXANA_REG

Register address: 0x39

Operation: ○ EEPROM ● Register

Bit selection: ☑ 31 ☑ 30 ☑ 29 ☑ 28 ☑ 27 ☑ 26 ☑ 25 ☑ 24 ☑ 23 ☑ 22 ☑ 21 ☑ 20 ☑ 19 ☑ 18 ☑ 17 ☑ 16 ☑ 15 ☑ 14 ☑ 13 ☑ 12 ☑ 11 ☑ 10 ☑ 9 ☑ 8 ☑ 7 ☑ 6 ☑ 5 ☑ 4 ☑ 3 ☑ 2 ☑ 1 ☑ 0

Binary: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0000000A

Write Operation: ● All bits ○ Single bit

EEPROM Single Byte Access

Address: 0x00000000

Data: 0x00

Type A — Protocol Layer

Layer 14443-3a

ATOA: 44 03

SAK: 0x20

UID: 04 29 18 F1 F5 25 80

Load Protocol ISO14443-A: 106 kBd/s

Perform Single/Endless REQA: ● Single REQA ○ Endless REQA

Inter-REQ: 0 ms

□ RFRESET

Time-out RFON: 0 ms

Layer 14443-4a

Select a baud rate: 106 kBd/s

ATS: 06 75 77 81 02 80

Layer 14443-4: Data Exchange with PICC

Data to be send: 60

☑ TXCRC Enable ☑ RXCRC Enable

Card response: AF 04 01 01 01 00 18 05

Application Layer

Command GetAppIds MF DesFire

Applications on the card:

Log Monitor

```
[2017.01.05 16:48:11]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 16:48:12]:INFO:RfFieldControlService:RF On
[2017.01.05 16:59:05]:INFO:RegistersService_RC663:Read Register TCONTROL_REG@0x0E. Value=0x00
[2017.01.05 16:59:12]:INFO:RegistersService_RC663:Read Register TXAMP_REG@0x29. Value=0x15
[2017.01.05 16:59:48]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:00:29]:INFO:RegistersService_RC663:Read Register TXL_REG@0x2B. Value=0x06
[2017.01.05 17:00:44]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:03]:INFO:RFProtocolTuningService_RC663:Load protocol RM_A_106
[2017.01.05 17:01:03]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 17:01:04]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:09]:INFO:RegistersService_RC663:Read Register DRVMOD_REG@0x28. Value=0x8E
[2017.01.05 18:17:29]:INFO:RegistersService_RC663:Read Register RCV_REG@0x38. Value=0x12
[2017.01.05 18:17:39]:INFO:RegistersService_RC663:Read Register RXANA_REG@0x39. Value=0x0A
```

VCOM_RC663 @\\.\COM:

INFO: Read Register RXANA_REG@0x39. Value=0x0A

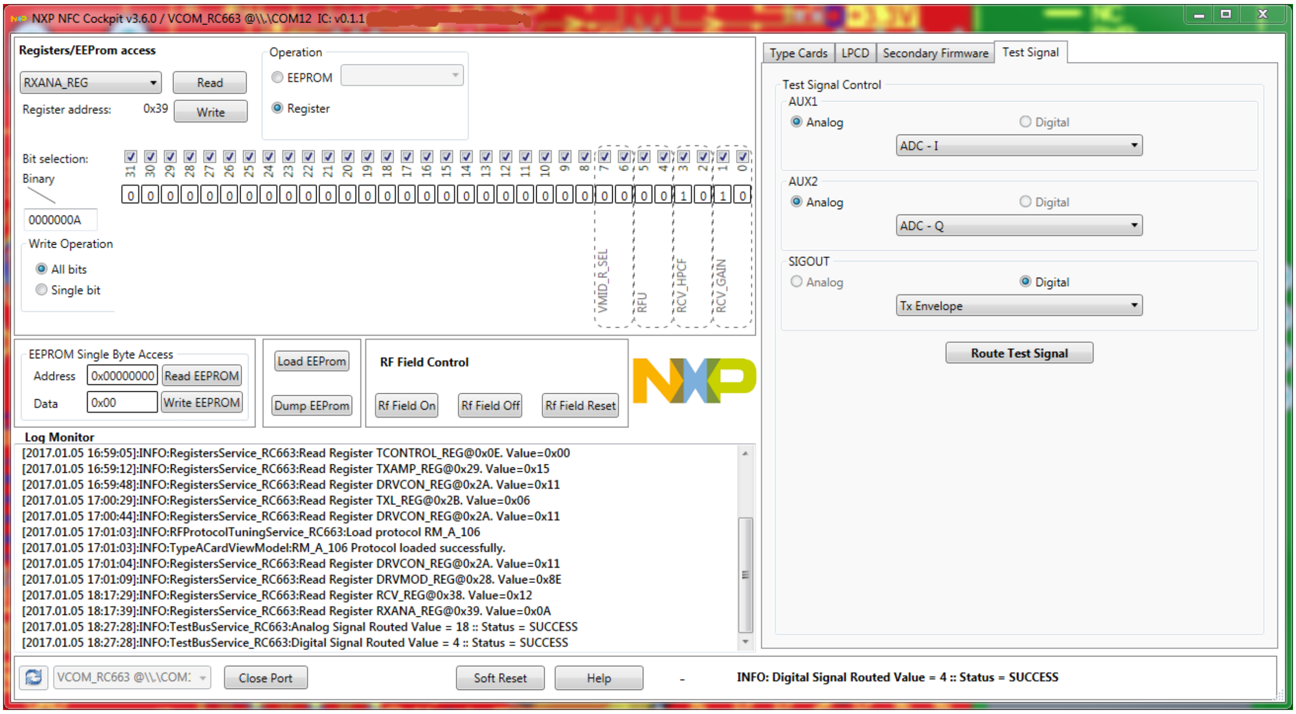
(1) Registers are temporary stored, i.e. might be overwritten with Load Protocol.

Fig 15. CLRC663 register access

3.2.2 CLRC663 analog and digital test signals

The NFC Cockpit allows to route the CLRC663 digital test signals to the SIGOUT pin, as well as to unlock and route the CLRC663 analog test signals to testpins AUX1 and AUX2, as shown in Fig 16.

The digital test pin SIGOUT can be found at the J301 (pin row), while the analog signals are routed to two test pads as close to the CLRC663 as possible (below the antenna tuning area).



NXP NFC Cockpit v3.6.0 / VCOM_RC663 @\\.\COM12 IC: v0.1.1 [redacted]

Registers/EEProm access

RXANA_REG ▾ Read

Register address: 0x39 Write

Operation: ○ EEPROM ● Register

Bit selection: ☑ 31 … ☑ 0

Binary: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0000000A

Write Operation: ● All bits ○ Single bit

EEPROM Single Byte Access: Address 0x00000000, Data 0x00

Test Signal Control

AUX1: ● Analog ○ Digital — ADC - I

AUX2: ● Analog ○ Digital — ADC - Q

SIGOUT: ○ Analog ● Digital — Tx Envelope

Log Monitor

```
[2017.01.05 16:59:05]:INFO:RegistersService_RC663:Read Register TCONTROL_REG@0x0E. Value=0x00
[2017.01.05 16:59:12]:INFO:RegistersService_RC663:Read Register TXAMP_REG@0x29. Value=0x15
[2017.01.05 16:59:48]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:00:29]:INFO:RegistersService_RC663:Read Register TXL_REG@0x2B. Value=0x06
[2017.01.05 17:00:44]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:03]:INFO:RFProtocolTuningService_RC663:Load protocol RM_A_106
[2017.01.05 17:01:03]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 17:01:04]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:09]:INFO:RegistersService_RC663:Read Register DRVMOD_REG@0x28. Value=0x8E
[2017.01.05 18:17:29]:INFO:RegistersService_RC663:Read Register RCV_REG@0x38. Value=0x12
[2017.01.05 18:17:39]:INFO:RegistersService_RC663:Read Register RXANA_REG@0x39. Value=0x0A
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Analog Signal Routed Value = 18 :: Status = SUCCESS
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Digital Signal Routed Value = 4 :: Status = SUCCESS
```

VCOM_RC663 @\\.\COM1

INFO: Digital Signal Routed Value = 4 :: Status = SUCCESS

Fig 16. CLRC663 analog and digital test signals

After selecting the signals <Route Test Signal> activates the chosen test signals at the chosen test pins.

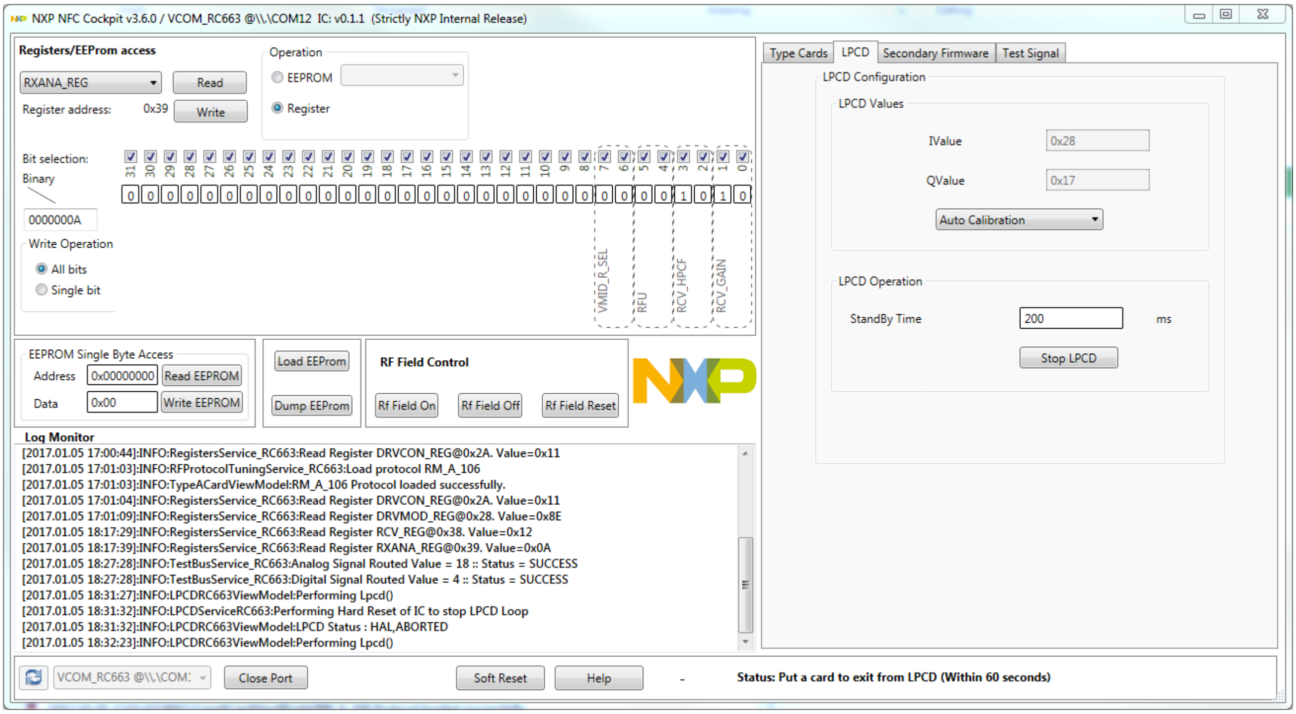
3.2.3 CLRC663 Low power card detection

The NFC Cockpit allows the configuration and test of the Low Power Card Detection (LPCD) of the CLRC663 as shown in Fig 17.

The LPCD parameter, which are used to define the LPCD performance (sensitivity versus robustness) can be entered manually, if needed (details refer to [1]).

Otherwise the standby time can be entered and the LPCD can be started. During the LPCD being activated the CLRC663 does not react on any command, so only a detuning (-> place a card) or a Reset (press <Stop LPCD>) can end the LPCD mode.

Note: The NFC Cockpit automatically stops the LPCD after 60 seconds.



NXP NFC Cockpit v3.6.0 / VCOM_RC663 @\\.\COM12 IC: v0.1.1 (Strictly NXP Internal Release)

Registers/EEProm access

RXANA_REG

Register address: 0x39

Operation: ○ EEPROM ● Register

Bit selection: ☑ 31 … ☑ 0

Binary: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0000000A

Write Operation: ● All bits ○ Single bit

EEPROM Single Byte Access: Address 0x00000000, Data 0x00

LPCD Configuration — LPCD Values: IValue 0x28, QValue 0x17, Auto Calibration

LPCD Operation: StandBy Time 200 ms

Log Monitor

```
[2017.01.05 17:00:44]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:03]:INFO:RFProtocolTuningService_RC663:Load protocol RM_A_106
[2017.01.05 17:01:03]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 17:01:04]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:09]:INFO:RegistersService_RC663:Read Register DRVMOD_REG@0x28. Value=0x8E
[2017.01.05 18:17:29]:INFO:RegistersService_RC663:Read Register RCV_REG@0x38. Value=0x12
[2017.01.05 18:17:39]:INFO:RegistersService_RC663:Read Register RXANA_REG@0x39. Value=0x0A
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Analog Signal Routed Value = 18 :: Status = SUCCESS
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Digital Signal Routed Value = 4 :: Status = SUCCESS
[2017.01.05 18:31:27]:INFO:LPCDRC663ViewModel:Performing Lpcd()
[2017.01.05 18:31:32]:INFO:LPCDServiceRC663:Performing Hard Reset of IC to stop LPCD Loop
[2017.01.05 18:31:32]:INFO:LPCDRC663ViewModel:LPCD Status : HAL,ABORTED
[2017.01.05 18:32:23]:INFO:LPCDRC663ViewModel:Performing Lpcd()
```

VCOM_RC663 @\\.\COM1

Status: Put a card to exit from LPCD (Within 60 seconds)

(1) LPCD had been started.

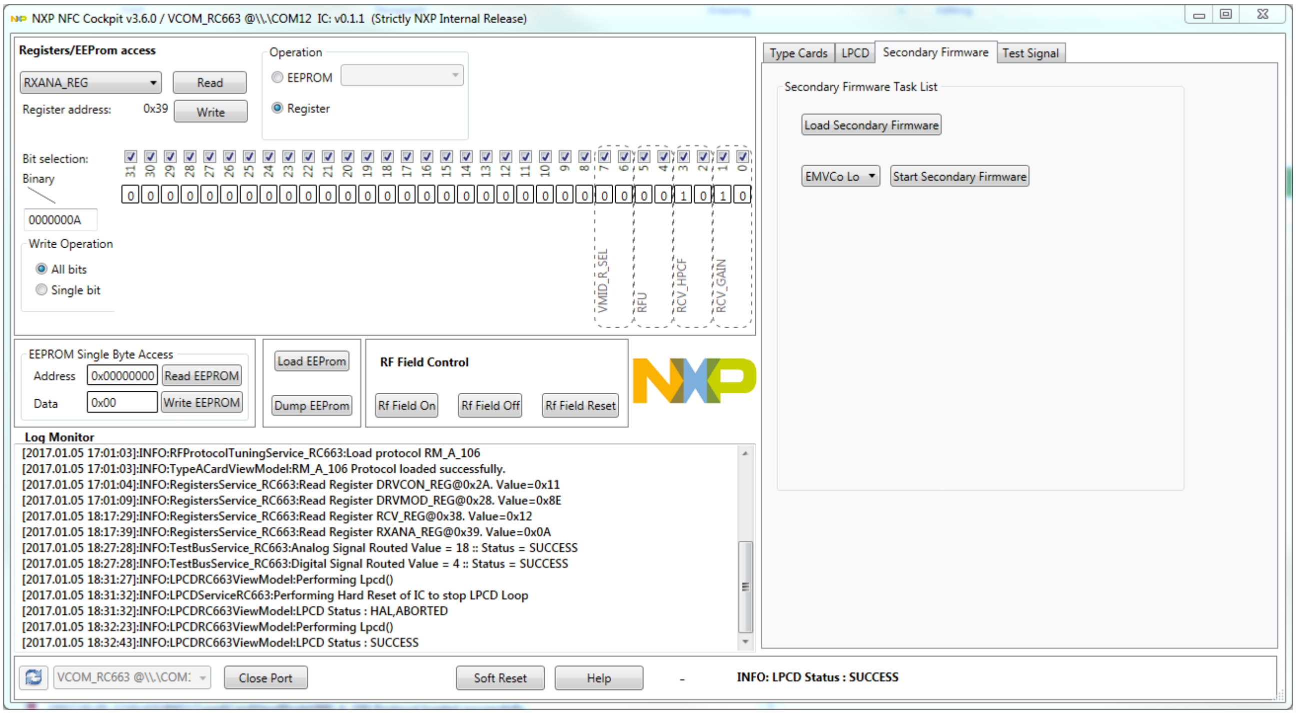
Fig 17. CLRC663 LPCD

3.2.4 Secondary Firmware options: EMVCo Loopback application

The NFC Cockpit offers the option to flash ("load") and start applications into the LPC µC. The default application is an EMVco Loopback function, but other samples are provided within the NFC Cockpit delivery package.

Each application can be easily flashed into the LPC by pressing the <Load Secondary Firmware>. The application then defines the user commands, as indicated in the NFC Cockpit.

The Fig 18 shows the default with the EMVCo Loopback which can be started and stopped.



NXP NFC Cockpit v3.6.0 / VCOM_RC663 @\\.\COM12 IC: v0.1.1 (Strictly NXP Internal Release)

Registers/EEProm access

RXANA_REG

Register address: 0x39

Operation: ○ EEPROM ● Register

Bit selection: ☑ 31 … ☑ 0

Binary: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0

0000000A

Write Operation: ● All bits ○ Single bit

EEPROM Single Byte Access

Address 0x00000000

Data 0x00

RF Field Control

Secondary Firmware Task List

EMVCo Lo

Log Monitor

```
[2017.01.05 17:01:03]:INFO:RFProtocolTuningService_RC663:Load protocol RM_A_106
[2017.01.05 17:01:03]:INFO:TypeACardViewModel:RM_A_106 Protocol loaded successfully.
[2017.01.05 17:01:04]:INFO:RegistersService_RC663:Read Register DRVCON_REG@0x2A. Value=0x11
[2017.01.05 17:01:09]:INFO:RegistersService_RC663:Read Register DRVMOD_REG@0x28. Value=0x8E
[2017.01.05 18:17:29]:INFO:RegistersService_RC663:Read Register RCV_REG@0x38. Value=0x12
[2017.01.05 18:17:39]:INFO:RegistersService_RC663:Read Register RXANA_REG@0x39. Value=0x0A
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Analog Signal Routed Value = 18 :: Status = SUCCESS
[2017.01.05 18:27:28]:INFO:TestBusService_RC663:Digital Signal Routed Value = 4 :: Status = SUCCESS
[2017.01.05 18:31:27]:INFO:LPCDRC663ViewModel:Performing Lpcd()
[2017.01.05 18:31:32]:INFO:LPCDServiceRC663:Performing Hard Reset of IC to stop LPCD Loop
[2017.01.05 18:31:32]:INFO:LPCDRC663ViewModel:LPCD Status : HAL,ABORTED
[2017.01.05 18:32:23]:INFO:LPCDRC663ViewModel:Performing Lpcd()
[2017.01.05 18:32:43]:INFO:LPCDRC663ViewModel:LPCD Status : SUCCESS
```

VCOM_RC663 @\\.\COM:

INFO: LPCD Status : SUCCESS

(1) Flash and start the EMVCo application

Fig 18. NFC Cockpit with EMVCo Loopback App

4. First time use

Make sure the LPC1769 is flashed with the correct Secondary FW (default after delivery). Check [5] in case, the CLEV663 had been used with customized code before, and the proper secondary firmware has to be flashed.

4.1 Jumper settings

The default jumper settings allow a direct use with the USB connector only. This might show limited performance due to a current limitation on the USB host. So for real performance measurements the external power supply should be used.

4.1.1 USB only

The jumper settings as shown in Fig 10 provide the default settings, using only USB for power supply (no external supply required).

4.1.2 External power supply

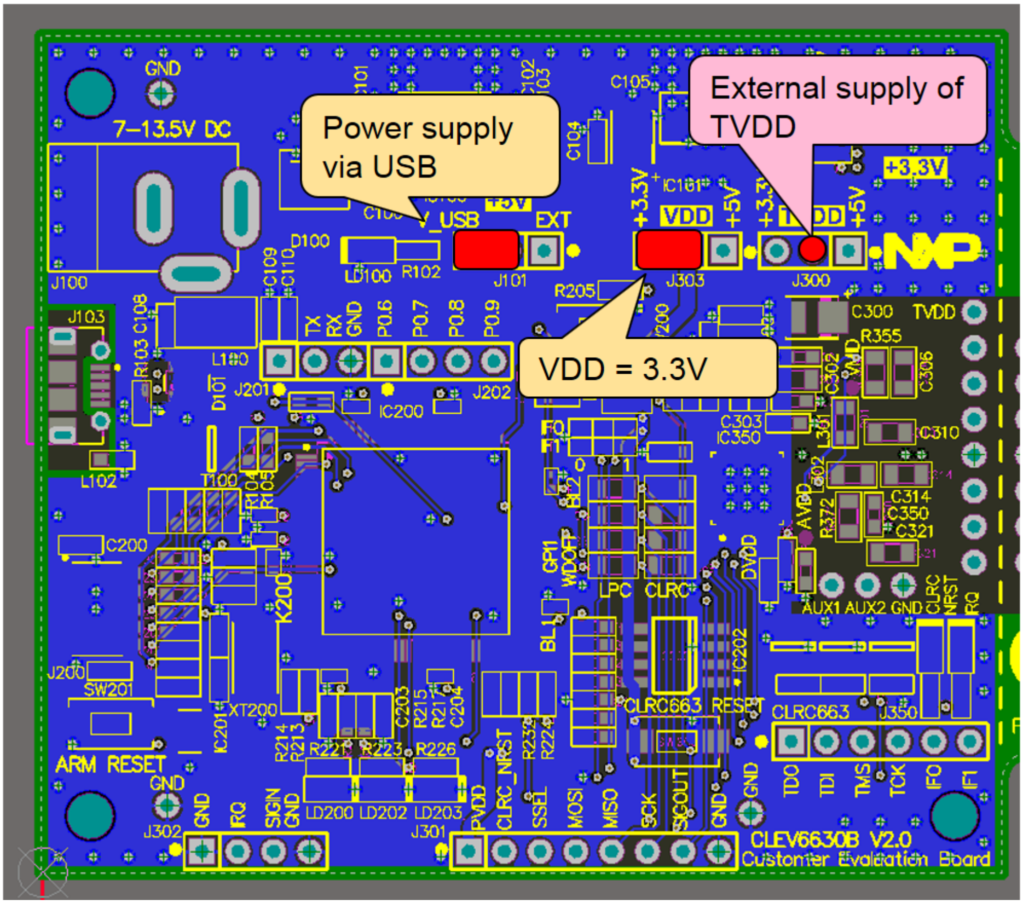
For the use of an external power supply the jumper J101 must be changed as shown in Fig 11.

The external power supply must provide a voltage level of V_{ext} = 7… 12V with 500mA.

For some of the analog tests (i.e. measuring ITVDD) it might be useful to only power the TVDD supply externally. This can be done using the jumper JP300, as shown in Fig 19.

Either the jumper can be replaced with a DC ampere meter to measure the ITVDD, or an external 5Vdc power supply can be directly connected to the center pin of JP300.

Note: Several GND pins are provided on the board. They all are connected to the same GND plane.



(1) JP300 can be used to externally supply TVDD.

Fig 19. CLEV663B jumper settings with external TVDD

5. Managing the CLRC663 SW projects with MCUXpresso IDE

The CLRC663 SW projects are delivered in a *NFC Reader Library for CLRC663* package available through product page or trough DocStore in case of Export controlled version. Example projects can be build and run with MCUXpresso IDE.

The MCUXpresso IDE is a low-cost highly integrated software development environment for NXP's LPC microcontrollers and includes all the tools necessary to develop high-quality software solutions in a timely and cost effective fashion. MCUXpresso IDE is based on Eclipse and has many enhancements to simplify development with NXP LPC microcontrollers. It also features the industry-standard GNU tool chain, with a choice of a proprietary optimized C library or the standard “Newlib” library. The MCUXpresso IDE can build an executable of any size with full code optimization.

Designed for simplicity and ease of use, the MCUXpresso IDE provides software engineers a quick and easy way to develop their applications.

This tool can freely be downloaded from the MCUXpresso website [8]. Before one can download the software, it is necessary to create an account. Creating an account is free.

5.1 Development environment

To use CLEV663 prepared software package all components listed in the Table 1 are required.

Table 1. Development Environment

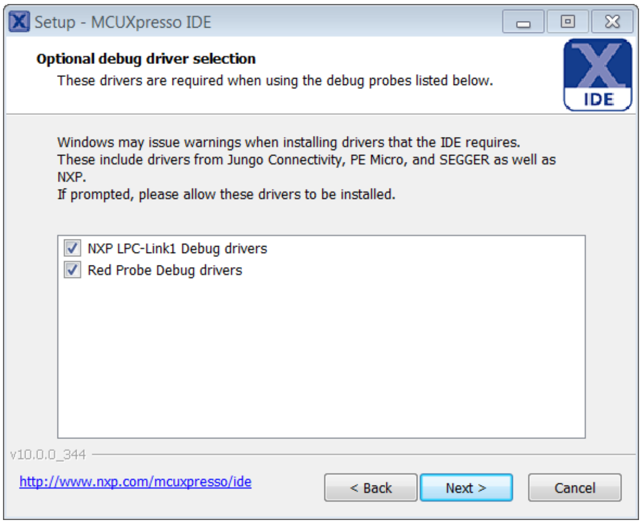
Item	Version	Description
CLEV6630B	1.0 or higher	CLEV6630B Customer Evaluation board (hardware)
LPC-Link 2	1.0	Standalone debug adaptor (hardware)
MCUXpresso IDE	10.0.0 or higher	Development IDE (PC software)

5.2 Installation procedure of the MCUXpresso IDE

The MCUXpresso IDE is installed into a single directory, of your choice. Unlike many software packages, the MCUXpresso IDE does not install or use any keys in the Windows Registry, or use or modify any environment variables (including PATH), resulting in a very clean installation that does not interfere with anything else on your PC. Should you wish to use the command-line tools, a command file is provided to set up the path for the local command window.

Multiple versions can be installed simultaneously without any issues.

The installation starts after double-clicking the installer file.



Setup - MCUXpresso IDE

Optional debug driver selection

These drivers are required when using the debug probes listed below.

Windows may issue warnings when installing drivers that the IDE requires.
These include drivers from Jungo Connectivity, PE Micro, and SEGGER as well as NXP.
If prompted, please allow these drivers to be installed.

☑ NXP LPC-Link1 Debug drivers
☑ Red Probe Debug drivers

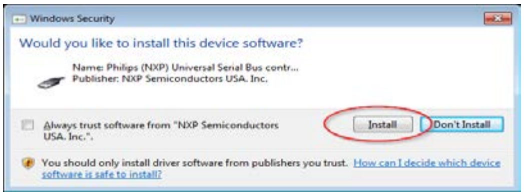
v10.0.0_344

http://www.nxp.com/mcuxpresso/ide

Fig 20. Check box for NXP debug drivers

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

During the installation, the user will be asked to install some required drivers. The installation of these drivers shall be accepted.



Windows Security

Would you like to install this device software?

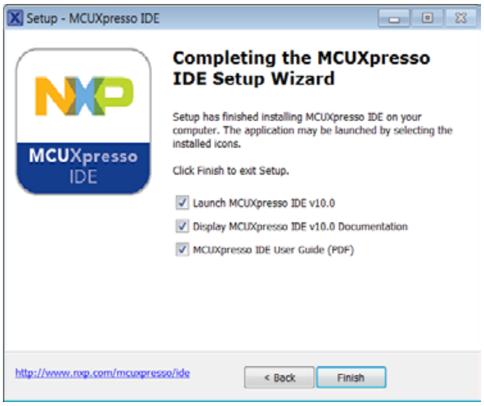
Name: Philips (NXP) Universal Serial Bus contr...
Publisher: NXP Semiconductors USA. Inc.

□ Always trust software from "NXP Semiconductors USA. Inc.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Fig 21. Windows security dialog

After the setup wizard, has finished, the newly installed IDE can be launched.



Setup - MCUXpresso IDE

Completing the MCUXpresso IDE Setup Wizard

Setup has finished installing MCUXpresso IDE on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

☑ Launch MCUXpresso IDE v10.0
☑ Display MCUXpresso IDE v10.0 Documentation
☑ MCUXpresso IDE User Guide (PDF)

http://www.nxp.com/mcuxpresso/ide

Fig 22. MCUXpresso IDE

5.3 Importing provided SW example projects

The use of quick start panel provides rapid access to the most commonly used features of the MCUXpresso IDE. Quickstart panel allows easy import projects, create new projects, build and debug projects.

The sequence of installing the software projects is indicated:

- Start the MCUXpresso IDE.
- Open new or dedicated workspace
- Select the option “Import project(s)” (see picture below).
- Browse the zip archive.
- MCUXpresso IDE unzips the software package.
- The software package is ready for use.

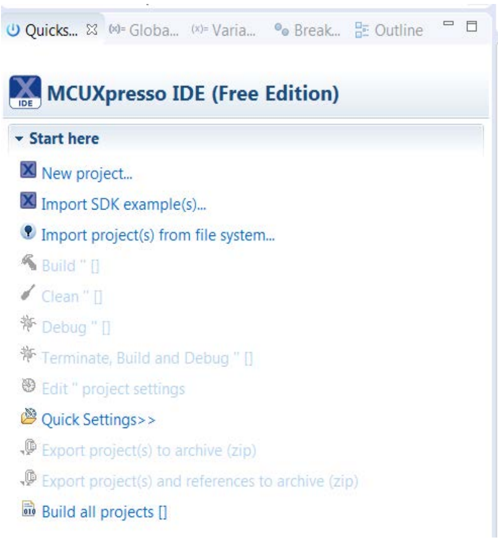
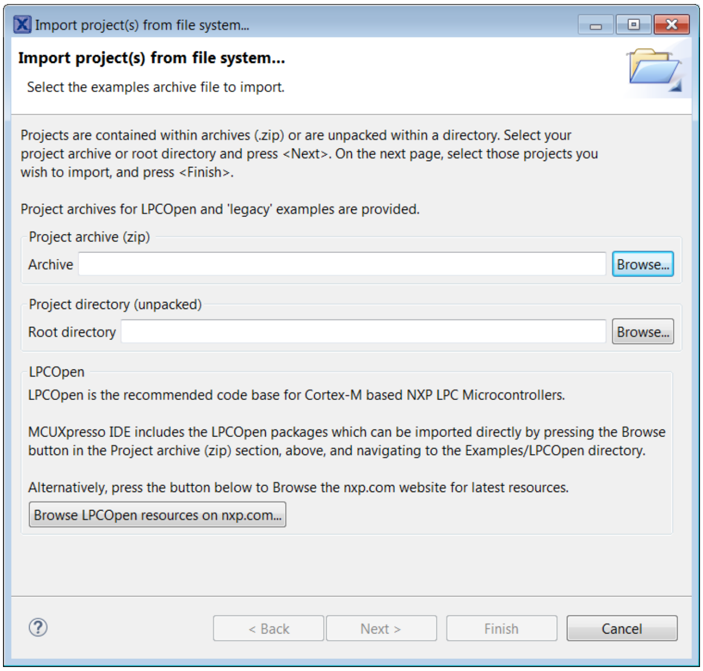


Fig 23. Importing project (1)

In the Quickstart panel on the left-hand side, choose "Import projects(s)".



Import project(s) from file system...

Import project(s) from file system...

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

Project directory (unpacked)

Root directory

LPCOpen

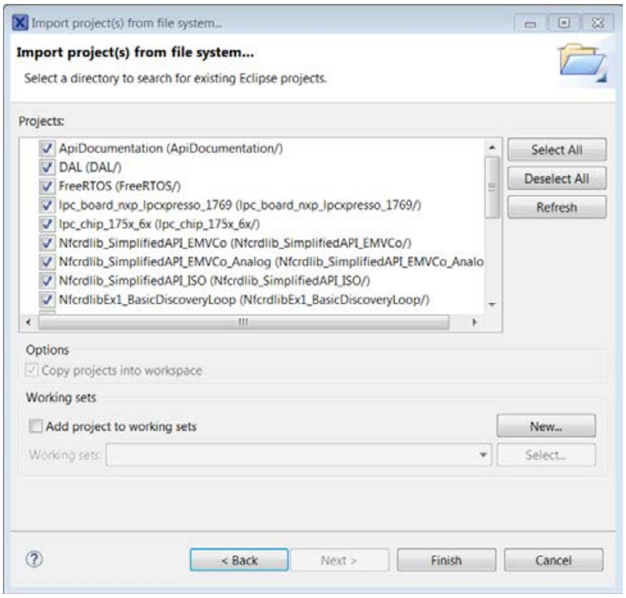
LPCOpen is the recommended code base for Cortex-M based NXP LPC Microcontrollers.

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.

Fig 24. Importing project (2)

Browse the desired package and click "Next".



Import project(s) from file system...

Import project(s) from file system...

Select a directory to search for existing Eclipse projects.

Projects:

- ☑ ApiDocumentation (ApiDocumentation/)
- ☑ DAL (DAL/)
- ☑ FreeRTOS (FreeRTOS/)
- ☑ lpc_board_nxp_lpcxpresso_1769 (lpc_board_nxp_lpcxpresso_1769/)
- ☑ lpc_chip_175x_6x (lpc_chip_175x_6x/)
- ☑ Nfcrdlib_SimplifiedAPI_EMVCo (Nfcrdlib_SimplifiedAPI_EMVCo/)
- ☑ Nfcrdlib_SimplifiedAPI_EMVCo_Analog (Nfcrdlib_SimplifiedAPI_EMVCo_Analo
- ☑ Nfcrdlib_SimplifiedAPI_ISO (Nfcrdlib_SimplifiedAPI_ISO/)
- ☑ NfcrdlibEx1_BasicDiscoveryLoop (NfcrdlibEx1_BasicDiscoveryLoop/)

Options

☑ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets:

Fig 25. Importing project (3)

For a working demo project, you need to import at least four sub projects. One example project, the NFC Reader Library, FreeRTOS, one chip library and one board library.

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

5.4 Building projects

Building projects in a workspace is a simple case of using the Quickstart Panel - ‘Build all projects’. Alternatively, a single project can be selected in the “Project Explorer View” and built separately. Note that building a single project may also trigger a build of any associated library projects.

The project can be built as shown in the Fig 26.

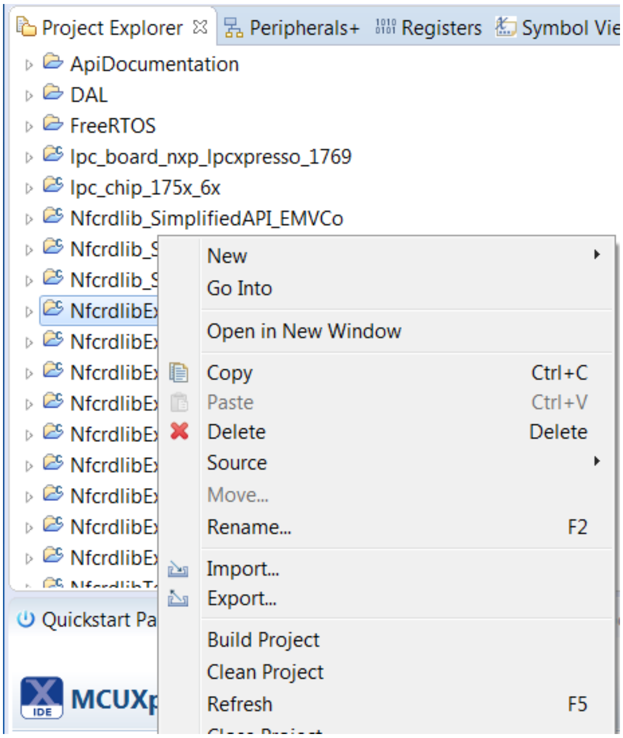


Fig 26. Building the project

As a part of the build output, the binary for the “User Flash” file is created. This binary file can also be used to update LPC1769 User Flash via USB mass storage interface.

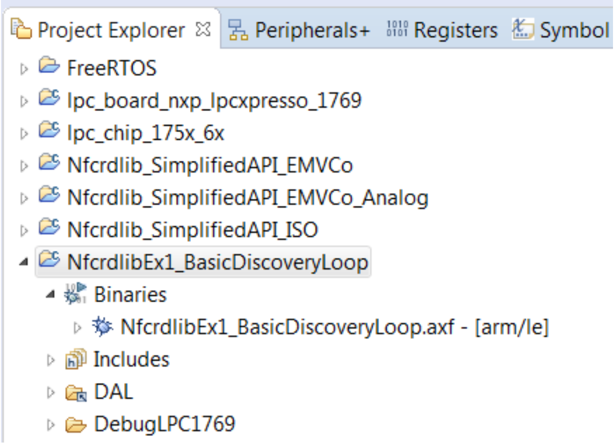

Project Explorer
Peripherals+
Registers
Symbol
FreeRTOS
lpc_board_nxp_lpcxpresso_1769
lpc_chip_175x_6x
Nfcrdlib_SimplifiedAPI_EMVCo
Nfcrdlib_SimplifiedAPI_EMVCo_Analog
Nfcrdlib_SimplifiedAPI_ISO
NfcrdlibEx1_BasicDiscoveryLoop
Binaries
NfcrdlibEx1_BasicDiscoveryLoop.axf - [arm/le]
Includes
DAL
DebugLPC1769

Fig 27. Figure title here

The project settings, compiler and link flags can be changed in the project properties dialog. To open the project properties dialog, select appropriate project in the “Project Explorer View” and click “Edit ‘selected-project’ project settings”.

5.5 Running and debugging a project

This description shows how to run the “*NfcrdlibEx1_CasicDiscoveryLoop*” example application for the CLEV6630B evaluation development board. The same basic principles will apply for all other examples. In cases where example will need additional configuration this will be detailed described in the example description.

Initially CLEV6630B evaluation board needs to be connected to the computer via LPC-Link 2, as shown in Fig 28.

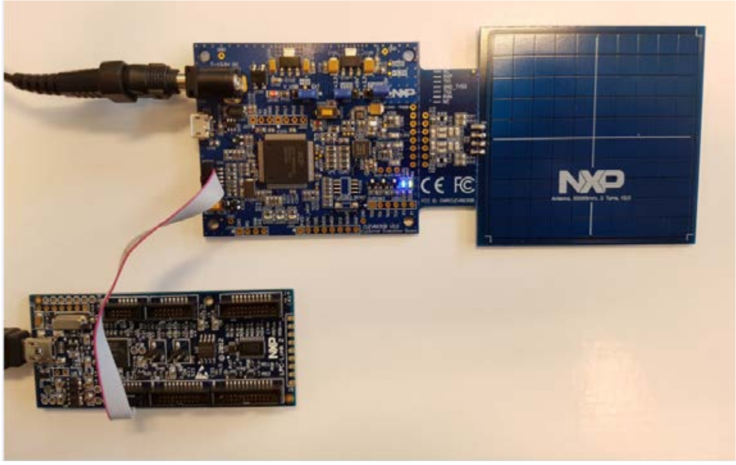


Fig 28. CLEV6630B with LPC-Link 2

When debug is started, the program is automatically downloaded to the target and it´s programmed to the LPC1769 flash memory; a default breakpoint is set on the first instruction in *main()*, the application is started (by simulating a processor reset), and code is executed until the default breakpoint is hit.

To start debugging your application on the CLEV6630B, simply highlight the project in the Project Explorer and then in the *Quickstart Panel* click Debug, as shown in Fig 29. The MCUXpresso IDE will first build application, flash application binary and then will start debug session.

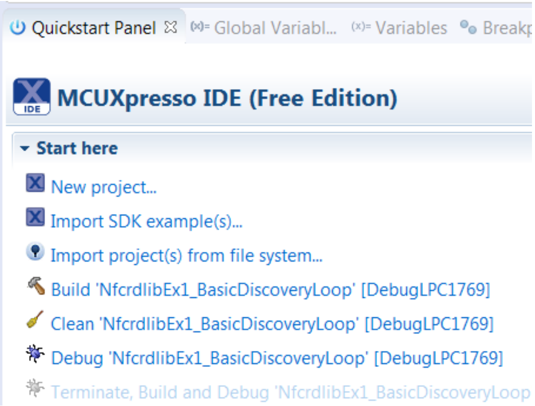
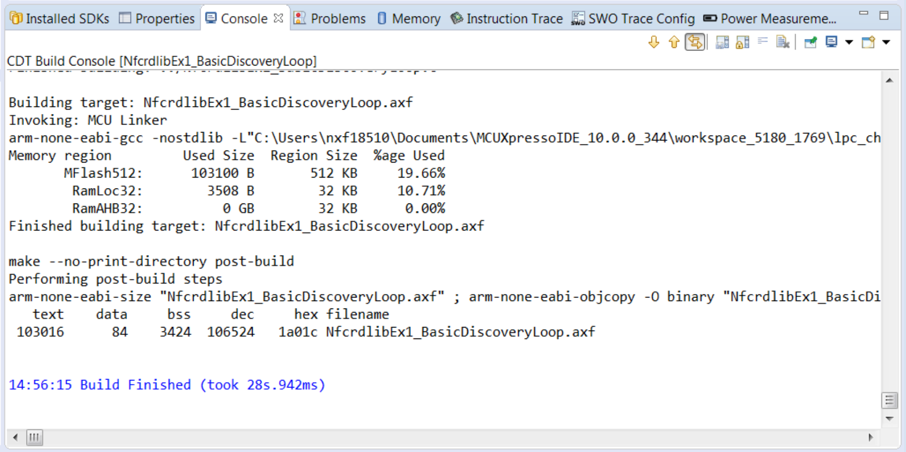

Quickstart Panel
Global Variabl...
Variables
Breakp
MCUXpresso IDE (Free Edition)
IDE
Start here
New project...
Import SDK example(s)...
Import project(s) from file system...
Build 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Clean 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Debug 'NfcrdlibEx1_BasicDiscoveryLoop' [DebugLPC1769]
Terminate, Build and Debug 'NfcrdlibEx1_BasicDiscoveryLoop

Fig 29. Launch debug session



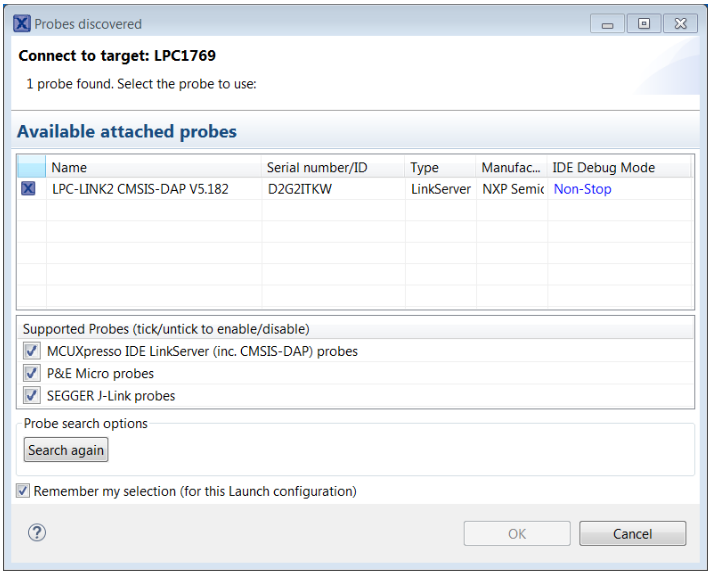
```
Building target: NfcrdlibEx1_BasicDiscoveryLoop.axf
Invoking: MCU Linker
arm-none-eabi-gcc -nostdlib -L"C:\Users\nxf18510\Documents\MCUXpressoIDE_10.0.0_344\workspace_5180_1769\lpc_ch
Memory region         Used Size  Region Size  %age Used
       MFlash512:      103100 B       512 KB     19.66%
        RamLoc32:        3508 B        32 KB     10.71%
        RamAHB32:          0 GB        32 KB      0.00%
Finished building target: NfcrdlibEx1_BasicDiscoveryLoop.axf

make --no-print-directory post-build
Performing post-build steps
arm-none-eabi-size "NfcrdlibEx1_BasicDiscoveryLoop.axf" ; arm-none-eabi-objcopy -O binary "NfcrdlibEx1_BasicDi
   text    data     bss     dec     hex filename
 103016      84    3424  106524   1a01c NfcrdlibEx1_BasicDiscoveryLoop.axf


14:56:15 Build Finished (took 28s.942ms)
```

Fig 30. Successful compile

Select “LPC-Link 2” as a debug probe.



Probes discovered

Connect to target: LPC1769

1 probe found. Select the probe to use:

Available attached probes

	Name	Serial number/ID	Type	Manufac...	IDE Debug Mode
☒	LPC-LINK2 CMSIS-DAP V5.182	D2G2ITKW	LinkServer	NXP Semic	Non-Stop

Supported Probes (tick/untick to enable/disable)

☑ MCUXpresso IDE LinkServer (inc. CMSIS-DAP) probes

☑ P&E Micro probes

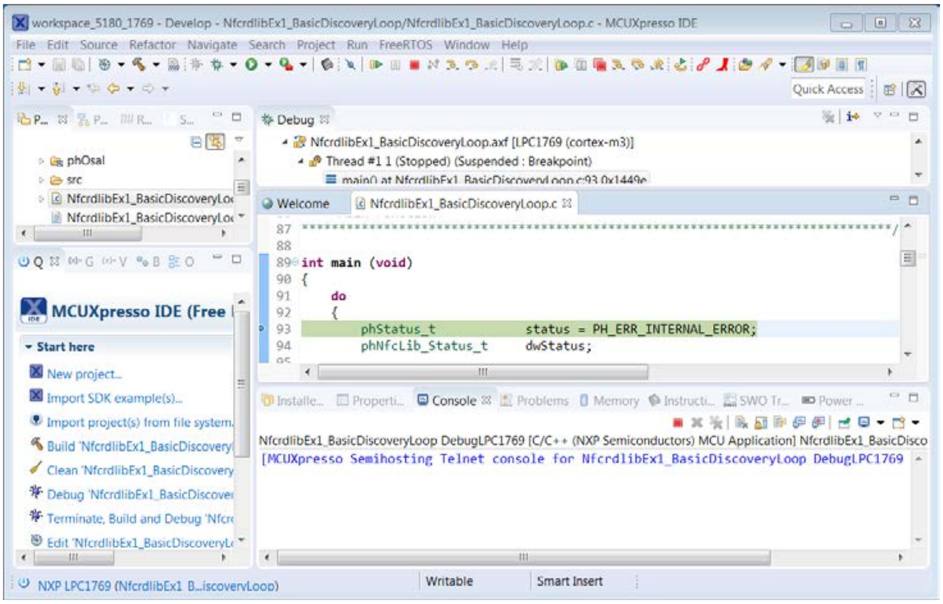
☑ SEGGER J-Link probes

Probe search options

☑ Remember my selection (for this Launch configuration)

Fig 31. Select the launch configuration

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



workspace_5180_1769 - Develop - NfcrdlibEx1_BasicDiscoveryLoop/NfcrdlibEx1_BasicDiscoveryLoop.c - MCUXpresso IDE

File
Edit
Source
Refactor
Navigate
Search
Project
Run
FreeRTOS
Window
Help
Quick Access
Debug
phOsal
src
NfcrdlibEx1_BasicDiscoveryLoo
NfcrdlibEx1_BasicDiscoveryLoo
MCUXpresso IDE (Free
Start here
New project...
Import SDK example(s)...
Import project(s) from file system...
Build 'NfcrdlibEx1_BasicDiscovery
Clean 'NfcrdlibEx1_BasicDiscovery
Debug 'NfcrdlibEx1_BasicDiscover
Terminate, Build and Debug 'Nfcr
Edit 'NfcrdlibEx1_BasicDiscoveryL
Welcome
NfcrdlibEx1_BasicDiscoveryLoop.c
Installe...
Properti...
Console
Problems
Memory
Instructi...
SWO Tr...
Power ...
NXP LPC1769 (NfcrdlibEx1_B...iscoveryLoop)
Writable
Smart Insert

```
NfcrdlibEx1_BasicDiscoveryLoop.axf [LPC1769 (cortex-m3)]
  Thread #1 1 (Stopped) (Suspended : Breakpoint)
    main() at NfcrdlibEx1_BasicDiscoveryLoop.c:93 0x1449e
```

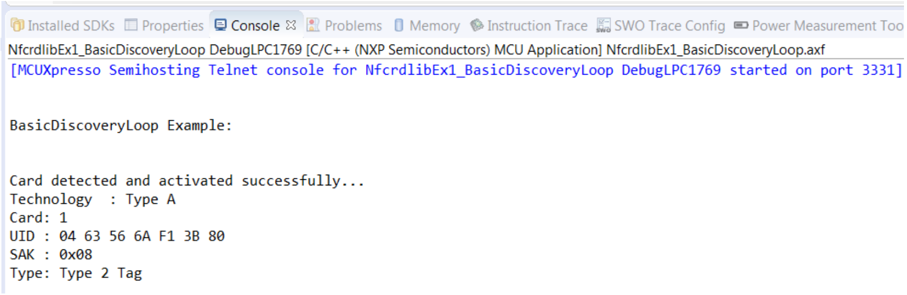
```
****************************************************************************/

int main (void)
{
    do
    {
        phStatus_t              status = PH_ERR_INTERNAL_ERROR;
        phNfcLib_Status_t       dwStatus;
```

```
NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 [C/C++ (NXP Semiconductors) MCU Application] NfcrdlibEx1_BasicDisco
[MCUXpresso Semihosting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769
```

Fig 32. Debug project

In the console window application debug outputs of the execution can be seen.



NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 [C/C++ (NXP Semiconductors) MCU Application] NfcrdlibEx1_BasicDiscoveryLoop.axf

```
[MCUXpresso Semihosting Telnet console for NfcrdlibEx1_BasicDiscoveryLoop DebugLPC1769 started on port 3331]


BasicDiscoveryLoop Example:


Card detected and activated successfully...
Technology  : Type A
Card: 1
UID : 04 63 56 6A F1 3B 80
SAK : 0x08
Type: Type 2 Tag
```

Fig 33. Application printouts in Console window

After the execution has reached the end of the main function please click the Terminate button to stop the execution. Otherwise rerun of the project will be possible.

Buttons in the debug toolbar provide next functionalities:

▶	Run the program.
↷	Step over C/C++ line.
↴	Step into a function.
■	Stop the debugger.
⏸	Pause execution of the running program.
i⇨	Instruction stepping mode (disassembly).

Fig 34. Debug Buttons

6. Associated projects

All example projects are available for download at the CLRC663 product page in the documents section and are being distributed in one single file.

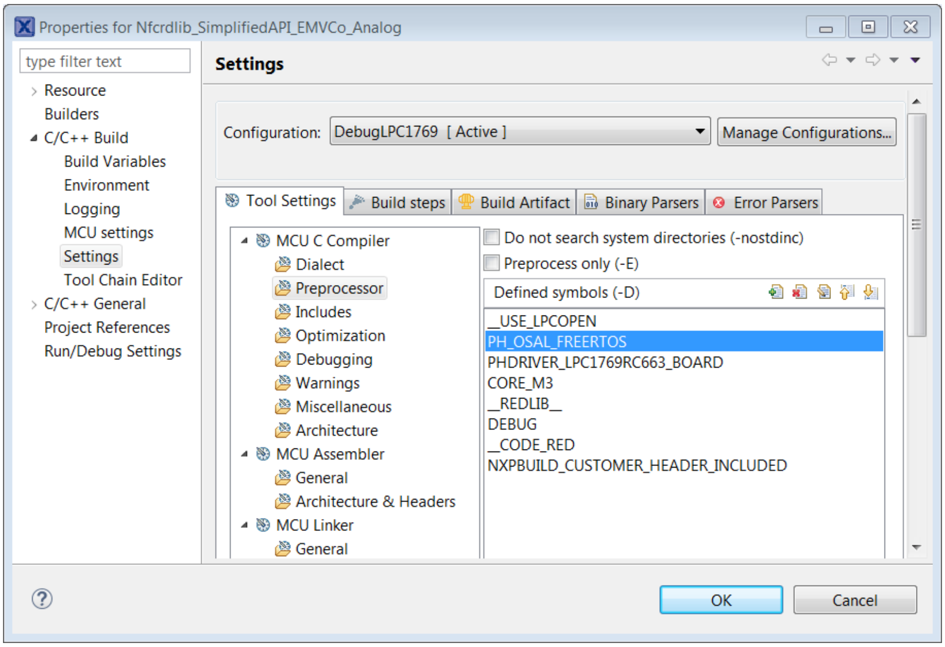
After downloading the zip file unzip it and run the installer. The installer makes a copy of all documents and SW on the hard disk.

By default, the projects are preconfigured to be run on the CLEV6630B development board. This is defined by preprocessor directive PHDRIVER_LPC1769RC663_BOARD (properties-> settings->preprocessor) and by defining appropriate macro in "*../intfs/ph_NxpBuild_App.h*".

```
//#define NXPBUILD__PHHAL_HW_PN5180
#define NXPBUILD__PHHAL_HW_RC663
```

Running the projects with, or without FreeRTOS

All projects described in the following sub chapters can be configured to run with or without FreeRTOS operating system. To enable or disable FreeRTOS support, define preprocessor directive `PH_OSAL_FREERTOS` or `PH_OSAL_NULLOS` respectively (Fig 35) and rebuild project.



Configuration: DebugLPC1769 [Active]

□ Do not search system directories (-nostdinc)

□ Preprocess only (-E)

Defined symbols (-D)

```
__USE_LPCOPEN
PH_OSAL_FREERTOS
PHDRIVER_LPC1769RC663_BOARD
CORE_M3
__REDLIB__
DEBUG
__CODE_RED
NXPBUILD_CUSTOMER_HEADER_INCLUDED
```

(1) Project properties -> Settings -> Preprocessor

Fig 35. Defining RTOS preprocessor symbols

6.1 Example 1 – Basic Discovery Loop

The Discovery Loop is the entry point when starting to communicate with an NFC tag or device. It scans the close environment for tags and devices of different technologies.

Example is implemented to work in POLL and LISTEN mode of the discovery loop. Information (like UID, SAK, and Product Type for MIFARE Cards) of the detected tags are printed out and it also prints information when it gets activated as a target by an external initiator/reader. Whenever multiple technologies are detected, example select first detected technology and resolve it.

In passive poll mode, Low Power Card Detection (LPCD) is enabled.

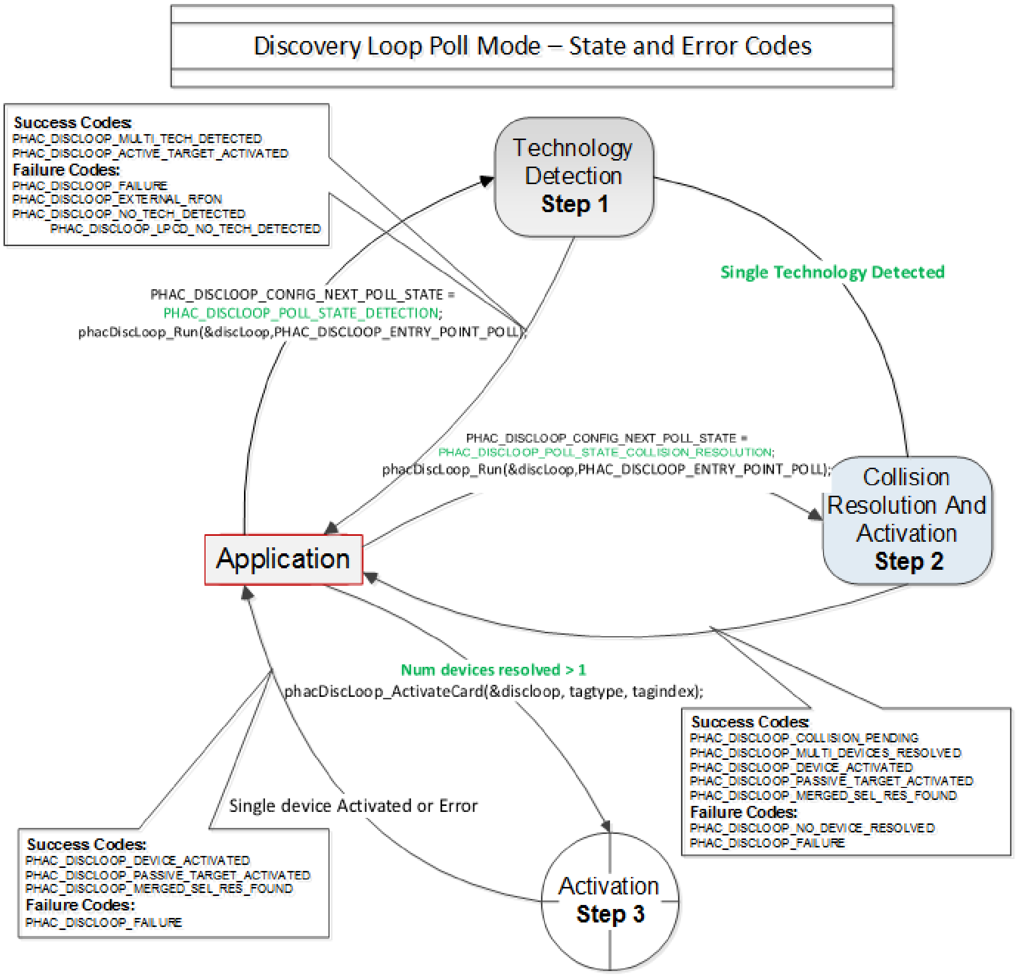

Discovery Loop Poll Mode – State and Error Codes
Success Codes:
PHAC_DISCLOOP_MULTI_TECH_DETECTED
PHAC_DISCLOOP_ACTIVE_TARGET_ACTIVATED
Failure Codes:
PHAC_DISCLOOP_FAILURE
PHAC_DISCLOOP_EXTERNAL_RFON
PHAC_DISCLOOP_NO_TECH_DETECTED
PHAC_DISCLOOP_LPCD_NO_TECH_DETECTED
Technology Detection Step 1
Single Technology Detected
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE =
PHAC_DISCLOOP_POLL_STATE_DETECTION;
phacDiscLoop_Run(&discLoop,PHAC_DISCLOOP_ENTRY_POINT_POLL);
PHAC_DISCLOOP_CONFIG_NEXT_POLL_STATE =
PHAC_DISCLOOP_POLL_STATE_COLLISION_RESOLUTION;
phacDiscLoop_Run(&discLoop,PHAC_DISCLOOP_ENTRY_POINT_POLL);
Collision Resolution And Activation Step 2
Application
Num devices resolved > 1
phacDiscLoop_ActivateCard(&discloop, tagtype, tagindex);
Success Codes:
PHAC_DISCLOOP_COLLISION_PENDING
PHAC_DISCLOOP_MULTI_DEVICES_RESOLVED
PHAC_DISCLOOP_DEVICE_ACTIVATED
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND
Failure Codes:
PHAC_DISCLOOP_NO_DEVICE_RESOLVED
PHAC_DISCLOOP_FAILURE
Single device Activated or Error
Success Codes:
PHAC_DISCLOOP_DEVICE_ACTIVATED
PHAC_DISCLOOP_PASSIVE_TARGET_ACTIVATED
PHAC_DISCLOOP_MERGED_SEL_RES_FOUND
Failure Codes:
PHAC_DISCLOOP_FAILURE
Activation Step 3

Fig 36. Discovery Loop in Poll mode

The core function of this example is “*BasicDiscoveryLoop_Demo()”,* where initialization of the NFC Reader library and polling for NFC technologies is implemented. After each polling loop, application is checking polling result and printout information about the detected tags or devices.

This example is using default DiscoveryLoop configuration, which enables all supported technologies and it is limited to one device for each technology.

Table 2. Supported technologies

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

6.2 Example 2 – Advanced Discovery Loop

Additionally, to Example 1 the Advanced Discovery Loop example explains the different configuration options of the Discovery Loop and configure DiscoveryLoop with default values based on the interested profile, NFC or EMVCo.

The configuration of the “DiscoveryLoop” is implemented in "*LoadProfile()*” function.

6.3 Example 4 – MIFARE Classic

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

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

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

6.4 Example 5 - ISO15693

Similar to the previous example, this one is also using only one technology, in that case ISO15693. “*DiscoveryLoop*” is configured to resolve only one device and in the example it is shown how to change Tx Guard Time for T5T cards, this is implemented in “phApp_Init()” function.

Once ICODE SLI is resolved and activated, application printout card information like type of the card and UID, and it will read and write from/to the memory block.

This example is good start in case of working with only one card or to see how to manage ISO15693 type of the cards.

For a much more extensive example, demonstrating the use of ISO/IEC 15693 and ISE/IEC 18000-3 Mode 3 tags (ICODE SLI and ICODE ILT). In order to assure ICODE SLI and ILT detection please check HAL digital delay define settings as described in chapter 4.

6.5 Example 7 – EMVCo Polling

The EMVCo Polling example it is demonstrated how to configure NFC Reader Library as specified by EMVCo specifications and starts polling for EMVCo cards.

Once an EMVCo compatible card is resolved and activated, it demonstrates the exchange of APDU commands. This example shall help the developers getting started more quickly when working with EMVCo cards.

6.6 Example 9 – NTAG-I2C

The NTAG-I2C example demonstrates the use of special features which are supported by NTAG-I2C. By using POLL mode of the discovery loop, example detect the NTag I2C cards and displays detected tag information like UID, ATQA, SAK, Version info and perform “*Page Read*” and “*PageWrite*” commands.

For more details about the NTAG-I2C and its functionalities please consult the related product page.

6.7 Example 10 – MIFARE DESFire

The MIFARE DESFire example demonstrates how to use MIFARE DESFire EV1 cards.

Once MIFARE DESFire card is resolved and activated, it displays MIFARE DESFire applications created by this example previously and it displays 32bit signed integer which is incremented after each successful detection of tag.

In case no application is present on the tag, new application will be created with two new files to hold NXPNFCRDLIB version used to create this application and another file to hold 32bit signed integer.

Note: This example including the required modules of the NFC Reader Library is only available via NXP Docstore.

6.8 Example 11 – ISO10373 PCD

This example is used to perform ISO 10373-6 PCD compliance validation. This example has to be executed in the DUT which has an ISO 14443 based PCD implementation. The ISO 10373-6 test methods verifies the compliance to the ISO 14443 protocols. An external tool like Micropross MP300 implements the test methods for the ISO 10373-6 and is used as the counterpart for this testing.

6.9 Test Example 12 – RC663LPCD

This example is a test suite application to test RC663 LPCD. This test suite contains test cases for RC663 HAL LPCD under different conditions. Test Cases comprises of combinations of Digital Filter, Charge Pump and Detection Options. The scenarios/combinations are as follows:

- *Scenarios 1 ==> Digital Filter: Disabled; Charge Pump: Enabled; Detection Option: NA*
- *Scenarios 2 ==> Digital Filter: Disabled; Charge Pump: Disabled; Detection Option: NA*
- *Scenarios 3 ==> Digital Filter: Enabled; Charge Pump: Disabled; Detection Option: Option 1*
- *Scenarios 4 ==> Digital Filter: Enabled; Charge Pump: Disabled; Detection Option: Option 2*
- *Scenarios 5 ==> Digital Filter: Enabled; Charge Pump: Enabled; Detection Option: Option 1*
- *Scenarios 6 ==> Digital Filter: Enabled; Charge Pump: Enabled; Detection Option: Option 2*
- *Scenarios 7 ==> In this scenario, the Calibration is performed with the load on the antennae and the load is removed during Lpcd Loop from the Antennae.*

6.10 Simplified API EMVCo

This application will configure Reader Library as per EMVCo specification and start EMVCo polling. This loop back application will send SELECT_PPSE command and is used to test EMVCo.3.1a(L1) digital compliance. Simplified approach, after library initialization, is using only three commands:

- *phNfcLib_Activate()*
- *phNfcLib_Transmit()*
- *phNfcLib_Receive()*

6.11 Simplified API EMVCo Analog

This example contains three mode of operations within itself for the user to choose as below.

- *EMVCo LoopBack Application*
- *Trans send Type A application*
- *Trans send Type B application*

Above Application modes are used to perform EMVCo2.6(L1) Analog compliance validation.

6.12 Simplified API ISO

This example is a reference application to demonstrate the usage of Simplified API with ISO profile. Application contains example of Type A Layer 4, Type B Layer 4, MIFARE DESFire, MIFARE Ultralight, MIFARE Classic, ISO5693 and ISO18000p3m3.

Example demonstrates how to use simplified API, which require, after successful library initialization, only three commands:

- *phNfcLib_Activate()*
- *phNfcLib_Transmit()*
- *phNfcLib_Receive()*

7. References

[1] http://www.nxp.com/products/:CLRC66303HN

[2] CLRC663 datasheet

[3] AN11019 CLRC663, MFRC630, MFRC631, SLRC610 Antenna Design Guide

[4] AN11145 CLRC663, MFRC631, MFRC 630, SLRC610 Low Power Card Detection

[5] AN11021 CLRC663, MFRC631, MFRC630, SLRC610 Software Design Guide for NXP®RDLib

[6] AN11211 Quick Start Up Guide RC663 Blueboard

[7] http://www.nxp.com/pages/:NFC-READER-LIBRARY

[8] MCUXpresso Integrated Development Environment (IDE)

http://www.nxp.com/products/software-and-tools/ run-time-software/mcuxpresso-software-and-tools/mcuxpresso-integrated-development-environment-ide:MCUXpresso-IDE

8. Legal information

8.1 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.

8.2 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.

8.3 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.

8.4 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 DESFire — is a trademark of NXP B.V.

9. List of figures

10. Contents

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

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

Date of release: 15 May 2017
205914
Document identifier: AN11022