AN11505

CLRC663, MFRC631, MFRC630, SLRC610 PC-Serial RS232 Quick Start Guide

Rev. 1.0 — 6 August 2014
284210

Application note
COMPANY PUBLIC

Document information

Info	Content
Keywords	CLRC663, MFRC631, MFRC630 and SLRC610, quick start guide
Abstract	The document provides information on powering the evaluation board and executing scripts with the CLRC663, MFRC631, MFRC630 and SLRC610 test program.

Revision history

Rev	Date	Description
1.0	20140806	First official release

Contact information

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

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

1. Introduction

The “RC663 Serial Test Program” is provided as an executable application with a graphical user interface. In principal it is build on very low level – just offering read and write register operations. It can also be operated with MFRC631, MFRC630 and SLRC610

The software can be used to communicate with CLRC663, MFRC631, MFRC630 and SLRC610 on register level. By the use of script files, single register operations are executed to configure the IC for a certain operating mode. The script files have the extension ***.jcf***.

Note that the RC663 Serial Test Program is intended to show the chip functionality only on a very basic level.

In the following steps the PC serial is described by using the CLRC663 red board. It can also be used with the MFRC631, MFRC630 or SLRC610.

For MFRC631, MFRC630 and SLRC610 the protocol settings have to be set with load protocol, because the registers are locked.

The following script files are provided: (for RC663,)

- **RC663_ISO14443A_4byteUID.jcf** - returns the UID of an ISO14443-3A card. Only 4byte UIDs are retrieved for this script.
- **RC663_ISO14443B_ReqB.jcf** - returns the UID of an ISO14443-3B card
- **RC663_ISO15693_Inventory.jcf** - performs an ISO15693 Inventory command
- **RC663_Felica_ReqC.jcf** - act as passive PICC Target
- **LPCD_ModeConfig_Part1.jcf** - configures the board to LPCD
- **LPCD_DetectPICC_Part2.jcf** – runs LPCD mode with prior defined settings

1.1 Getting Started

Please connect the CLRC663 Reader to the RS232 interface and power it up using the enclosed switched power supply.

7.5 Volts must be used as input voltage.

The polarity of the connector plug of power supply is not relevant.

The Baudrate must be set to 115200.

1. Start the *RC663 Serial RS232 Test Progra*m by clicking on the executable file. It is a standalone program, no further installation is required. An empty window opens as outlined in Fig 1:

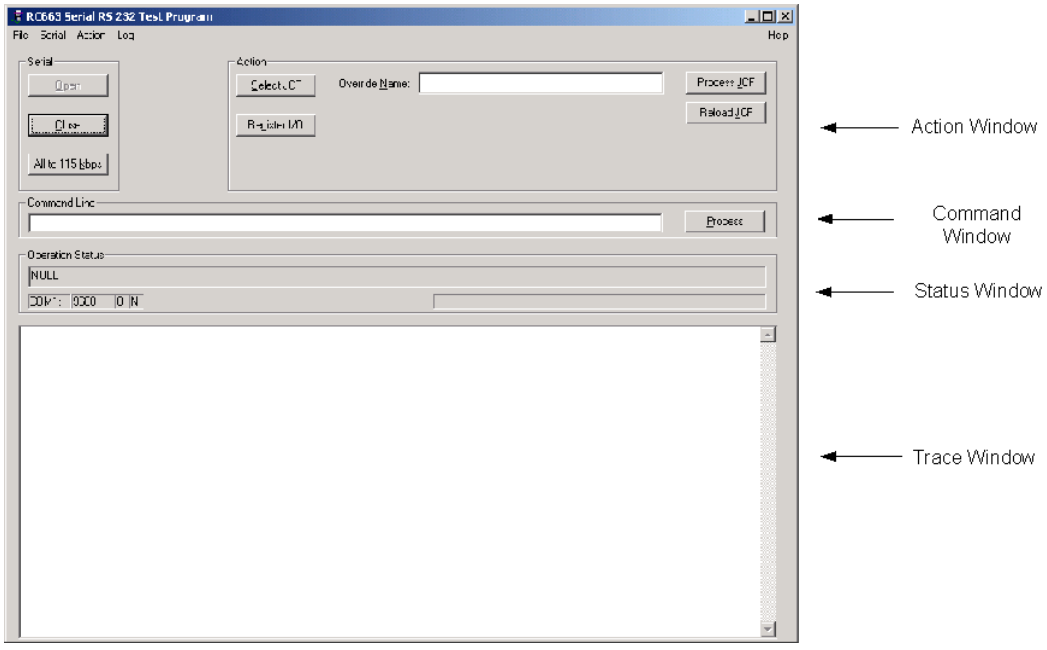

Fig 1. Main window of RC663 Serial

The *RC663 Serial RS232 Test Program* is a simple User Interface (UI) utility for register-based access to the RC663 hardware using the serial interface.

2. The RS232 COM port has to be configured: **Port/Settings…**

Choose the appropriate RS232 COM port for your serial communication, **Verify** (checks if the port is used or not) your selection and press **Apply**.

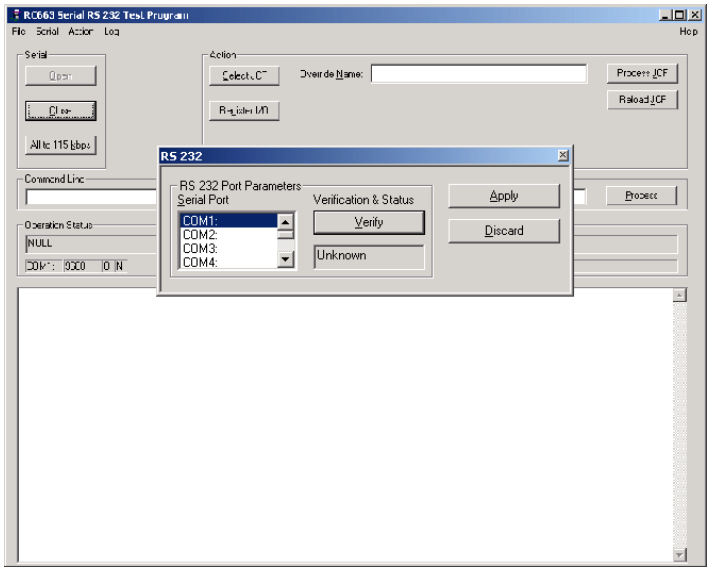

Fig 2. Com Port Selection

3. By using the **Open** button the communication channel should be established. An indicator of correct settings is the changing of all buttons but Open from inactive to active state.
4. By pressing **Select JCF** a window with all delivered scripts is opened. For this first try, the file "**RC663_ISO14443A_4byteUID**" is taken.

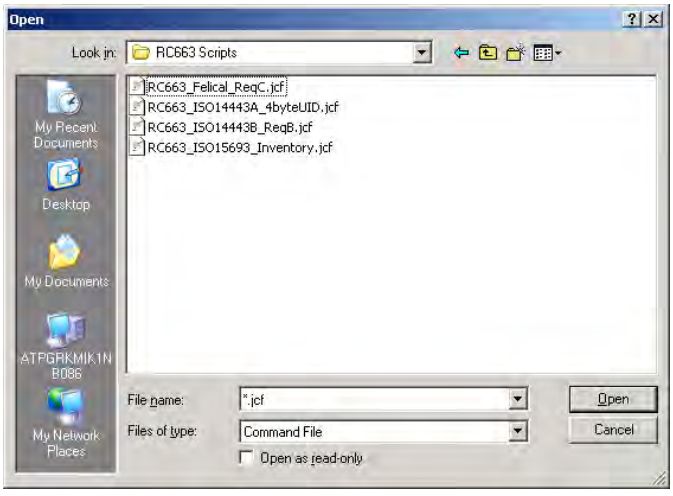

Fig 3. Select JCF file

It might be useful while investigating the different modes to open the files in a text editor, too, due to inline documentation which will not be displayed in the trace window.

5. Put a MIFARE Classic 1k on the Demo-Board and start the communication by using the **Process JCF** button.

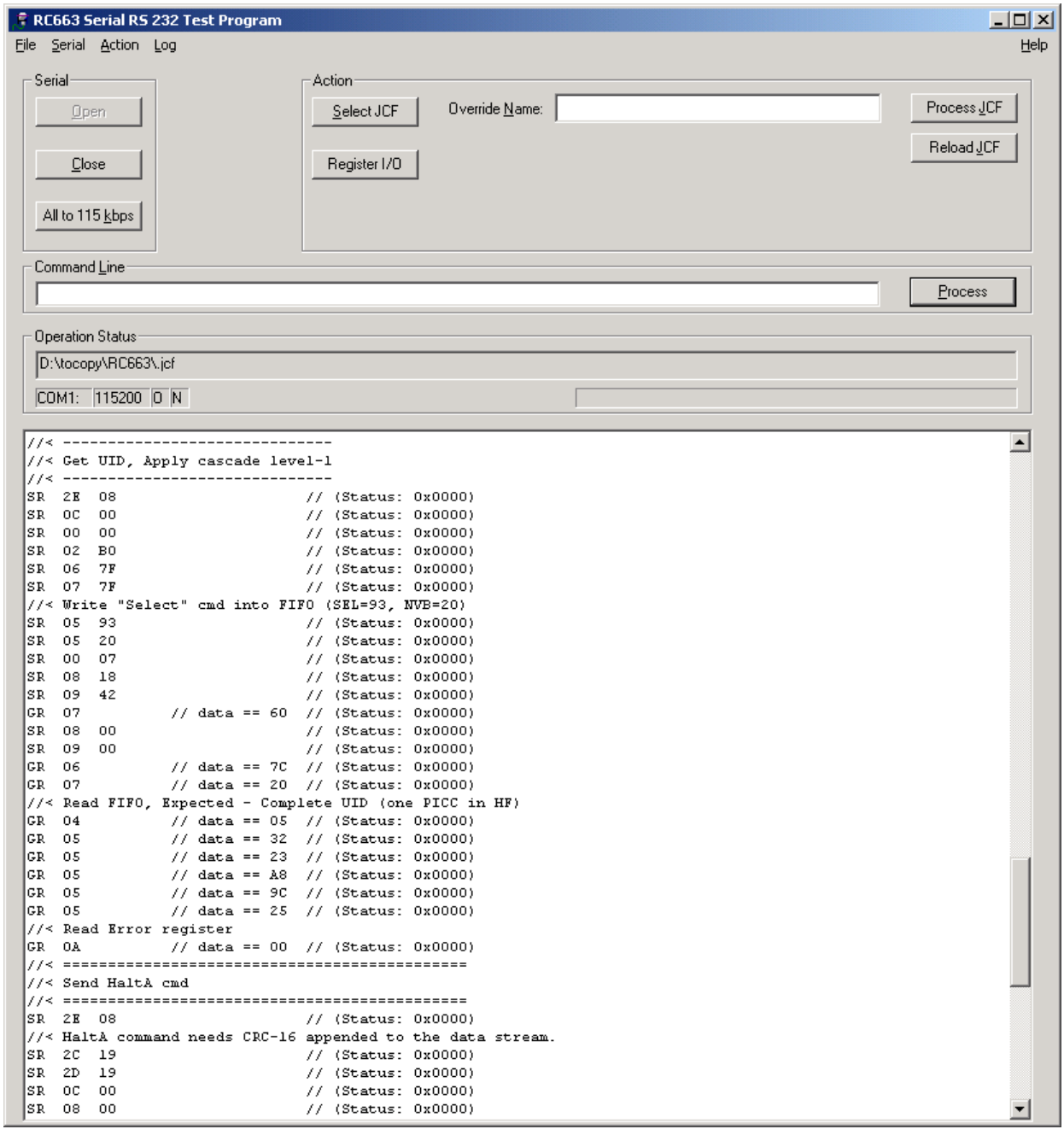

Fig 4. Reading 4byte UID

All the possible commands (e.g. SR, RE…) used in the scripts are explained when selecting **Help/Commands**.

As mentioned above more details on the scripts can be viewed by opening the *.jcf file with a text editor.

The first comments of the script chosen: RC663_ISO14443A_4byteUID.jcf to compare it to the output of the trace window:

```
1    CLL
2    CHB 115200
3
4    //> ============================================
5    //> RC663 Script for (Iso14443-3A protocol):
6    //>   * ReqA
7    //>   * Get UID (Select: Casade level 1)
8    //>   * HaltA
9    //>
10   //> Note: Only one PICC shall be in HF
11   //> ============================================
12
13   //> ============================================
14   //> RC663 ApplyProtocolSettings:  ISO14443A=01
15   //> ============================================
16   //
17   //> Configure Timers
18   //
19   // Set Timer-0, T0Control_Reg:
20   // Starts at the end of Tx. Stops after Rx of first data. Auto-reloaded. 13.56
     MHz input clock.
21      SR 0F 98
22
23   // Set Timer-1, T1Control_Reg:
24   // Starts at the end of Tx. Stops after Rx of first data. Input clock - cascaded
     with Timer-0.
25      SR 14 92
26
27   // Set Timer-2, T2Control_Reg:  Timer used for LFO trimming
28      SR 19 20
29
30   // Set Timer-2 reload value (T2ReloadHi_Reg and T2ReloadLo_Reg)
31      SR 1A 03
32      SR 1B FF
33
34   // Set Timer-3, T3Control_Reg:
35   // Not started automatically. Not reloaded. Input clock 13.56 MHz
36      SR 1E 00
37
38   //> Configure FIFO Size=255 and Water-level
```

```
39  // Set FifoControl_Reg, Fifo size=255 bytes. Flush FIFO
40     SR 02 90
41
42  // Set WaterLevel =(FIFO length -1)
43     SR 03 FE
44
45  // RxBitCtrl_Reg(0x0c)     Received bit after collision are replaced with 1.
46     SR 0C 80
47
48  // DrvMod reg(0x28), Tx2Inv=1
49     GR 28
50     SR 28 80
51
52  // TxAmp_Reg(0x29)
53     SR 29 00
54
55  // DrvCon_Reg(0x2A)
56     SR 2A 01
57
58  // TxI_Reg(0x05),(0x05)
59     SR 2B 05
60
61  // RxSOFD_Reg(0x34),(0x00),
62        SR 34 00
63
64  // Rcv_Reg(0x38),(0x12)
65        SR 38 12
66  //
67  //> ============================================
68  //> 2. LoadProtocol( bTxProtocol=0, bRxProtocol=0)
69  //> ============================================
70
71  //> Terminate any running command. Flush_FiFo
72     SR 00 00
73     SR 02 b0
74
75  // Clear all IRQ 0,1 flags
76     SR 06 7f
77     SR 07 7f
78
79  //> Write in Fifo: Tx and Rx protocol numbers(0,0)
80     GR 04
81     SR 05 00    // Rx protocol=0
82     SR 05 00    // Tx prot=0
83
84  // Enable IRQ0 interrupt sources
85  //
86  // Idle interrupt(Command terminated), RC663_BIT_IDLEIRQ=0x10
87     GR 08
88     SR 08 10
89
```

```
90    // Enable Global IRQ propagation.
91       GR 09
92       SR 09 40
93
94    //> Start RC663 command "Load Protocol"=0x0d
95       SR 00 0D
96    …
97    …
98    …
```

1.2 Command Line

Besides the script files, single commands can be handled, too. This can be done by typing e.g. **GR 0A** (Gets the status of register 0A (error register)) in the Command Line and execute it with **Process**. In principle all lines of the scripts provided could be fed into the command line sequentially, each followed with a **Process**.

1.3 PC serial command set

Command: SR

Synopsis: SR <address> <data>

Description:

The SR function sets a Joiner register, located at address <address> according to data <data>, both specified as an 8-bit HEX value.

Command: GR

Synopsis: GR <address>

Description:

The GR function gets data from a Joiner register, located at address <address> (8-bit HEX). The retrieved value is stored in IOR.

Command: MR

Synopsis: MR <address> <mask> <set>

Description:

The MR function modifies a Joiner register, located at address <address>. The mask <mask> specifies which bits to modify by having the corresponding bits set. If the <set> parameter is nonzero, the corresponding bits are set, otherwise cleared. All values are in 8-bit HEX format.

Command: RE

Synopsis: RE <address> <data>

Description:

The RE function compares a Joiner register, located at address <address> to data, specified in the <data> parameter. If equal, IOR is 0, otherwise 1. All values are in 8-bit HEX format.

Command: RF

Synopsis: RF <address> <data> <mask>

Description:

The RF function compares a Joiner register, located at address <address> to data, specified in the <data> parameter, AND'ed with the content of <mask>. If equal, IOR is 0, otherwise 1. All values are in 8-bit HEX format.

APPLICATION COMMANDS:

Command: CHB

Synopsis: CHB <bitrate>

Description:

The CHB function sets the <bitrate> (in bps) of the PC serial port. Possible values are: {9600, 19200, 38400, 57600, 115200}.

Command: WIE

Synopsis: WIE <timeout_ms>

Description:

The WIE function waits for an edge at the serial port's RI pin. Maximum waiting time is specified by <timeout_ms>, in [ms]. This function should be used with caution only (not recommended).

Command: WIL

Synopsis: WIL <level> <timeout_ms>

Description:

The WIL function waits for the serial port's RI pin to reach a certain logical level, specified by <level>, (= {0, 1}) Maximum waiting time is specified by <timeout_ms>, in [ms]. This is the preferred intr. function.

Command: SLP

Synopsis: SLP <timeout_ms>

Description:

The SLP function waits for the time is specified by <timeout_ms>, in [ms] to expire.

Command: CLL

Synopsis: CLL

Description:

The CLL function removes all content from the application's LOG window.

Command: //

Synopsis: // <Comment Text>

Description:

The // function does nothing but allow comments being added to a script.

The text <Comment Text> must be separated from the command by at least one blank.

Command: //>

Synopsis: //> <Message Text>

Description:

The //> function allows messages to be displayed during script execution.

The text <Message Text> must be separated from the command by at least one blank.

Command: //#

Synopsis: //# <data>

Description:

The //# function allows data to be displayed during script execution.

The <data> parameter can be either plain data (8-bit HEX) or a User Register.

Command: JMP

Synopsis: JMP <destination>

Description:

The JMP function skips script commands until a label with the name <destination> is found. The label name <destination> should contain only {a..z, A..Z, _}.

Command: JNE

Synopsis: JNE <value> <compare_value> <destination>

Description:

The JNE function compares User Register or plain data <value> to <compare_value>. If unequal, the function skips script commands until a label with the name <destination> is found. Data are in 8-bit HEX format.

Command: JNM

Synopsis: JNM <value> <compare_value> <mask> <destination>

Description:

The JNM function compares User Register or plain data <value> which is AND'ed with the content of <mask>, to <compare_value>. If unequal, the function skips script commands until a label with the name <destination> is found. Data are in 8-bit HEX format.

Command: :::

Synopsis: ::: <destination>

Description:

The ::: function is the <destination> of the JUMP commands.

The label name <destination> should contain only {a..z, A..Z, _}.

Command: MOV

Synopsis: MOV <destination> <source>

Description:

The MOV function copies User Register or plain data from <source> to <destination>. Data are in 8-bit HEX format.

Command: INC

Synopsis: INC <user_register>

Description:

The INC function increments a user register.

Command: DEC

Synopsis: DEC <user_register>

Description:

The DEC function decrements a user register.

Command: BRK

Synopsis: BRK

Description:

The BRK function stopps the execution of the current script.

Command: SAV

Synopsis: SAV [<File Name>]

Description:

The SAV function stores the log output to the current working directory.
The File Name is used if present (max. length of 32 char). If no parameter is present a file dialog is opened to specifiy the location and the name.

U S E R R E G I S T E R S :

User registers are divided into special purpose registers IOR and IOE and general purpose registers MLn, where n = 0..7. The special purpose register IOR contains the result (data, if any) of an I/O operation.
IOE is the error register, zero means success, one failure.

2. Legal information

2.1 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.

2.2 Disclaimers

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

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

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

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

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

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

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

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

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

2.3 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.

2.4 Trademarks

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

3. Contents

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

Date of release: 6 August 2014
284210
Document identifier: AN11505