

UM10989

NTAG I²C *plus* Demo Android app Developer start-up guide

Rev. 2.0 — 15 March 2017
367620

User manual
COMPANY PUBLIC

Document information

Info	Content
Keywords	NTAG I ² C, NTAG I ² C <i>plus</i> , Explorer Kit, Android, NFC tag
Abstract	This User Manual aims at describing the functionalities and how to start using NTAG I ² C <i>plus</i> Android application source code.



Revision history

Rev	Date	Description
2.0	20170315	Completely reworked version Removed redirect links
1.0	20160316	First version

Contact information

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

1. Object

NTAG I²C Explorer kit is an all-in-one demonstration and development resource to demonstrate the unique properties of the NTAG I²C tag chip. By including a full complement of hardware and software tools, users can not only investigate the capabilities of the chip through the various demonstrations, but also develop and test their own applications.

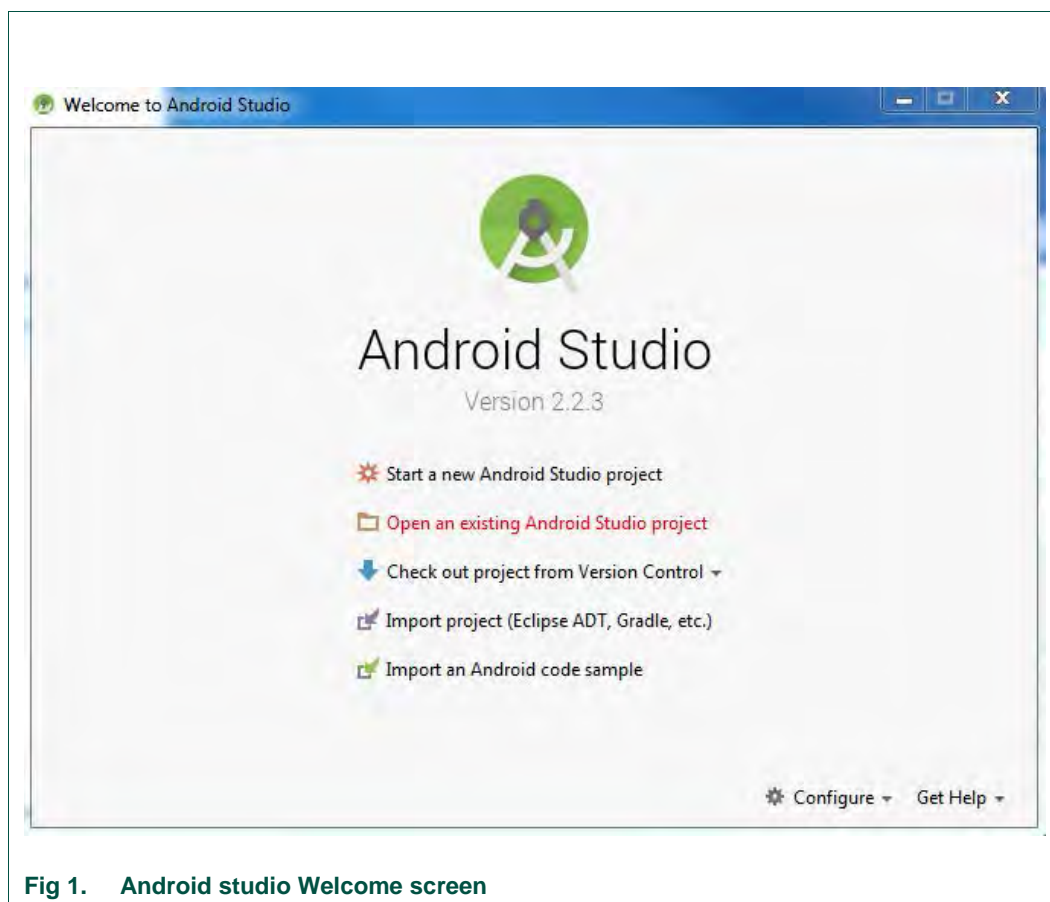
This User Manual explains how to use the source code of NTAG I²C demo application for Android in Android Studio. Technical aspects related to the IC functioning (i.e. the configuration registers) are beyond the scope of this document. Source code is re-usable for NXP NTAG I²C (1K and 2K of user memory) and NTAG I²C *plus* (1K and 2K user memory) products. For detailed functionalities please take a look to dedicated datasheets [NTAGI2Cplus] and [NTAGI2C].

For use of this user manual, basic Android programming skills should be already obtained. Please follow <https://developer.android.com/index.html> webpage to get started.

2. Android studio import

Download and install latest instance of Android Studio (2.2.3 at the time of writing this document) Make sure you have the latest Android SDK installed. Since this app is only working with NFC enabled devices, AVD (Android Virtual Device) is not necessary for debugging.

Download zipped source files from NXP website [link]. Extract the .zip to your local hard drive. Open Android studio. On the welcome screen select "Open an existing Android Studio project".



Select the previously extracted source directory. Importing will start.

If everything is updated, there is no need for internet access. Optionally, “Gradle” synchronization will start – internet connection is needed for first debug session.

Project is loaded into Project pane.

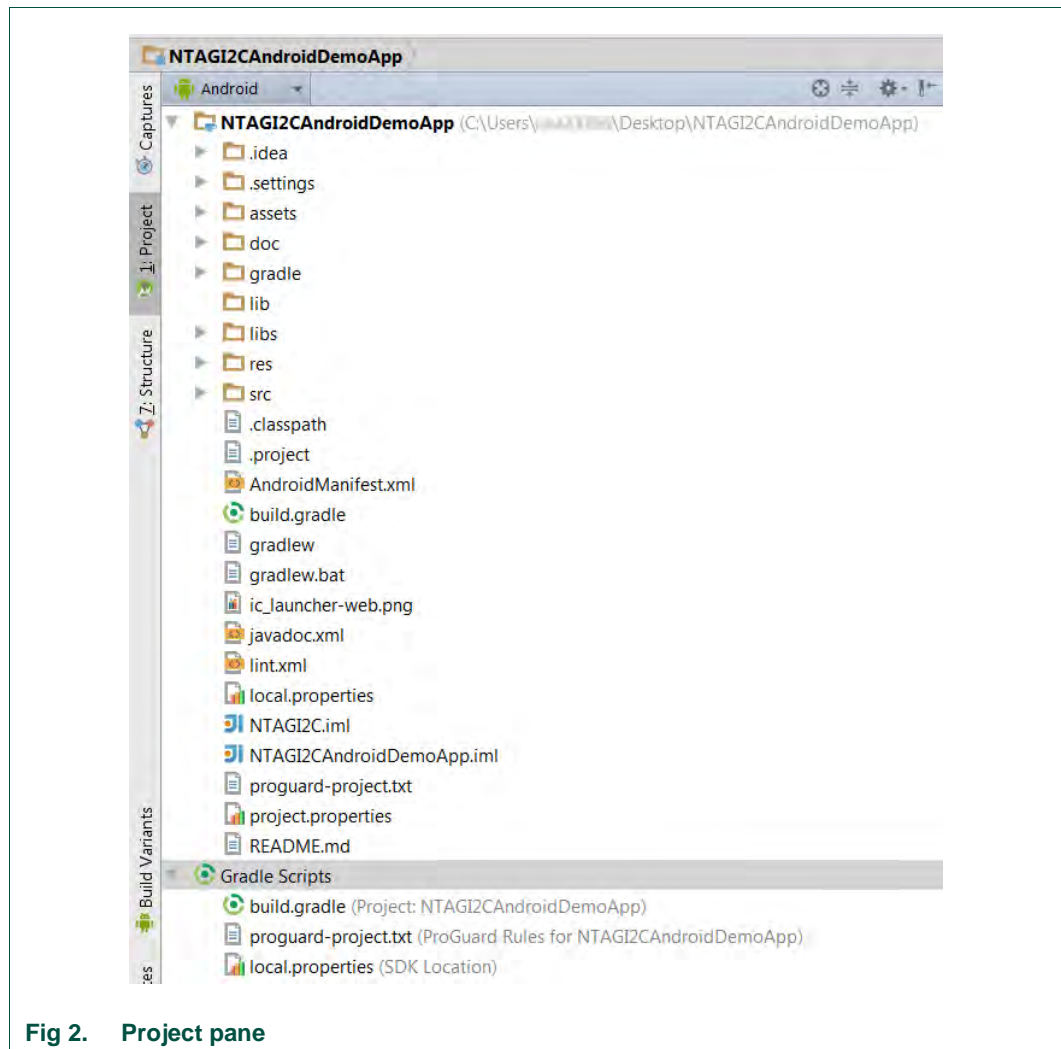


Fig 2. Project pane

2.1 Source code design

Application is using standard Android NFC API. It provides access to Near Field Communication (NFC) functionality. Classes and methods are implemented according NTAG I²C *plus* datasheet. External “ndeftools” library is used for easy manipulation of NDEFs.

3. Android Phone and Android studio set-up for debugging

3.1 Enable Debug mode on the mobile phone

Debug mode should be “on” – under “Settings” go to “Developer options” Set “USB debugging to ON”. This will enable debugging on your phone. Connect phone via USB to your PC. Latest drivers for the phone being debugged should be installed (e.g. Nexus phones should have Google USB driver installed).

On some phones where “Debug mode” is not visible in “Settings” mode, find the “Build Number” under “Settings” → “About device”. Tap on the section 7 times. After few taps, a small pop up notification should appear saying “you are now X steps away from being a developer” with a number that counts down with every additional tap.

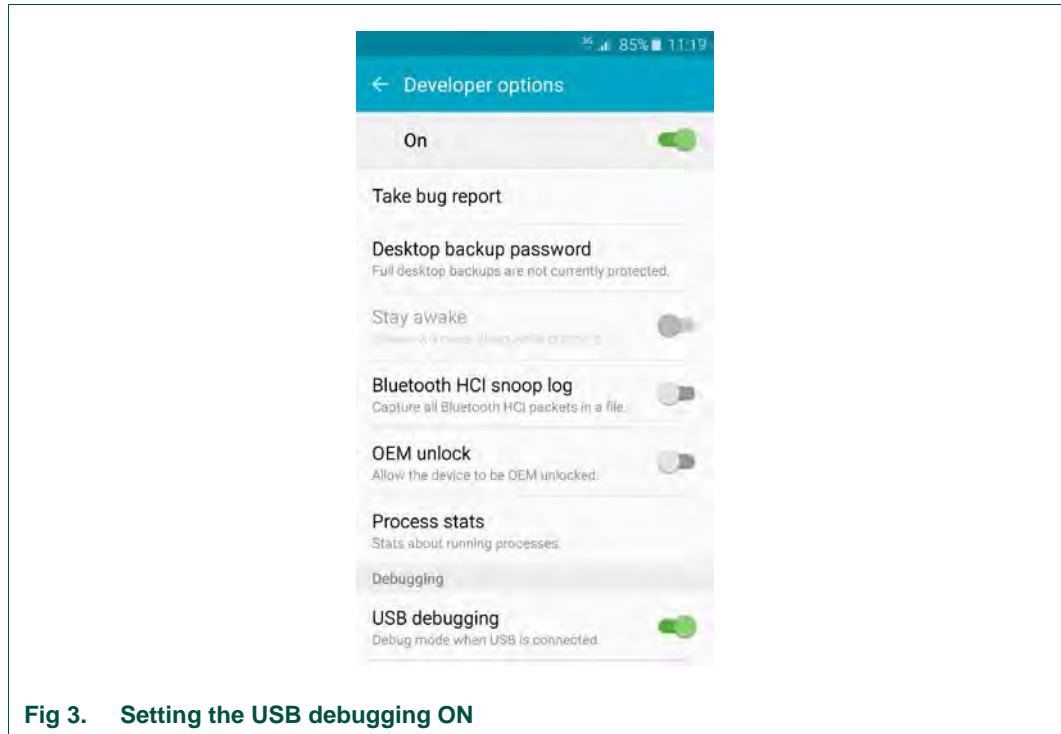


Fig 3. Setting the USB debugging ON

3.2 Select and connect your phone as debugging target

1. In Android studio go to “Run”
2. Edit Configurations...
3. Select your Android Application under “Module:”
4. Under “Deployment Target Options” select “USB Device”
5. Press “Apply” and “OK”

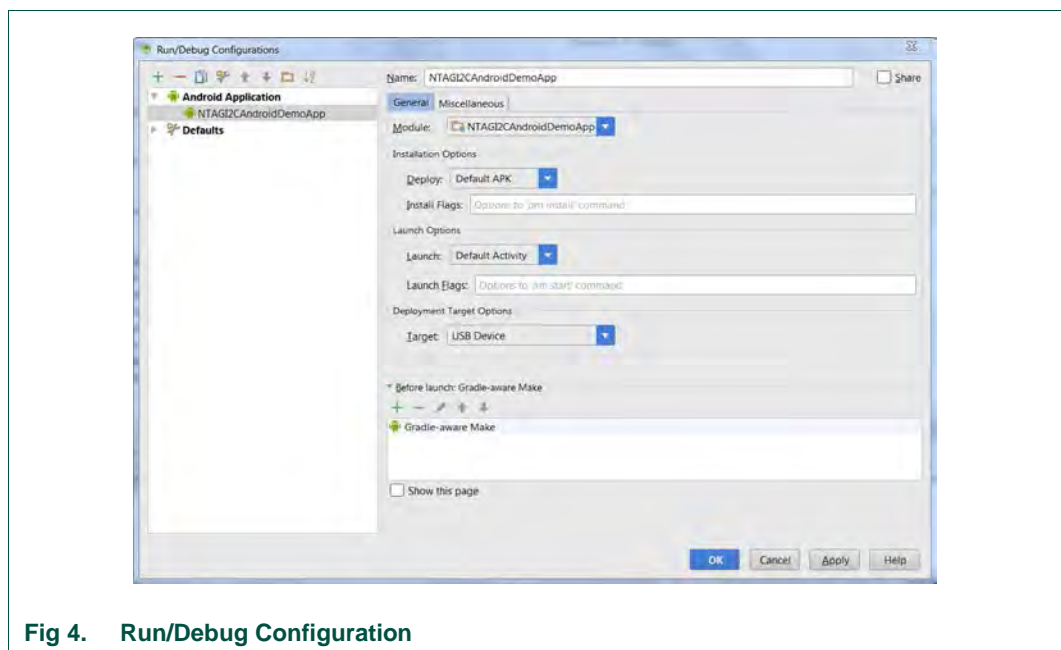


Fig 4. Run/Debug Configuration

3.3 Flash project target

1. In Android studio go to “Run”
2. Select “Run [project name]“
3. App will be flashed to your USB plugged phone
4. Debugging is started

If it doesn’t work, open the Settings in the icon List and install all plugins except the google plugins, because they are not necessary.

4. Special Methods for NTAG I²C *plus*

NTAG I²C *plus* has some special features like FAST_WRITE, Authentication, pass-through mode etc. Below are interesting parts of source code to show these capabilities. In this section brief explanation of specific methods are given. For easy searching through sources in Android Studio, “Search Everywhere” is implemented. With 2 tabs on the Shift Key, the dialog is popped-up. This way it is possible to quickly access the portion of the source code which is already imported.

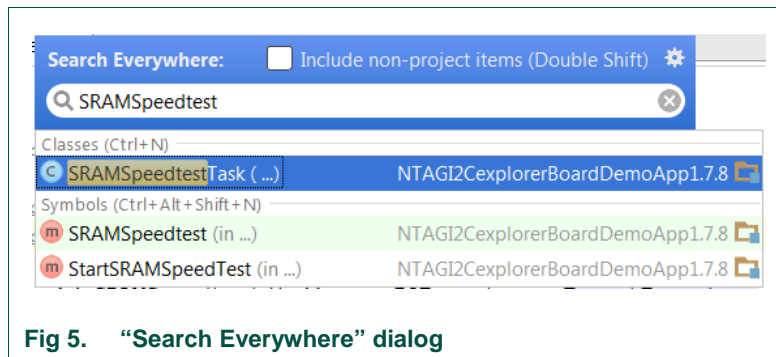


Fig 5. “Search Everywhere” dialog

In further instructions, search string will be provided to search over source code:

Search Everywhere (2 times Shift Key): Sample_Method

4.1 SRAM speed test

SRAM Speed Test writes 64 bytes (SRAM Size) as many times as requested by the user and then those bytes are read. The CRC32 is used to ensure the integrity of the data transmitted / received. In the method “doInBackground” the tests are done. The last byte to read/write is marked by writing “finish_S_[byte]”. **Search Everywhere (2 times Shift Key): SRAMSpeedtestTask**

4.2 AUTHENTICATION

Access to NTAG I²C *plus* can be password protected. “ObtainAuthStatus”, together with “getProtectionPlus” method defined in “NTAG_I2C_Commands.java” and “Minimal_NTAG_Commands.java” return the protection status every time an IC is detected. Authentication request window is shown with three pre-defined passwords in case that NTAG I²C *plus* is PWD protected.

Search Everywhere (2 times Shift Key): ObtainAuthStatus

Method “Auth” returns password protection status of the tag. It is used when the user taps on the lock icon. Depending on the status of the tag the user can choose:

- If the status is unprotected, user wants to protect the tag.
- If the tag is already authenticated, then we want to unprotect – remove authentication.
- If the tag is in protected status, then we want to authenticate to be able to complete the demos on password protected tags.

Search Everywhere (2 times Shift Key): Boolean Auth(

4.3 NFC MCU Firmware flash

Explorer kit in addition of NTAG I²C *plus* antenna board with the on-chip SRAM, permits temporary storage of data during the transfer, enabling the tag chip to act as a modem. Programming is performed by erase and write operation on the on-chip flash memory, as directed by the end-user application code. This is called an “In application programming” (IAP) which is supported by NXPs’ LPC11U24 MCU and special secondary bootloader. For details please see [UM10966].

The way how flash memory on MCU is organized, whole sectors of NTAG are flashed and the minimum size of data to write is pages. When flashing is done, MCU is informed about status of flashing process.

Search Everywhere (2 times Shift Key): Boolean Flash(

4.4 NDEF

Standard NDEF operations according to NDEF specifications are implemented.

4.4.1 NDEF Text Message method

A New method can be declared by “createNdefMessage” type method with desired name to create a new NDEF text message. It is Android class.

Example:

```
NdefMessage msg = createNdefTextMessage(“your text”).
```

The NdefTextMessage Object is returned.

More resources on the link below:

<https://developer.android.com/reference/android/nfc/NdefMessage.html>

Search Everywhere (2 times Shift Key): createNdefTextMessage(

4.4.2 NDEF URI Message method

A New method can be declared by “createNdefUriMessage” type method with desired name to create a new NDEF URI message. It is Android class.

Example:

```
NdefMessage msg = createNdefUriMessage("your url").
```

The NdefUriMessage Object is returned.

More resources on the link below:

<https://developer.android.com/reference/android/nfc/NdefMessage.html>

Search Everywhere (2 times Shift Key): createNdefUriMessage (

4.4.3 NDEF Bluetooth secure Simple Pairing Message method

A New method can be declared by “createNdefBSSPMessage” type method with desired name to create a bluetooth secure Simple Pairing Message. It is Android class.

Example:

```
NdefMessage msg = createNdefBSSPMessage("").
```

The NdefBSSPMessage Object is returned.

More resources on the link below:

<https://developer.android.com/reference/android/nfc/NdefMessage.html>

Search Everywhere (2 times Shift Key): createNdefBSSPMessage (

4.4.4 NDEF Smart Poster Message method

A New method can be declared by “createNdefSpMessage” type method with a desired title and URI to create a NDEF SmartPoster Message. It is Android class.

Example:

```
NdefMessage msg = createNdefSpMessage ("").
```

The NdefSpMessage Object is returned.

More resources on the link below:

<https://developer.android.com/reference/android/nfc/NdefMessage.html>

Search Everywhere (2 times Shift Key): createNdefSpMessage(

5. References

- [NTAGI2Cplus] NT3H2111/NT3H2211, NTAG I²C *plus*, NFC Forum Type 2 Tag compliant IC with I²C interface datasheet
http://www.nxp.com/documents/data_sheet/NT3H2111_2211.pdf
- [NTAGI2C] NT3H1101/NT3H1201, NTAG I²C, NFC Forum Type 2 Tag compliant IC with I²C interface datasheet
http://www.nxp.com/documents/data_sheet/NT3H1101_1201.pdf
- [NFCT2T] NFC Forum – Type 2 Tag Operation Ver. 1.2 Technical Specification
http://members.nfc-forum.org/specs/spec_list/
- [UM10966] NTAG I²C *plus* Explorer Kit and Android Demo user manual
http://nxp.com/documents/user_manual/UM10966.pdf

6. Legal information

6.1 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.

6.2 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 of 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.

6.3 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.

6.4 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.

PC-bus — logo is a trademark of NXP B.V.

NTAG — is a trademark of NXP B.V.

7. List of figures

Fig 1. Android studio Welcome screen4
Fig 2. Project pane.....5
Fig 3. Setting the USB debugging ON6
Fig 4. Run/Debug Configuration.....6
Fig 5. "Search Everywhere" dialog.....7

8. Contents

1.	Object	3
2.	Android studio import.....	3
2.1	Source code design	5
3.	Android Phone and Android studio set-up for debugging.....	5
3.1	Enable Debug mode on the mobile phone	5
3.2	Select and connect your phone as debugging target.....	6
3.3	Flash project target	7
4.	Special Methods for NTAG I²C plus	7
4.1	SRAM speed test	7
4.2	AUTHENTICATION.....	7
4.3	NFC MCU Firmware flash	8
4.4	NDEF	8
4.4.1	NDEF Text Message method	8
4.4.2	NDEF URI Message method.....	9
4.4.3	NDEF Bluetooth secure Simple Pairing Message method	9
4.4.4	NDEF Smart Poster Message method	9
5.	References	10
6.	Legal information	11
6.1	Definitions	11
6.2	Disclaimers.....	11
6.3	Licenses.....	11
6.4	Trademarks.....	11
7.	List of figures.....	12
8.	Contents.....	13

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