UM10966

NTAG I²C *plus* Explorer Kit - Android Demo

Rev. 2.1 — 09 March 2017 **User manual**
360121 **COMPANY PUBLIC**

Document information

Info	Content
Keywords	NTAG I²C *plus*, Explorer Kit, Android, NFC tag
Abstract	This User Manual describes the functionalities and how to use the Android and PC application of the NTAG I²C *plus* Explorer kit. Both applications provide the same functionality and have the same look and feel so this User Manual is valid for both.

Revision history

Rev	Date	Description
2.1	20170309	Update - Bluetooth Pairing added
2.0	20170201	Updates
1.0	20160216	Initial Version

Contact information

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

1. Object

NTAG I²C *plus* Explorer kit is an all-in-one demonstration and development resource to demonstrate the unique properties of the NTAG I²C *plus* connected tag. By including a full complement of hardware and software tools, users can investigate the capabilities of the chip through the various demonstrations, develop and test their own applications (with additional LPC-Link2 debug probe[1]).

This User Manual explains how to use the NTAG I²C *plus* demo application for Android. The Windows app is not detailed in this User Manual since it provides identical functionalities as the Android app, therefore the same User Manual is valid.

Technical aspects related to the IC features are beyond the scope of this document. To get further technical details please consult the dedicated Datasheet “NTAG I²C *plus*, NFC Forum Type 2 Tag compliant IC with I²C interface” (refer to [NTAGI2Cplus]).

[1] www.nxp.com/LPC-LINK2

2. NTAG I²C *plus* introduction

The NTAG I²C *plus* part of NXP's NTAG family offering both contactless and contact interfaces. In addition to the passive NFC Forum compliant RF interface, the NTAG I²C *plus* provides an I²C interface that allows the IC to communicate with the microcontroller when the chip is powered by an external device, i.e. a NFC mobile phone.

The NTAG I²C *plus* operating in energy harvesting mode provides the possibility to supply external low power devices (e.g. microcontrollers) with the energy generated from the RF field of the external NFC device.

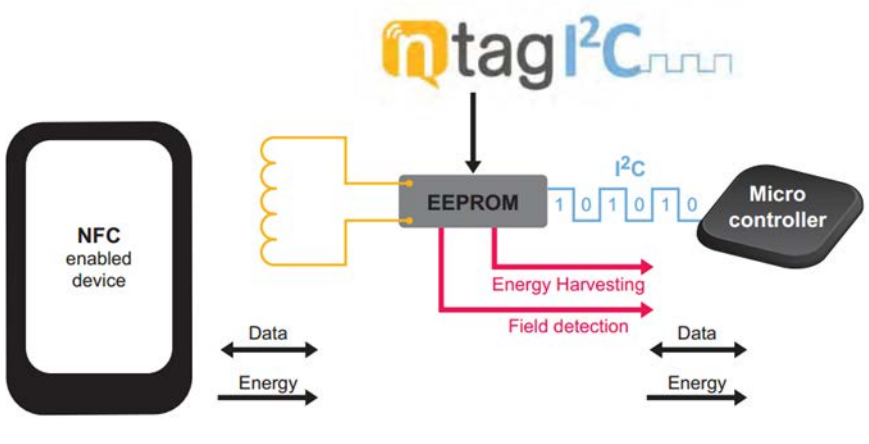

Fig 1. NTAG I²C *plus*

The NTAG I²C *plus* product has two types of memories:

1. EEPROM memory compliant with the NFC Forum Type 2 Tag implementation.
2. 64-byte SRAM memory, which is mapped to the EEPROM memory and it is externally powered.

The NTAG I²C *plus* features a pass-through mode that allows fast download and upload of data from the RF interface to the I²C interface and vice versa. This functionality makes use of the SRAM memory that allows fast data transfer between interfaces without the EEPROM performance limitations.

In addition to the I²C interface functionality, the NTAG I²C *plus* product features an Event Detection pin for waking up the connected host devices or synchronizing the data transfer between the two interfaces.

The NTAG I²C *plus* offers the possibility to protect the memory access. This protection is done by authenticating the tag with a password. When the tag is protected, authentication is needed to access the memory. The NTAG I²C *plus* also improves the speed when writing into the SRAM memory.

3. NTAG I²C *plus* Explorer kit contents

The NTAG I²C *plus* Explorer kit (NEK) consists of hardware and software tools that developers can use to understand the NXP NTAG I²C *plus* functionalities and create first prototypes to demonstrate its potential for other application. The kit includes:

3.1 Hardware components

3.1.1 NTAG I²C *plus* Explorer Board

A hardware board based on the NXP LPC 11U24 microcontroller (refer to [LPC11U24]), with on-board LCD display, NXP LM75B temperature sensor (refer to [LM75B]), voltage monitors, I²C serial bus connector, JTAG/SWD debug connector, RGD LED micro USB connector and five push buttons.

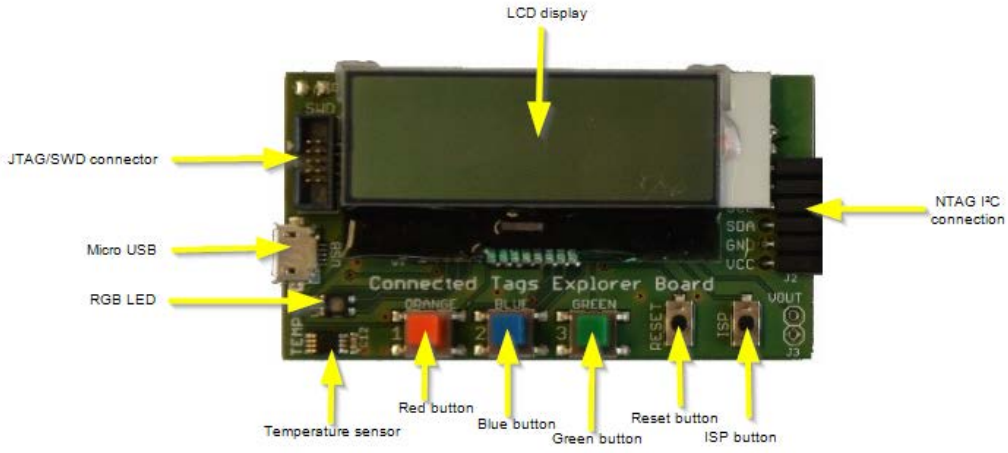

Fig 2. NTAG I²C *plus* Explorer Board (refer to [EXPLORER])

3.1.2 Antenna board

The antenna board carries the NTAG I²C *plus* 2k version itself and provides two interfaces:

- The RF interface to an NFC device.
- The I²C interface to the host, e.g. the NTAG I²C *plus* Explorer board.

The design files for both the PCB and Flex antenna can be found on the web page (refer to [PCB Antenna] and [Flex Antenna]

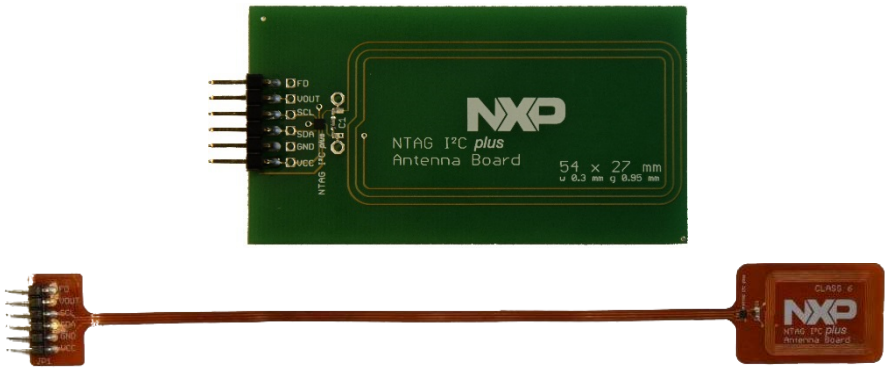

Fig 3. PCB and Flex antenna boards with NTAG I²C *plus* IC

3.1.3 Field detector board

The position of the antenna of NFC devices vary from device to device. To use the Explorer Kit with an NFC Device, NFC radio needs to be enabled. To find the position of the antenna, it is convenient to use the included field detector board. The LED helps to find the antenna position.

Design files may be downloaded from the demo board homepage (refer to [Field Detector])

Fig 4. Field detector board

3.1.4 USB reader

Instead of a NFC device, the USB (PCSC) reader (Identiv uTrust CLOUD 3700F) may be used in combination with the Windows app to develop applications.

Fig 5. Identiv uTrust CLOUD 3700F reader

To install the reader, download latest driver from the web page (refer to [Identiv]), extract "Identiv uTrust Installer.zip" and install it with no reader connected.

In Windows 7 operative systems, when a smartcard is placed over the reader there is frequently an issue regarding the smartcard mini-drivers. Although a solution to this problem is given in this section, more information can be found on the Windows Support Webpage[2]. To solve this issue the Smart Card Plug has to be disabled and Play in local Group Policy has to be changed to primary group policy settings for smart cards. The procedure is as follows:

1. Click **Start**, type *gpedit.msc* in the **Search programs and files** box, and then press ENTER.
2. In the console tree under **Computer Configuration**, click **Administrative Templates**.
3. In the details pane, double-click **Windows Components**, and then double-click **Smart Card**.
4. Right-click **Turn on Smart Card Plug and Play service**, and then click **Edit**.
5. Click **Disabled**, and then click **OK**.
6. Click **Start**, type regedit.exe in the **Search programs and files** box, and then press ENTER.
7. Go through the tree key, on the left, up to the key
   - **HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Cryptography \ Calais** for 32-bit system or
   - **HKEY_LOCAL_MACHINE \ SOFTWARE \ Wow6432Node \ Microsoft \ Cryptography \ Calais** for 64-bit system
8. Add a new DWORD value named CardDisconnectPowerDownDelay and set its value to 0.
9. Click **Start**, type services.msc in the **Search programs and files** box, and then press ENTER.
10. Find the smart card service in the list, right-click and click Restart.
11. Now you may plug the reader

[2] https://support.microsoft.com/en-us/kb/976832.

3.1.4.1 USB Reader firmware update

In some cases, reader is not functioning properly with Windows “NTAG I²C plus Demo” application. This is due to newer Reader´s firmware. In this case, it needs to be updated, using patch tool[3].

3.2 Software components

3.2.1 NTAG I²C *plus* Explorer board firmware

The firmware runs on the NTAG I²C *plus* Explorer board and is flashed during production at the MCU which supports the demonstration functionality of the hardware. The delivered NTAG I²C *plus* Explorer board firmware consists of three applications:

- **NTAG_I²C_Explorer_Bootloader**: This project implements the secondary bootloader application. It is flashed at on-chip memory address starting at 0x0000 0000 and it is the first application to be executed after the MCU boots. This application has three functions:
  - Jump to the start memory of the user application.
  - Enter into flashing mode functionality.
  - Enter into USB mode (Peek and Poke).
- **NTAG_I²C_Explorer_Demo**: This project implements the logic supporting the Android / Windows demonstration applications. It is flashed at on-chip flash memory starting at 0x0000 4000 address and it is executed after the bootloader jumps to the application start address.
- **NTAG_I²C_Explorer_Blink**: This is a sample project that sets into blinking mode the NTAG_I²C Explorer board as soon as the RF field is detected. This application is provided to illustrate the NFC flashing functionality and its binary image is provided embedded by default into the Android app (see Section 4.5).

3.2.2 Android app

The demo application on an Android NFC phone (“NFC mobile”) showcasing the various features of the NTAG I²C *plus*. The NTAG I²C Demo application is available for download from the public NXP website as well as at Google Play.

[3] http://www.nxp.com/demoboard/OM5569-NT322ER/documents/software/SW4044.zip

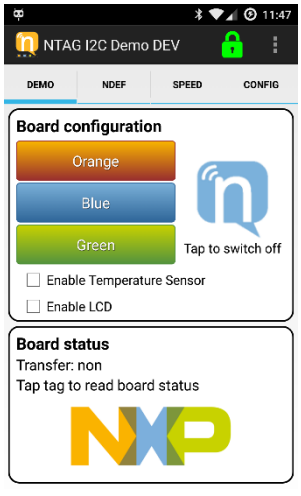

Fig 6. Android app GUI

Note: The UM10989 (refer to [UM10989]) describes in detail how to get started with the development of Android Applications.

3.2.3 Windows app

Together with the USB reader, the Windows app can be utilized to substitute a missing NFC mobile phone. The Windows app has similar functionalities as the Android app. This software component is available as a download from the public NXP website. The software will include a setup file which will install the Windows App in the folder 'Program files/NXP Semiconductors', this installation process will create a shortcut to the Windows App on your desktop. No further procedure is required to run this application.

1. Download the NTAG I²C Demo App (refer to [PC App])
2. If not done, install the Identiv uTrust driver as described above and the NTAG I²C Demo App
3. Launch the NTAG I²C Demo App

There are some points to consider when using the Windows App, since it is connected to an external reader via USB. There are sometimes issues with the connection between the reader and the tag. On these cases, it is recommended to restart the app and re-plug the reader.

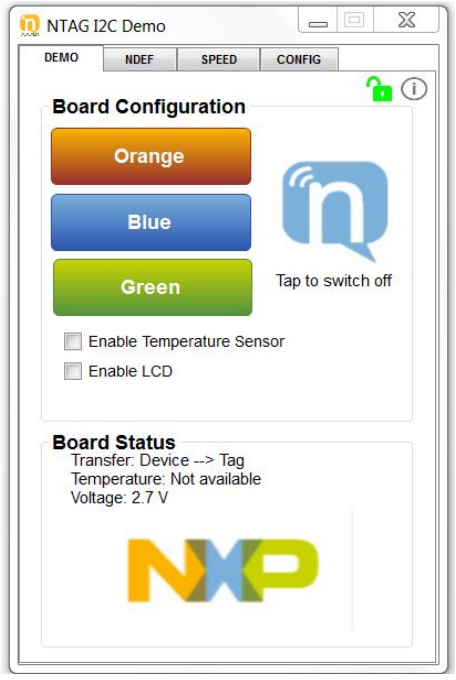

Fig 7. Windows application GUI

Note: The Windows application functionalities, GUI and look and feel are intentionally made the same as the Android application. The Windows application is intended to run in a Windows environment together with the Identiv uTrust CLOUD 3700F reader as a substitute in case an NFC phone is not available. Therefore, this User Manual is valid for both the Android app and Windows application. However, only Android app screenshots are shown in this document.

3.2.4 Peek and Poke GUI

The Peek and Poke GUI is a Windows app that can be used to examine the detailed memory contents of the NTAG I²C *plus* EEPROM via I²C interface. From hardware point of view only a USB cable connection from the board to the PC is needed (no need of the USB NFC reader). This software component is available as a download from the public NXP website (refer to [Peek&Poke]).

.

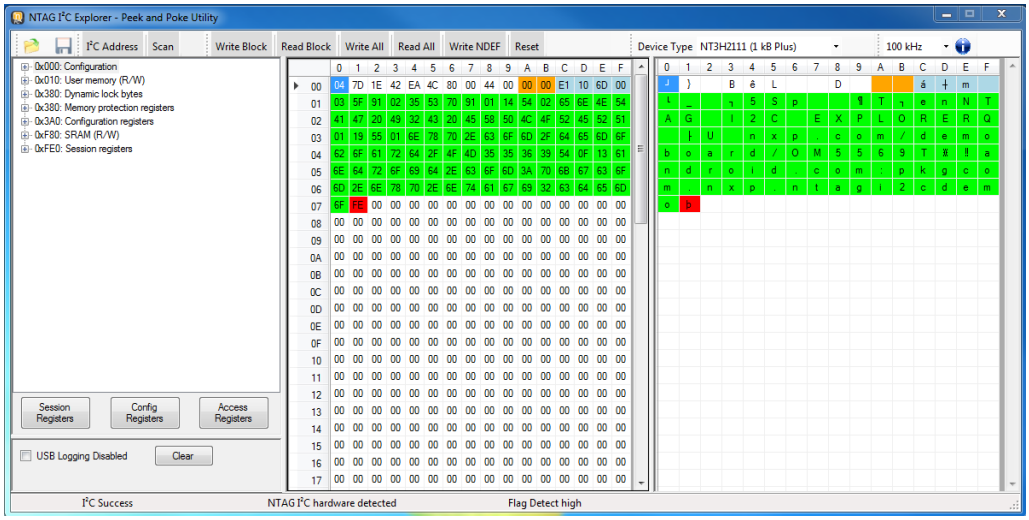

Fig 8. Peek and Poke GUI

Note: The UM10967 (refer to [UM10967]) describes in detail how to get started with the Peek and Poke tool and the different functionalities it offers.

4. NTAG I²C *plus* Explorer Demo app

The Android application is intended to operate on devices running Android version 4.0 and above. The application has been optimized for a correct visioning of the graphical elements in smartphones featuring different resolutions.

As seen in Fig 9, the Android application consists of two demos that can be launched from the main activity of the application and four configuration functionalities that are accessed from the configuration selection view in the main activity.

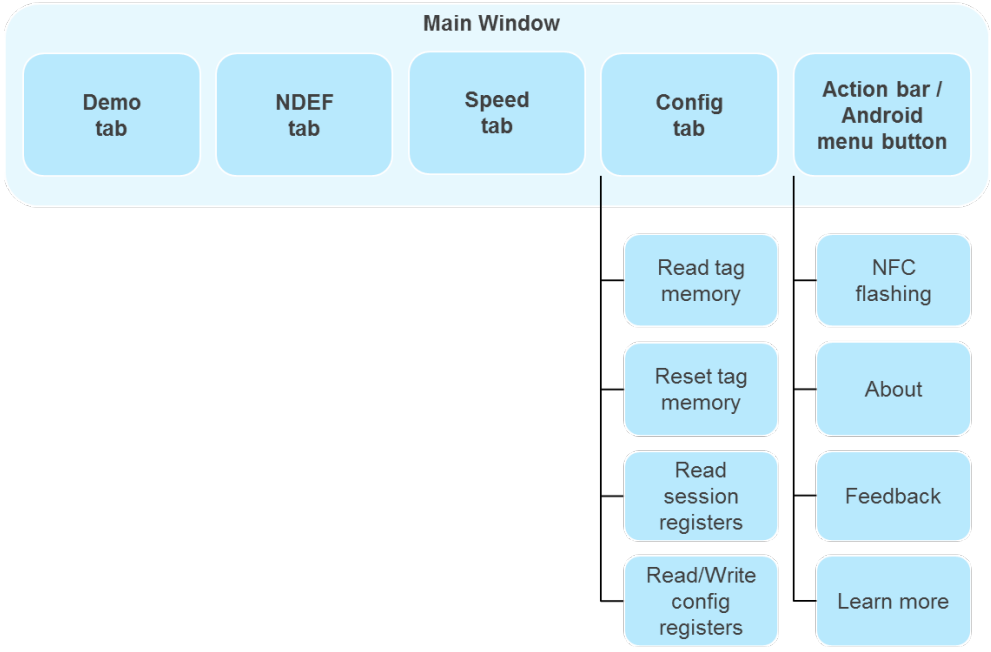

Fig 9. Application overview

4.1 Demo tab

After the Splash window closes, the Demo tab will appear. This screen allows the user to launch the Demo tab itself, the NDEF demo tab, the Speed demo tab and access the Configuration tab, with all configuration functionalities supported by the application.

Regarding the Demo tab, it allows demonstrating:

- The **Energy harvesting** functionality of the NTAG I²C *plus* that enables the powering up of the complete demo kit board with the energy harvested from the RF interface without any battery.
- The **RF to I²C** communication enabling us to modify the LED color by pressing the related color button on the NFC device screen.
- The **I²C to RF** communication as the set of push buttons pressed by the user on the demo kit board reflected on the NFC device screen.
- The dynamic **bidirectional communication** between the two interfaces as the temperature value as well as the voltage on the energy-harvesting pin get dynamically updated on both the low power screen and the NFC device screen.

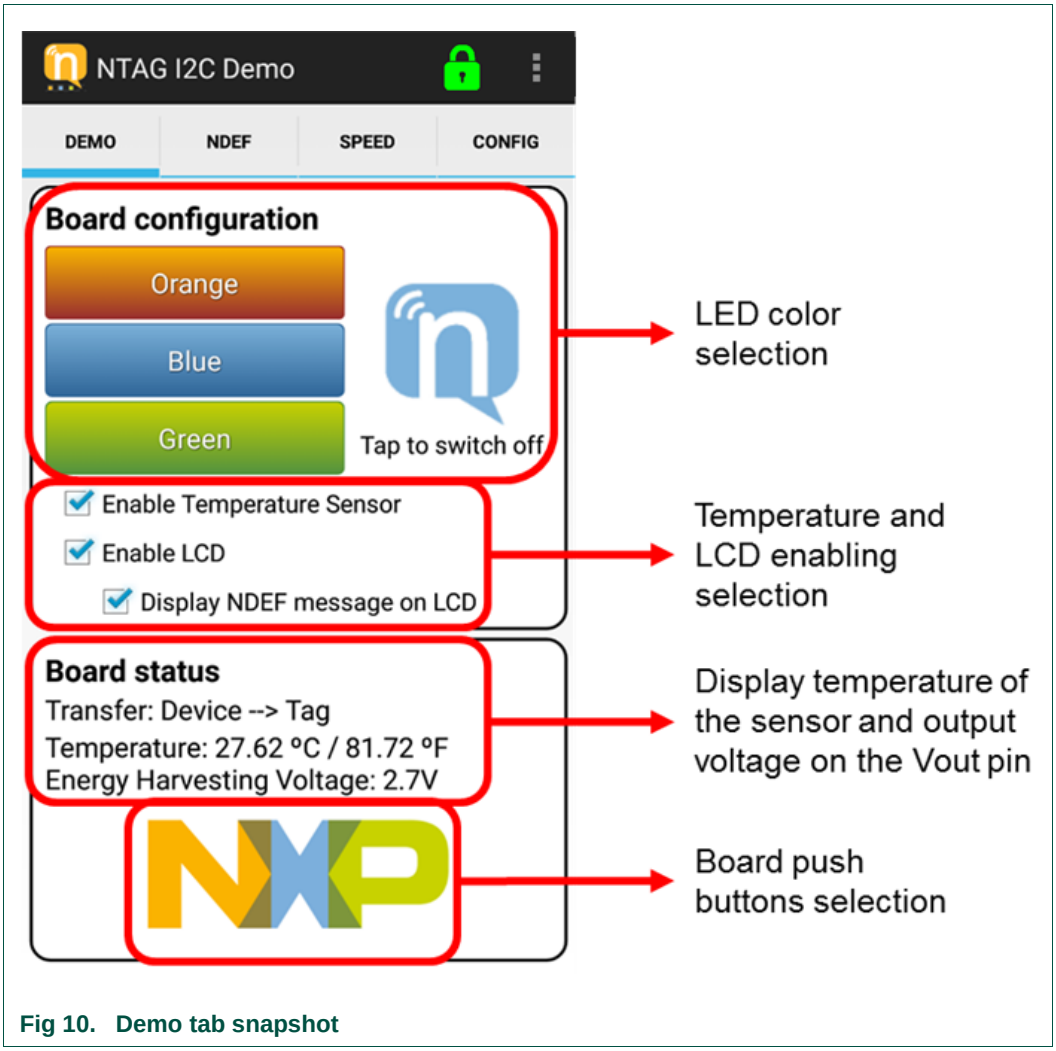

Fig 10. Demo tab snapshot

4.1.1 Configuring NEK board to demonstrate RF to I²C communication

The board configuration part of the demonstration shows how the NTAG I²C *plus* passes a command from the RF input through the SRAM and I²C serial interface output to the microprocessor, which acts upon the command and lights the appropriate LED.

1. Touch one of the colored board configuration buttons on your mobile device. The NTAG icon then changes color, indicating the color selection.
2. Tap your mobile device on the antenna. When properly placed, the LED will illuminate in the chosen color. Optionally, the LCD display will display the harvested voltage, the temperature sensed by the board, the default text or the stored NDEF message if the corresponding options are enabled in the board configuration menu.

Fig 11. Green LED board configuration selection

4.1.2 Reading board input to demonstrate I²C to RF communication

The three colored buttons on the NEK board demonstrate information from the board beings transferred from the microprocessor through the I²C serial bus to the NTAG I²C *plus*, which then sends it via the RF field to the mobile device for display. When pressed, each colored button on the NEK board will cause a corresponding shade off on the board input NXP logo.

Fig 12. Reading board input

4.1.3 Temperature sensor

The NEK board incorporates a temperature sensor for measuring the ambient temperature. This information can be shown in the Demo tab of the Android/PC application. Additionally it can be shown in the LCD display if the option is checked in the menu.

This operation demonstrates the NTAG I²C *plus* operating in pass-through mode, in which data passes through its SRAM. It is also another demonstration of passing data from the microprocessor through the I²C serial bus to the NTAG I²C *plus*, which can be read through the RF interface for display on the mobile device.

Fig 13. Temperature sensor demonstrates SRAM pass through mode

4.1.4 Enabling LCD display

You can enable the NEK board LCD by touching the *Enable LCD* selection box. When doing so, the LCD will show the predefined *NTAG I²C plus Explorer* text message, the energy harvested and, if enabled, the temperature sensed by the board in Celsius and Fahrenheit degrees.

Once again, this operation demonstrates the NTAG I²C *plus* chip operating in pass-through mode sends a command from the RF input through the I²C serial interface output to the microprocessor, which in turn acts upon the command and turns on the LCD display.

Fig 14. Enabling NEK board LCD

4.1.5 Displaying NDEF text message on the NEK board LCD

You can display the NDEF text message on the NEK board LCD by touching the *Display NDEF message on LCD* selection box.

Again, this operation demonstrates the NTAG I²C *plus* operating in pass-through mode. First, the NDEF message is read from the NTAG I²C *plus* EEPROM memory and is transferred to the MCU using the SRAM memory, which displays it in the display.

Fig 15. Displaying NDEF text message on NEK board LCD

4.2 NDEF tab

The NDEF tab allows the user to read or write an NDEF message to the NTAG I²C *plus* EEPROM. Actually, it could be used to read or write an NDEF message to any NFC Forum Type 2 Tag, such as a MIFARE Ultralight, NTAG21x, ...

On the **Read NDEF** mode, the application reads the NDEF message TLV from the NTAG I²C *plus* and returns its content and the type of NDEF message.

On the **Write NDEF** mode, the application allows the user to write a Text type, URI type, Bluetooth pairing type or Smart Poster NDEF message.

- *Text record* type is used to store plain text data.
- *URI record* type allows NFC tags to trigger actions on the NFC device (usually the smartphone), such as opening a webpage or sending an SMS message.
- *Bluetooth pairing* NDEF messages contain information about a Bluetooth device that allows the smartphone to pair with that Bluetooth device by just tapping the tag. In the case of writing a Bluetooth pairing type message, it is important to remember that the MAC address shall be 6 bytes, in hexadecimal (therefore, 12 characters from 0 to F).
- *Smart Poster* record type defines a structure including a URI record type and a Text record type as building blocks.

The NDEF tab also contains a ***Write default NDEF message*** button. When this button is pressed, the application automatically writes a NDEF Smart Poster message with the 2 records (optionally 3):

- Text record "*NTAG I2C plus EXPLORER*"
- URI content: *www.nxp.com/demoboard/OM5569*.
- Optionally: AAR (Android Application Record)

More on ***Write default NDEF message*** is described below in section 4.2.2.

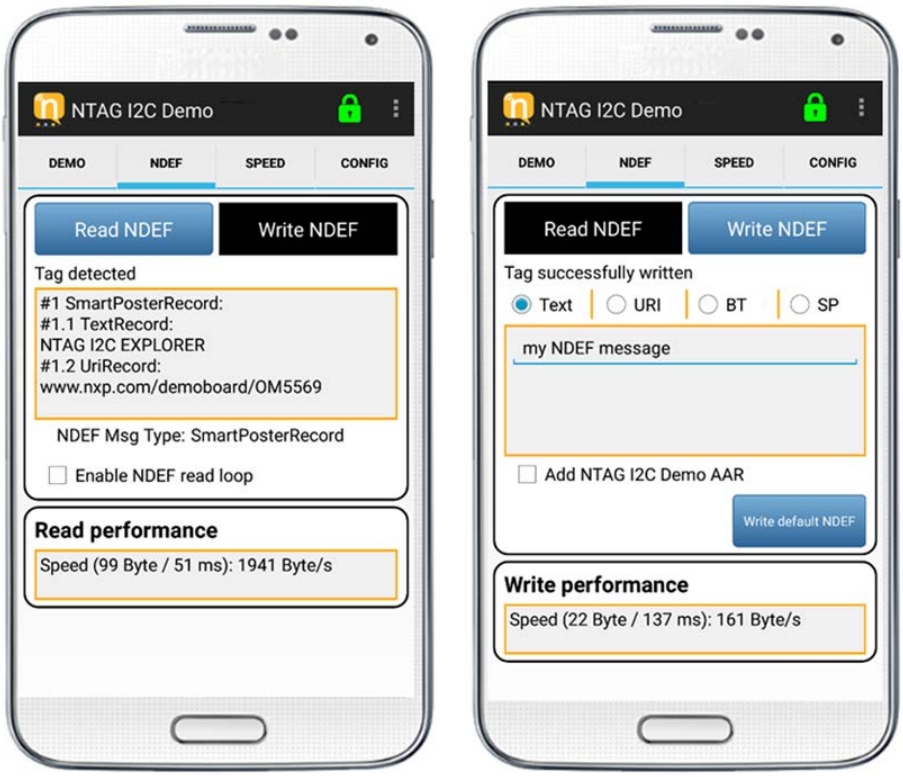

Fig 16. NDEF tab snapshot

When the NDEF message has been successfully written the “write tag successfully done” popup appears temporarily in the application informing the user that the message has been correctly written. There is no need to tap the tag again if you want to write multiple NDEF messages and read them in between.

In order to learn more about the different types of NDEF messages and what structure they have, please refer to the following NFC Forum specifications:

“NFC Data Exchange Format (NDEF) Technical Specifications”

“NFC Record Type Definition (RTD) Technical Specification”

“NFC Text RTD Technical Specification”

“NFC URI RTD Technical Specification”

“NFC Forum Connection Handover Technical Specification”

“NFC Smart Poster RTD Technical Specification”

“Bluetooth Secure Simple Pairing Using NFC”

4.2.1 Reading NDEF data

To read a NDEF message out of NTAG I²C *plus*:

1. Select *Read NDEF* from the tab on the right of the mobile device screen.
2. Tap the mobile device onto the antenna.
3. A proper read will result in a message on the mobile device indicating *Read tag successfully done*. If no NDEF message is found or NTAG I²C *plus* is not formatted as defined in NFC Forum Type 2 Tag spec, it will result in a message *NTAG I2C plus product is not NDEF formatted*. Additionally, the *Enable NDEF read loop* selection box can be used to set the application to permanently read the EEPROM memory, so any change will automatically appear in the window.

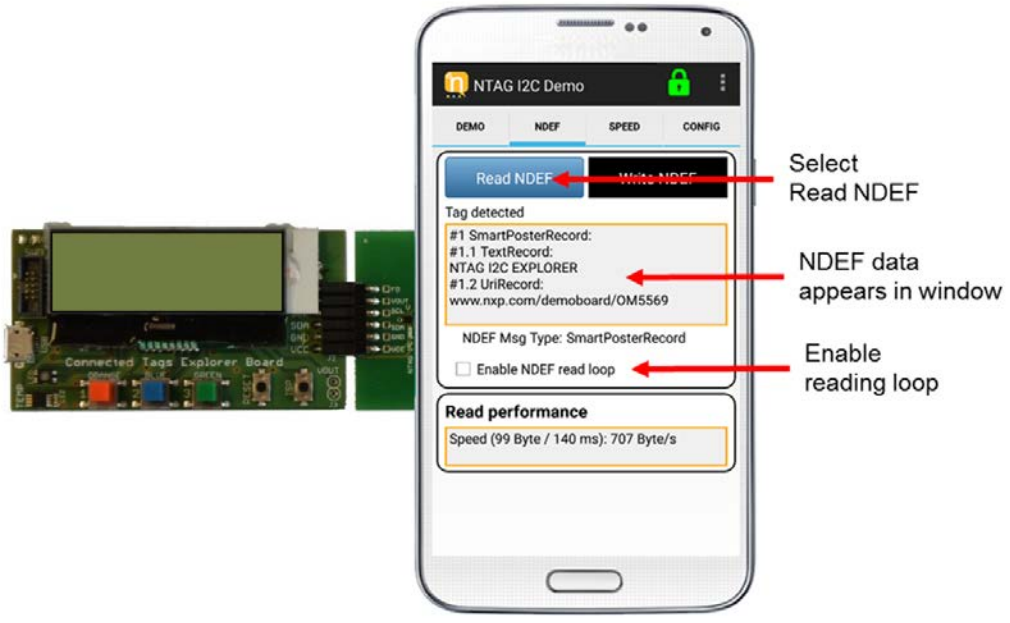

Fig 17. Reading NDEF message stored in EEPROM of NTAG I²C *plus*

4.2.2 Writing NDEF data

To write an NDEF message into the NTAG I²C *plus*:

1. Select *Write NDEF* from the tab on the right of the mobile device screen.
2. Type a message into the NDEF message area. Or alternatively, press *Write Default NDEF message* on your screen, which writes a NDEF Smart Poster message with the text content: "*NTAG I2C plus EXPLORER*" and the URI content: *www.nxp.com/demoboard/OM5569*. Additionally, you can add to your NDEF message an Android Application Record (AAR), which adds the package name of the NTAG I²C *plus* Demo application embedded inside the NDEF record. You can add an AAR to any NDEF record of your NDEF message because Android searches the entire NDEF message for AARs. If it finds an AAR, it starts the application based on the package name inside the AAR. If the application is not present on the device, Google Play is launched to download the application.
3. Tap the mobile device onto the antenna.
4. A proper write will result in a message on the mobile device indicating *write tag successfully done*.

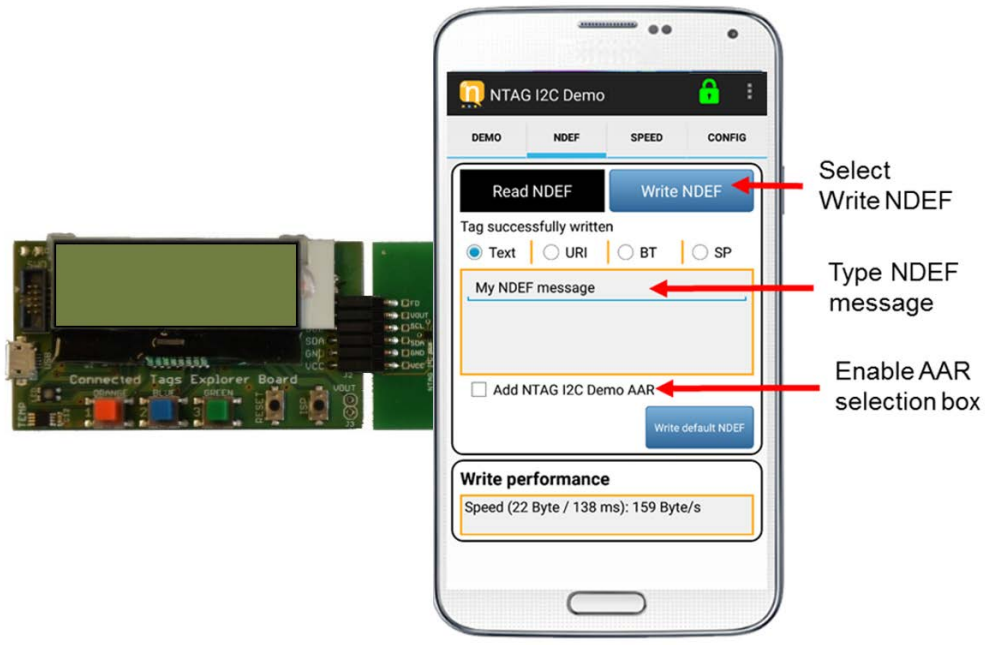

Fig 18. Writing NDEF message to NTAG I²C *plus* EEPROM

4.3 Speed tab

The speed Test Demo measures the transfer bit rate when communicating with the NTAG I²C *plus* Explorer board according to different configurations.

4.3.1 SRAM speed test

The SRAM speed Test measures the speed at which the data is transferred from the application to the microcontroller through the SRAM (with the NTAG I²C *plus* in pass-through mode), and vice versa. First, the application writes data to the SRAM several times, and the microcontroller reads from it. This way, the throughput of the communication in pass-through mode from the application to the microcontroller can be obtained. Once this first test has finished, the microcontroller starts writing to the SRAM memory, while the application reads from it. This way, the throughput from the microcontroller to the application can be measured.

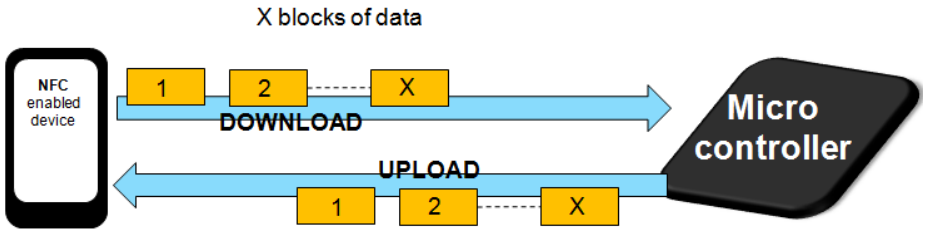

Fig 19. Data transfer speed measurement concept behind the SRAM selection

Since the size of the SRAM memory is 64 bytes, the data to be transmitted and received has to be a multiple of 64. Actually, what is transmitted is the number of 64-byte blocks defined by the user in the Block multiplier field (10 by default).

The integrity of the data transferred in both directions is checked by appending a CRC32 value in the last block. The CRC32 is calculated for the whole message that has been transferred (for all the blocks). If the CRC32 from the message received by the application is right, it will show an “Integrity of the data: OK” message. On the other hand, if the CRC32 from the message received by the board is right, it will turn on the green LED at the end of the Speed Test.

Once the test has been finished, both the application and the microcontroller indicate whether the integrity check was successful (through a green LED at the microcontroller), and the application shows the time, mean speed and data transferred for both directions.

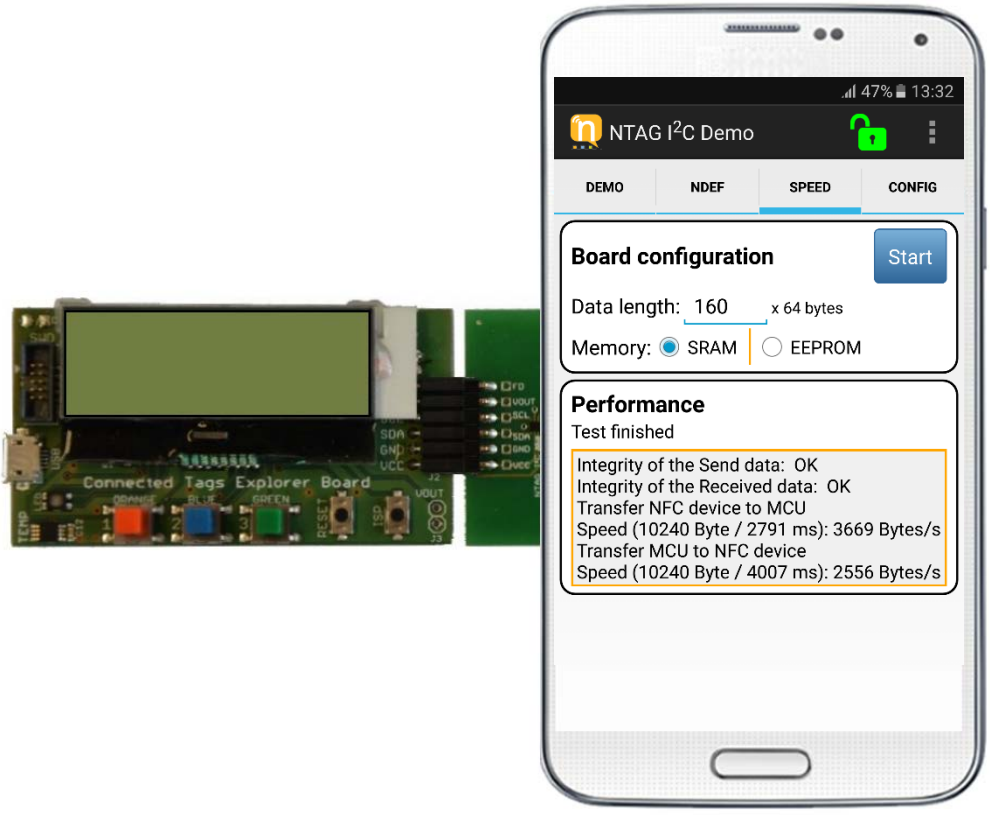

Fig 20. SRAM Speed test snapshot

4.3.2 EEPROM speed test

The EEPROM Speed Test measures the speed at which the application is able to write and read the EEPROM memory of the NTAG I²C *plus*. In order to do this, an NDEF message is written to the EEPROM, and then, the same NDEF message is read. The written NDEF message is a text type message that contains the data in the textbox of the application repeated as many times as indicated in the Block multiplier field.

First, the application creates the NDEF message to be written by creating a string that contains the content from the textbox as many times as indicated, and adding the appropriate header. Then, it writes it to the EEPROM memory by sending as many NFC Forum Type 2 Tag Write commands as necessary, and measuring the time it takes to do so. Once it has finished writing, it reads the NDEF message from the NTAG I²C *plus* EEPROM memory.

Once the test is finished, the application shows the number of bytes, mean speed and time for both the reading and writing process. The user can also check the content of the memory to ensure that the NDEF message has been written appropriately.

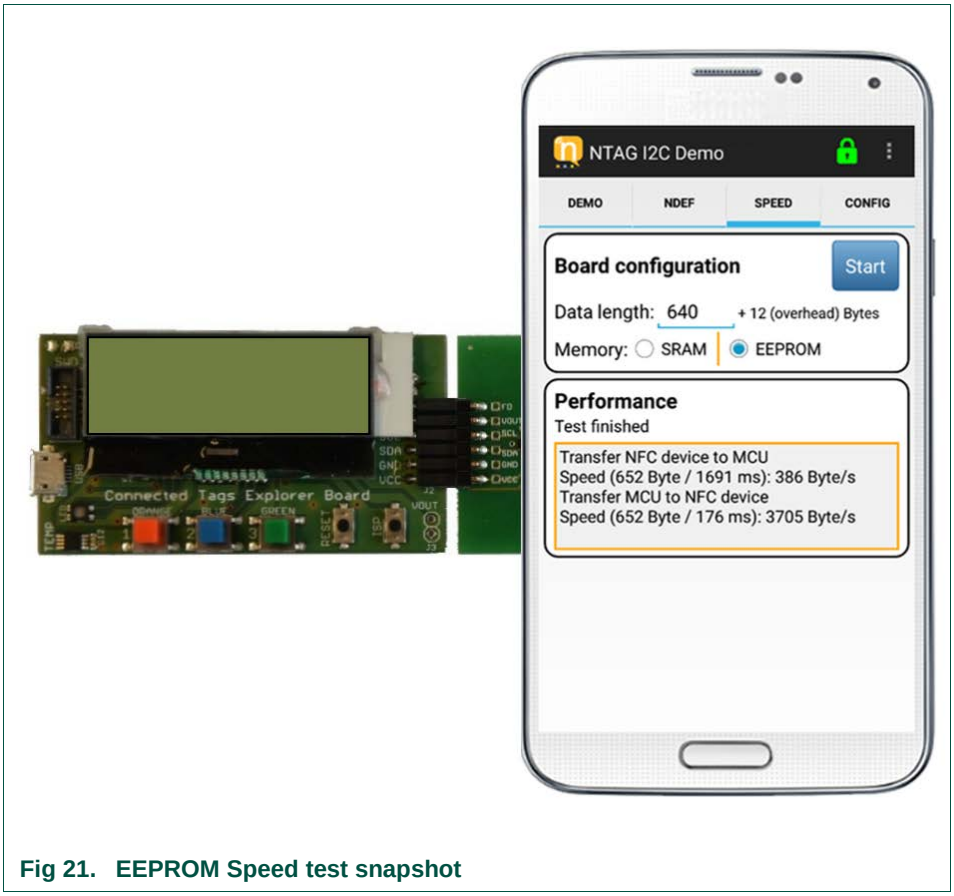

Fig 21. EEPROM Speed test snapshot

Comparison between EEPROM read/write Fig 21 and SRAM read/write Fig 20 is represented. Performing operations on SRAM memory with FAST_READ and FAST_WRITE commands, highly improve overall communication speed.

4.4 Configuration tab

This tab shows a selection menu that provides access to the different configuration activities of the NTAG I²C *plus* supported by the application. From this screen, the user may select *Read Tag Memory*, *Reset Tag Memory*, *Read Session Registers* or *Read/Write Configuration registers.*

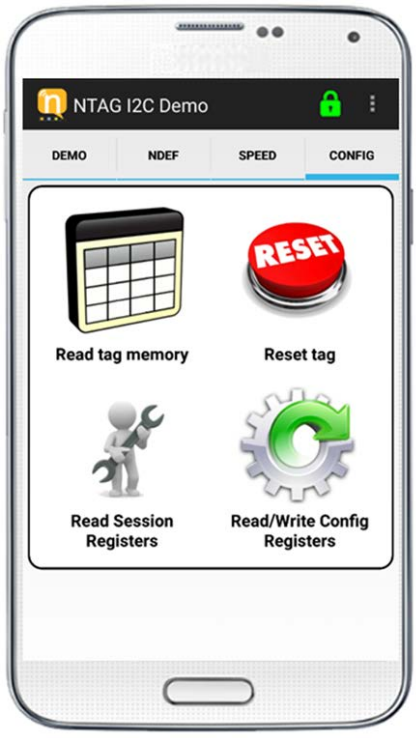

Fig 22. CONFIG tab menu

4.4.1 Reading tag memory

This option reads and displays on the screen the content of the whole memory of the NTAG I²C *plus* product. The complete content can be scrolled on the screen. To read the tag memory:

1. Select *Read Tag Memory* from the Configuration tab.
2. Tap the mobile device to the antenna (it might take 2-3 seconds to read the memory content.
3. The screen will display the entire memory contents.

For security reasons, the pages from 0xE2 to 0xFF in the sector 0 are hidden when reading all the memory. These bytes define the PWD, PACK, AUTH0, PT_I2C and ACCESS bytes.

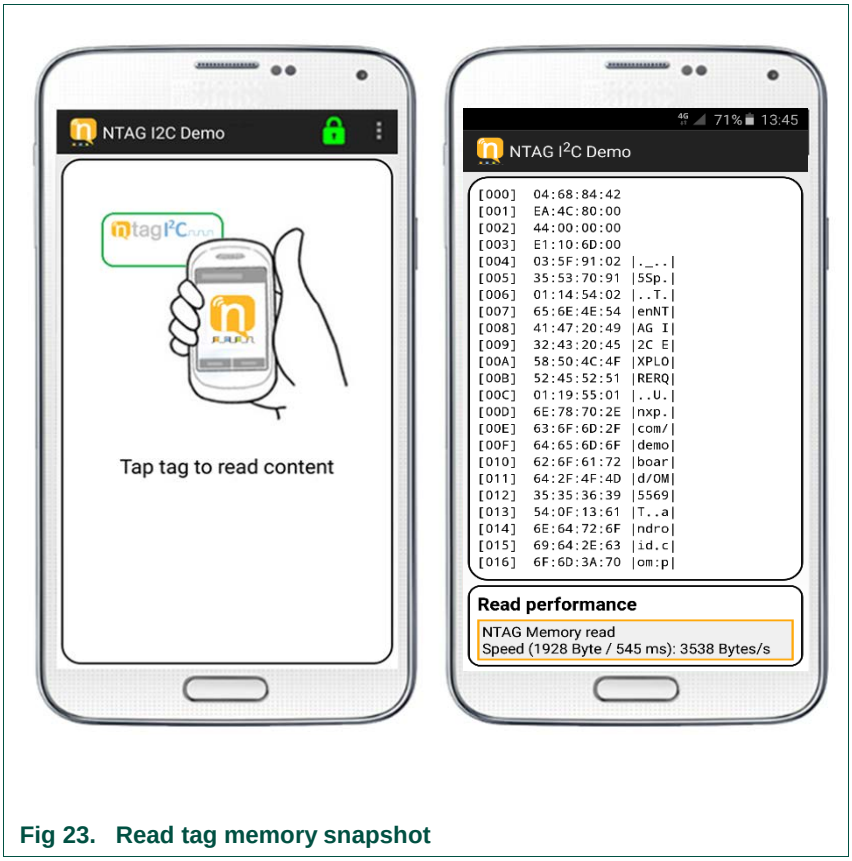

Fig 23. Read tag memory snapshot

4.4.2 Resetting tag memory

This configuration functionality is about resetting the whole user memory of the NTAG I²C *plus* product to the original content that was programmed during production.

1. Select *Reset Tag* memory from the *Config* tab screen.
2. Tap the mobile device to the antenna tag. The user should tap the NTAG I²C *plus* device for 2-3 seconds to reset the memory content.
3. Upon successful reset, a banner indicating completion will be displayed along the bottom of the screen.

In addition, when the user runs the reset memory on the NTAG I²C PLUS version, it will write the PWD to 0xFFFFFFFF, AUTH0 to 0xFF, PACK to 0x0000, ACCESS to 0x00 and PT_I2C to 0x00. These are the default values for the access configuration.

Fig 24. Reset tag snapshot

Alternatively, we can also reset the tag contents if we keep the second button pressed and tap the phone with the NFC communication enabled. When the LED is set to green, the tag memory has been reset to the default NDEF SmartPoster.

Fig 25. Alternative way to reset tag

4.4.3 Reading session registers

This option displays the content of the session registers in the NTAG I²C *plus*. Session registers are used to configure or monitor the registers values of the current communication session. Session registers values can be modified within a particular communication session (only via the I²C interface, so the application is not able to modify them). However, after Power-On-Reset, these values go back to their default configuration values, which are obtained from the configuration registers.

Session registers values can be read on pages F8h to F9h (sector 3) via the RF interface or at block FEh via the I²C interface.

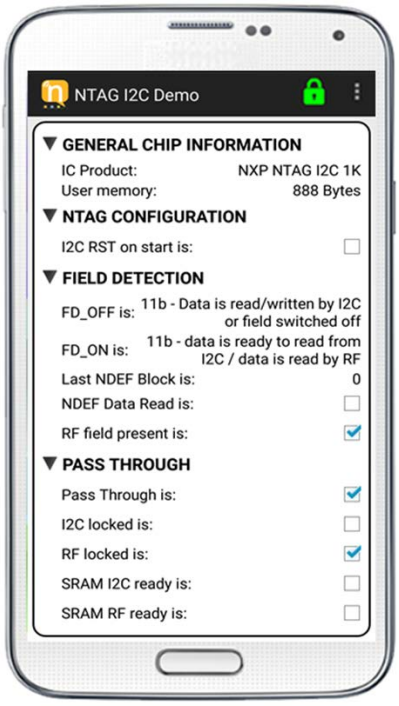

Fig 26. Read Session registers snapshot

To read the tag session registers:

1. Select *Read Session Registers* from the Config tab screen.
2. Tap the mobile device to the tag antenna.
3. Upon successful read, the session registers values will be shown. Tapping on any of the right-facing arrows will bring up more details. A blue mark indicates ON or selected.

Session register values are displayed on the screen divided into different groups:

- **General Chip Information**: General information about the discovered NTAG I²C *plus* product. It shows the following information:
  - **IC product**: NTAG I²C *plus* chip version.
  - **User Memory**: Size of the user memory of the NTAG I²C *plus* chip.
- **NTAG Configuration:** General NTAG configuration:
  - **I²C RST on start is:** Shows the content of the I²C_RST_ON_OFF bit, which is responsible for enabling a soft reset through an I²C repeated start.
- **Field Detection:** Information related to the field detection functionality, which is used for the smart pairing with devices, as it triggers a signal to the connected host when the NTAG I²C *plus* product is powered by an external NFC device.
  - **FD_OFF:** Shows the content of the FD_OFF bits, which define the event upon which the signal output on the field detection pin is brought up.
  - **FD_ON:** Shows the content of the FD_ON bits, which define the event upon which the signal output on the field detection pin is brought down.
  - **Last NDEF Page is:** Shows the content of the LAST_NDEF_BLOCK byte, this is the address of the last block (I²C interface addressing) of the NDEF message.
  - **NDEF Data Read is:** Shows the content of the NDEF_DATA_READ bit.
  - **RF Field present is:** Shows the content of the RF_FIELD_PRESENT bit, which indicates if an RF field is detected.
- **Pass through:** Information related to the pass-through functionality, which allows the fast transfer of data between the RF and the I²C interface by using a 64 byte SRAM memory.
  - **Pass Through is:** Shows the content of the PTHRU_ON_OFF bit, which is responsible for enabling the data transfer via the SRAM memory.
  - **I²C locked is:** Shows the content of the I²C_LOCKED bit, which can lock the access to memory to the I²C interface.
  - **RF locked is:** Shows the content of the RF_LOCKED bit, which can lock the access to memory to the RF interface.
  - **SRAM I²C ready is:** Shows the content of the SRAM_I²C_READY bit, which indicates if the data in the SRAM memory is ready to be read by the I²C interface.
  - **From RF to I²C is:** Shows the content of the PTHRU_DIR bit, which defines the data flow direction for the data transfer.
- **SRAM Memory settings:** Information about the SRAM mirroring feature. The SRAM memory can be mirrored in the User Memory for RF access by enabling the SRAM mirroring feature.
  - **SRAM Mirror:** Shows the content of the SRAM_MIRROR_ON_OFF bit.
  - **SRAM Mirror block:** Shows the content of the SRAM_MIRROR_BLOCK byte, which indicates the address of the first block (I²C interface addressing) of the mirror of SRAM memory.

- **I²C Settings:** Information about the I²C management configuration.
  - **WD_LS Timer is**: Shows the content of the WDT_LS byte: the LSB of the watchdog time control register.
  - **WD_MS Timer**: Shows the content of the WDT_MS byte: the MSB of the watchdog time control register.
  - **I²C Clock stretch**: Shows the content of the I²C_CLOCK_STR bit, which is responsible for enabling the I²C clock stretching.

For further information about the session registers bytes, please refer to the NTAG I²C *plus* product data sheet (refer to [NTAGI2Cplus]).

4.4.4 Reading / Writing configuration registers

Configuration registers define the default functionalities of the NTAG I²C *plus* to be used for the communication after a Power-On-Reset.

Configuration registers values can be read and written in Pages E8h and E9h (sector 0 for NTAG I²C *plus* 1k). The operation to be performed, read or write, in each tap is selected by pressing the Read/Write config buttons on top of the window.

For further information about the configuration registers bytes, please consult the NTAG I²C *plus* product datasheet (refer to [NTAGI2Cplus]).

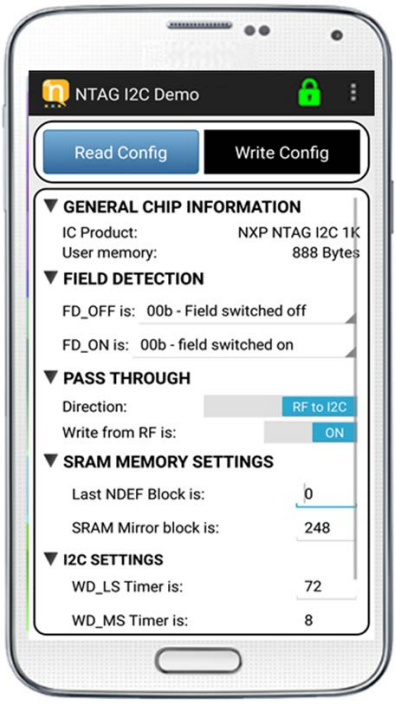

Fig 27. Read / Write configuration registers snapshot

To read configuration registers:

1. Select *Read/Write Config* registers from the Config tab screen.
2. Tap the mobile device to the antenna tag.
3. Upon successful read, the configuration registers values will be shown.
4. To write the configuration registers, switch to Write config. Register contents are controlled either via dropdown menus, direct input or by slider controllers where grey indicates OFF and blue indicates ON.

Configuration register values are displayed on the screen divided into different groups:

- **General Chip Information**: General information about the discovered NTAG I²C *plus* product. It shows the following information
  - **IC product**: NTAG I²C *plus* chip version
  - **User Memory**: Size of the user memory of the NTAG I²C *plus* chip
- **Field Detection:** Information related to the field detection functionality, which is used for the smart pairing with devices, as it triggers a signal to the connected host when the NTAG I²C *plus* product is powered by an external NFC device.
  - **FD_OFF:** Shows the content of the FD_OFF bits, which define the event upon which the signal output on the field detection pin is brought up
  - **FD_ON:** Shows the content of the FD_ON bits, which define the event upon which the signal output on the field detection pin is brought down
- **Pass through:** Information related to the pass-through functionality, which allows the fast transfer of data between the RF and the I²C interface by using a 64 byte SRAM memory.
  - **Functionality:** Shows the content of the PTHRU_ON_OFF bit, which is responsible for enabling the data transfer via the SRAM memory.
  - **Direction:** Shows the content of the TRANSFER_DIR bit, which defines the data flow direction for the data transfer.
  - **Write from RF:** Shows the content of the TRANSFER_DIR bit (same as direction), which defines the data flow direction for the data transfer
- **SRAM Memory settings:** Information about the SRAM mirroring feature. The SRAM memory can be mirrored in the User Memory for RF access by enabling the SRAM mirroring feature.
  - **Last NDEF Page:** Shows the content of the LAST_NDEF_BLOCK byte, this is the address of the last block (I²C interface addressing) of the NDEF message.
  - **SRAM Mirror:** Shows the content of the SRAM_MIRROR_ON_OFF bit.
  - **SRAM Mirror block:** Shows the content of the SRAM_MIRROR_BLOCK byte, which indicates the address of the first block (I²C interface addressing) of the mirror of SRAM memory.
- **I²C Settings:** Information about the I²C management configuration.
  - **WD_LS Timer**: Shows the content of the WDT_LS byte: the LSB of the watchdog time control register

- **WD_MS Timer**: Shows the content of the WDT_MS byte: the MSB of the watchdog time control register
- **I²C Clock stretch**: Shows the content of the I²C_CLOCK_STR bit, which is responsible for enabling the I²C clock stretching.
- **I²C RST on start:** Shows the content of the I²C_RST_ON_OFF bit, which is responsible for enabling a soft reset through an I²C repeated start.

- **Access Configuration:** Information about the I²C management configuration.
  - **AUTH0**: This is the first page of the Sector 0 where the authentication is needed in order to access the user memory from NFC perspective. By default the AUTH0 value is 0xFF, this means that the password protection is effectively disabled. When the user introduces a new password this AUTH0 will be automatically set to 0x03.
  - **NFC Prot**: Shows the memory protection bit. If set to OFF only the write to the protected area is protected by the password. If set to ON, read and write into the protected area is protected by password.
  - **NFC DIS SEC1**: Shows the NFC access protection to Sector 1. If set to OFF the Sector 1 will be accessible in 2K version. If set to ON, Sector 1 is not accessible and will return NACK0.
  - **AUTHLIM:** Shows limitation of negative password authentication attempts. If the limit is reached then the protected area will no longer be accessible. Set to 0 means that limitation is disabled.
  - **2K Prot:** Shows the password protection for Sector 1 for 2k version. If set to OFF, the password authentication for Sector 1 is disabled. Otherwise, if set to ON the password is needed to access to Sector 1.
  - **SRAM Prot:** Shows the password protection for pass-through and mirror mode. If set to OFF the password authentication for pass-through mode is disabled. If set to ON then the password authentication to access SRAM in pass-through is required.
  - **I2C Prot:** Shows the access to protected area from I²C perspective. If 0 then the whole user memory is accessible from I²C. If this value is 1 then the protected area is read only. Other value will disable the access to protected area.

For further information about the configuration registers bytes, please refer to the NTAG I²C *plus* product datasheet (refer to [NTAGI2Cplus]).

4.5 Action bar

4.5.1 NFC Flashing via NTAG I²C *plus* Explorer demo

NTAG I²C *plus* enables data download capability. The on-chip SRAM permits temporary storage of data during the transfer, enabling the tag to act as a modem.

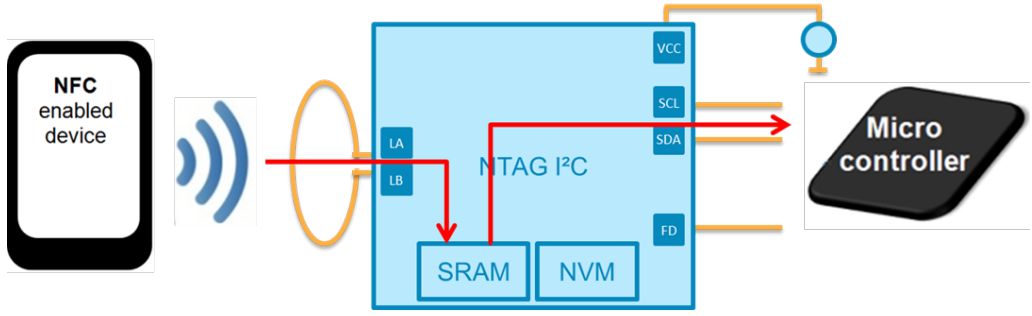

Fig 28. Figure title here

4.5.1.1 Design implementation choice

The NEK board embeds a LPC11U24. The following table shows the correspondence between sector numbers and memory addresses for LPC11U2x/1x devices.

Sector number	Sector size [kB]	Address range	LPC11U12/ LPC11U22	LPC11U13/ LPC11U23	LPC11U14/ LPC11U24
0	4	0x0000 0000 - 0x0000 0FFF	yes	yes	yes
1	4	0x0000 1000 - 0x0000 1FFF	yes	yes	yes
2	4	0x0000 2000 - 0x0000 2FFF	yes	yes	yes
3	4	0x0000 3000 - 0x0000 3FFF	yes	yes	yes
4	4	0x0000 4000 - 0x0000 4FFF	-	yes	yes
5	4	0x0000 5000 - 0x0000 5FFF	-	yes	yes
6	4	0x0000 6000 - 0x0000 6FFF	-	-	yes
7	4	0x0000 7000 - 0x0000 7FFF	-	-	yes

Fig 29. LPC11U1x/2x flash sectors

The on-chip flash memory of the LPC11U24 is grouped in sectors. The flash memory is divided into 8 sectors of 4 Kb each. Therefore, for LPC11U24 we have 32Kb of Flash memory available.

NXP's LPC11U24 microcontroller provides two methods to update the flash contents:

- In-system-programming (ISP): We use this method when we flash new contents using the USB port and a computer (drag-and-drop a binary file).
- In application programming (IAP): Programming is performed by erase and write operation on the on-chip flash memory, as directed by the end-user application code → this is the method we use for NFC flashing.

A secondary bootloader has been implemented in order to enable firmware flashing through NFC interface. A secondary bootloader is a piece of code, which allows a user application code to be downloaded via alternative channels to the USB (i.e.: NFC flashing).

The primary bootloader is the firmware that resides in a microcontroller's boot ROM block and is executed on power-up and resets. After the boot ROM's execution, the secondary bootloader is executed. The secondary bootloader in turn will then execute the end-user application.

Both the secondary bootloader and the user application reside in flash. Therefore, for the secondary bootloader to flash the user application without modifying any of its own code, the user application should be flashed starting the next available sector.

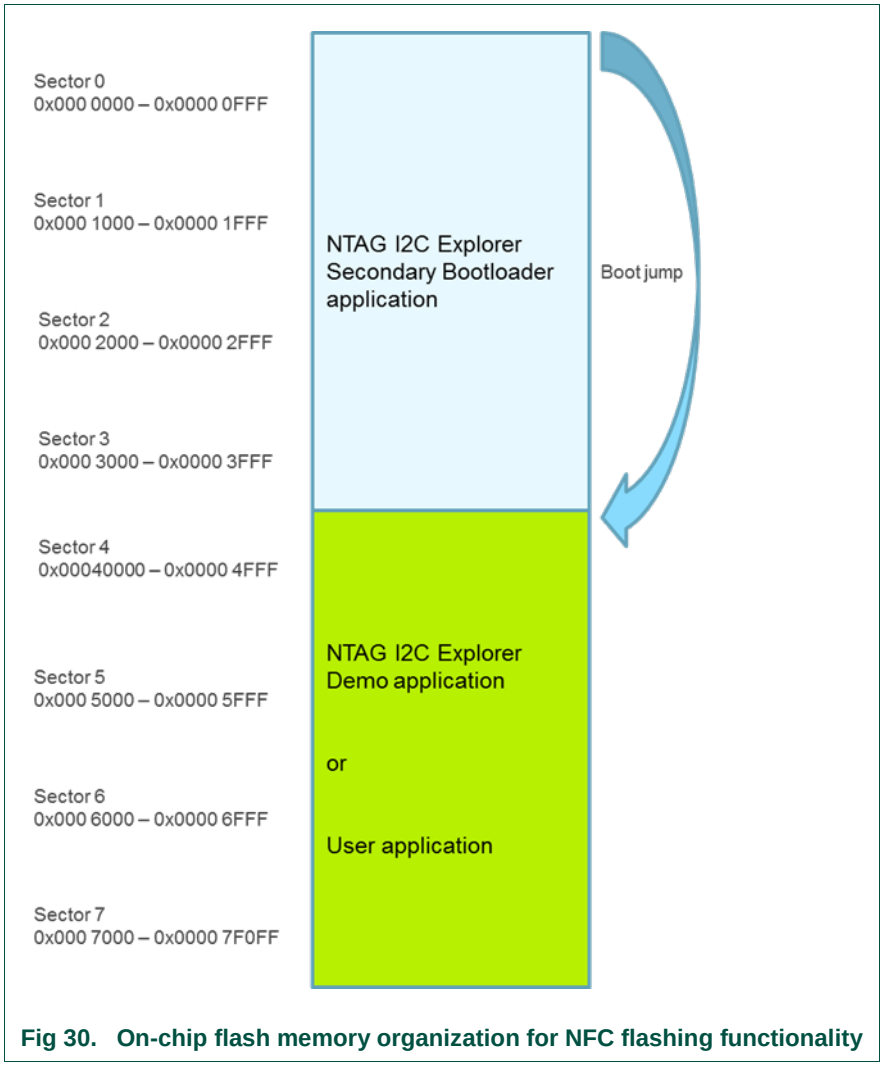

Fig 30. On-chip flash memory organization for NFC flashing functionality

The NEK secondary bootloader implements the IAP functionality that allows to erase/write flash memory sectors directed by the application and the HID drivers for USB communication. In our case, the NEK secondary bootloader occupies 4 flash sectors. This means that this four first memory sectors are never erased since they belong to the secondary bootloader (they must always be there in order to have the NFC flashing functionality available).

Therefore, we have 4 extra flash sectors available for end-user application (16 Kb). In this memory area, we store the NEK Demo (LED, Speed Tests, etc.), the blink application or any other user application. NFC flashing functionality actually re-writes this 16 Kb with the user application send by the Android or PC app via NFC interface. **For this reason, the maximum size of the binary file that can be upgraded through the RF interface is 16 KB.**

4.5.1.2 How to flash new firmware through NFC

As an illustrative example, we are going to flash the provided Blink application via NFC flashing (green LED blinks when detects the presence of the RF field). After this process, the default NTAG Explorer Demo will be replaced by the Blink application

Pre-condition:

(1) Install NTAG I²C *plus* Demo Android application v1.7.4

(2) Verity that NEK board firmware version is v3.4

Main flow:

(1) Tap on the Context menu button on your phone or in the app (depending on which phone you are using)

(2) Select the NFC Flashing option in the Android app

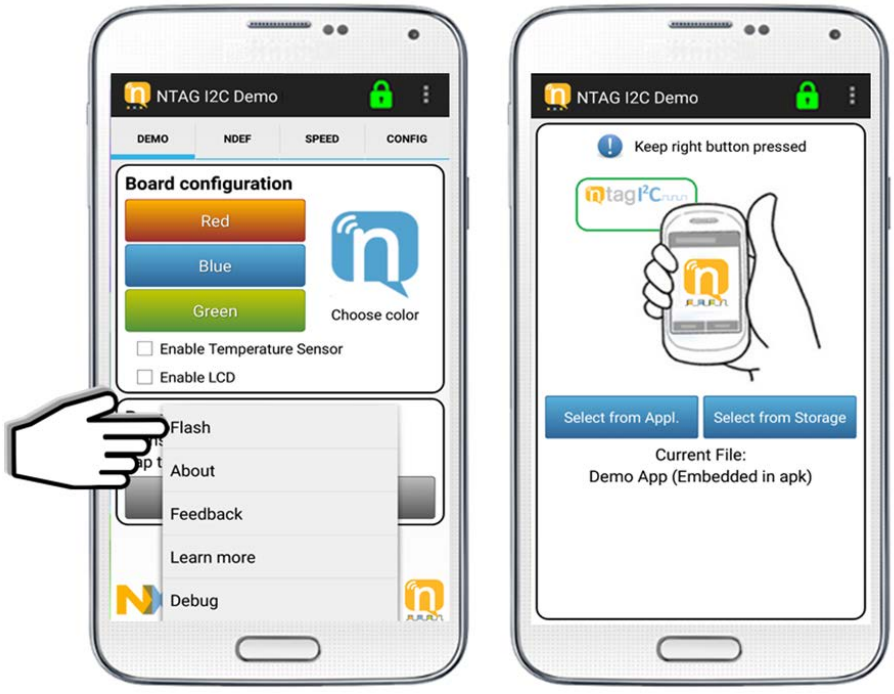

Fig 31. NFC Flashing option selection within the Android app

(3) Once in the NFC Flashing menu, we need to select the binary file to be flashed into the NEK board MCU. We have two options:

- Select the Blink application or Demo application embedded by default in the Android app by clicking on “Select from Appl” button.

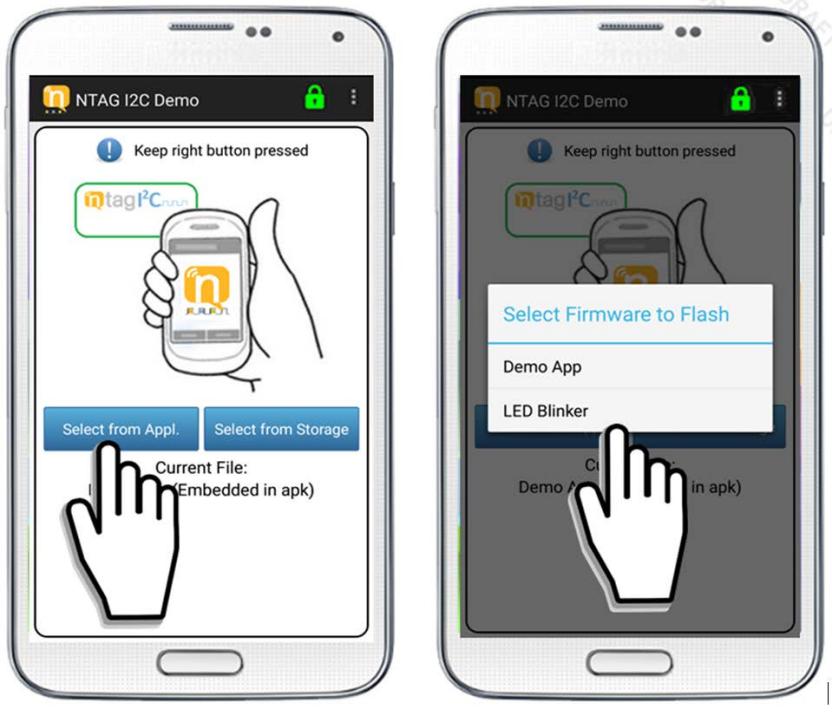

Fig 32. Selection of binary file to be flashed into MCU memory

- Alternatively, we can select from the phone memory a binary file by clicking on the “Select from Storage” button.

Fig 33. Selection of binary file to be flashed into MCU memory (II)

(4) NFC Flashing mode is enabled in the NEK board by **pressing the third colored button (right button) before power up**. While keeping the button pressed, we tap the NEK board into the phone NFC sweet spot.

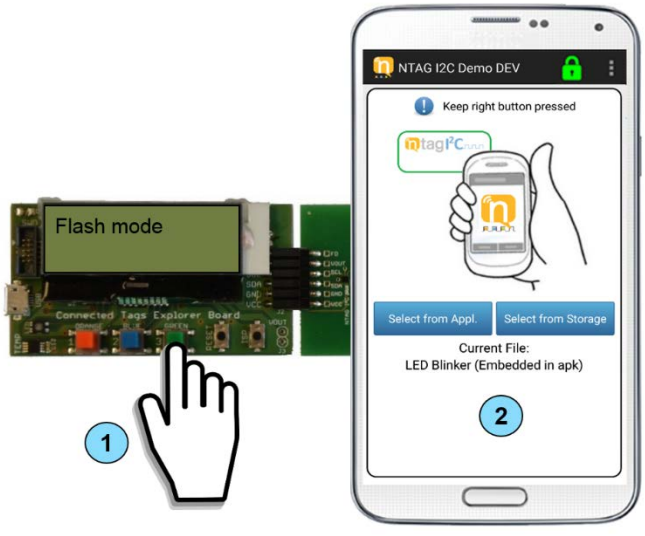

Fig 34. Entering NFC Flashing mode

As soon as we tap the NEK board, the NFC flashing will start. Once the NFC Flashing process has started, we can release the right button. This process may take some seconds, it is important to keep the phone steady until the NFC flashing process has finished (as indicated by the progress bar and green blinking LED).

In this example, we flash the Blink application binary file into the MCU memory. For doing so, we need to click on *Select from Appl* button and click on *LED Blinker*.

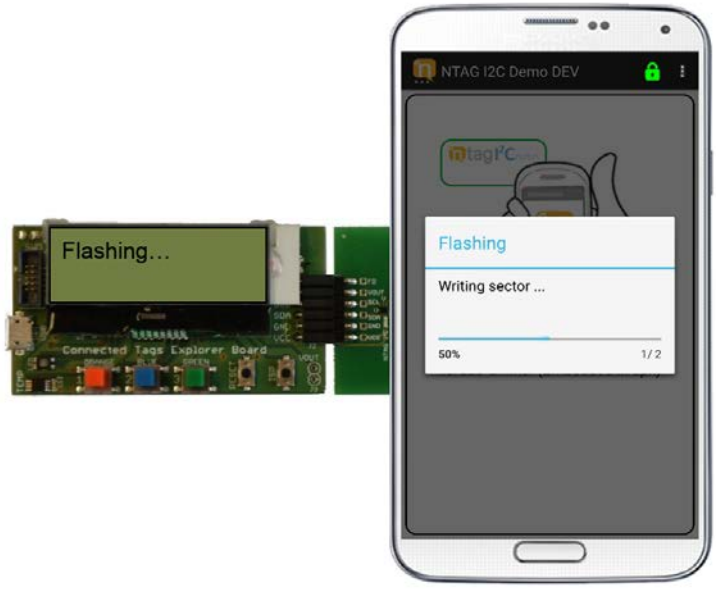

Fig 35. NFC Flashing progress bar

(5) When the NFC Flashing process has finished, both the Android app and the NEK board notifies it to the user. In the Android app, a message appears on the bottom part of the screen. The NEK board blinks the green LED three times and displays “Flashing OK”.

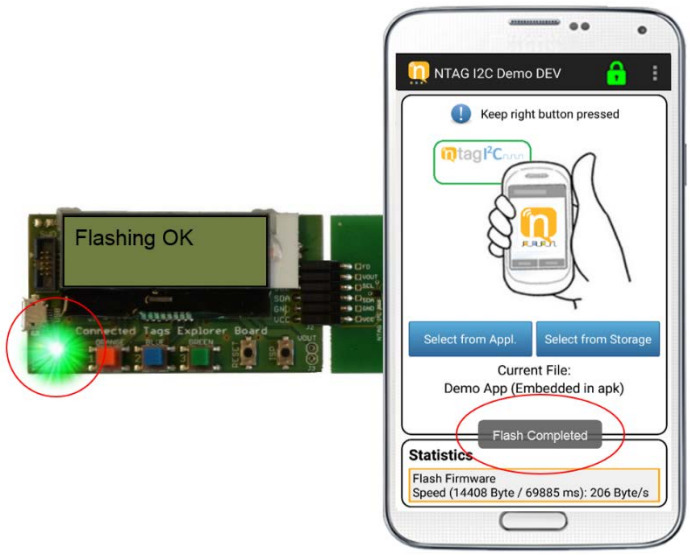

Fig 36. NFC Flashing successful execution

(6) If the process has finished successfully, we can now check if the Blink application is installed and has replaced the previous Demo app.

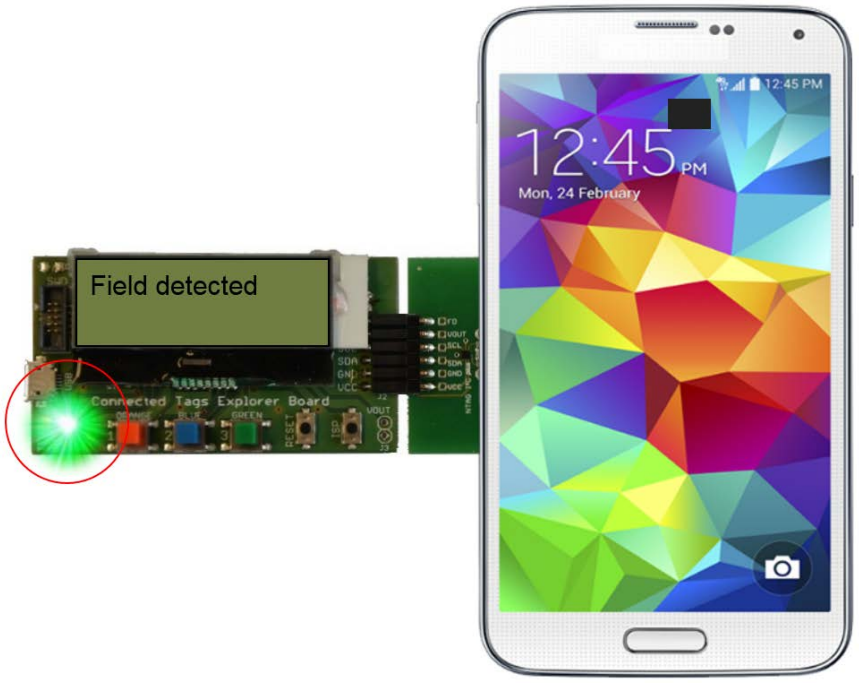

Fig 37. LED Blinker NFC Flashing upgrade verification

(7) If the process has not finished successfully, (e.g. due to an RF field lost), we can retry as many times as required. For that, we need to go back to Step 1 and repeat the entire process. Note that, if the flashing process has not finished successfully, the previous installed firmware may be not usable, and therefore, we need to retry flashing a new firmware.

4.5.2 About

The about functionality can be used to get the version being used for the board design, firmware of the board and the Android app by tapping the board with the NFC enabled device.

Fig 38. About tab snapshot

4.5.3 Feedback

This option allows you to report feedback on the Android app directly to the NXP support team.

4.5.4 Learn more

This option will bring you to a set of links with further information about NXP NTAG I²C *plus* tag, application notes, and design files and to the Android app source code.

All these functions can be found in the context menu

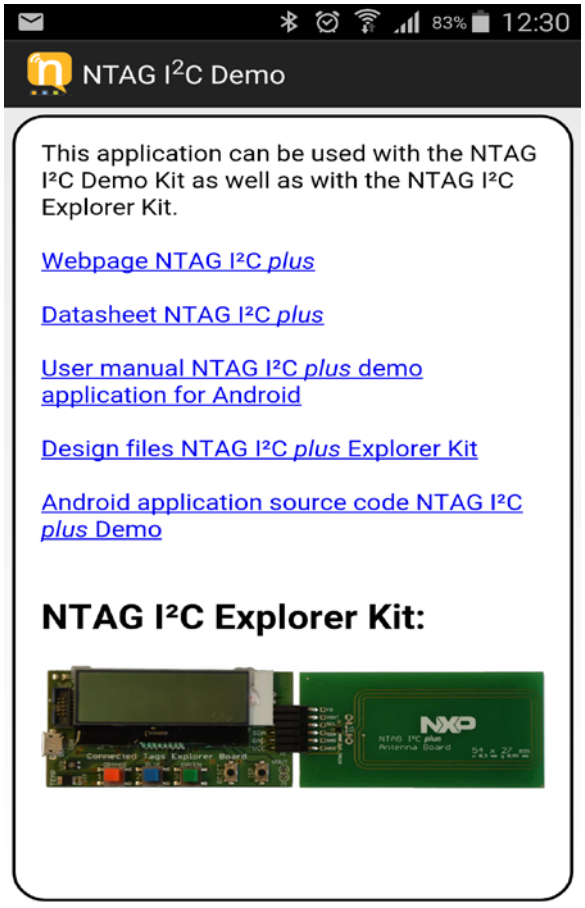

Fig 39. *Learn more* snapshot

4.6 Authentication

The NTAG I²C Plus offers authentication to protect memory operations. The device can be locked or not. If it is not locked, the user is able to perform any memory operation. If the device is locked, the user must introduce the correct password to authenticate and access the memory.

The application provides the feature to lock and unlock the device. In the upper right side of the screen there is a locker icon. In case the device is a NTAG I²C, which has no authentication feature, the locker will be white and open. This means, the user can interact will all memory. Nonetheless, if the device is a NTG I²C *plus*, the following states are possible: the device can be locked or unlocked, and the device can be authenticated or not. In the case, the device is unlocked, there is no need of authentication. This is shown with an open green locker. On the other side, if the device is locked, for a non-authenticated state, the icon shown is a red closed locker, while if it is authenticated then is a green closed locker. The different states are shown in Fig. 40.

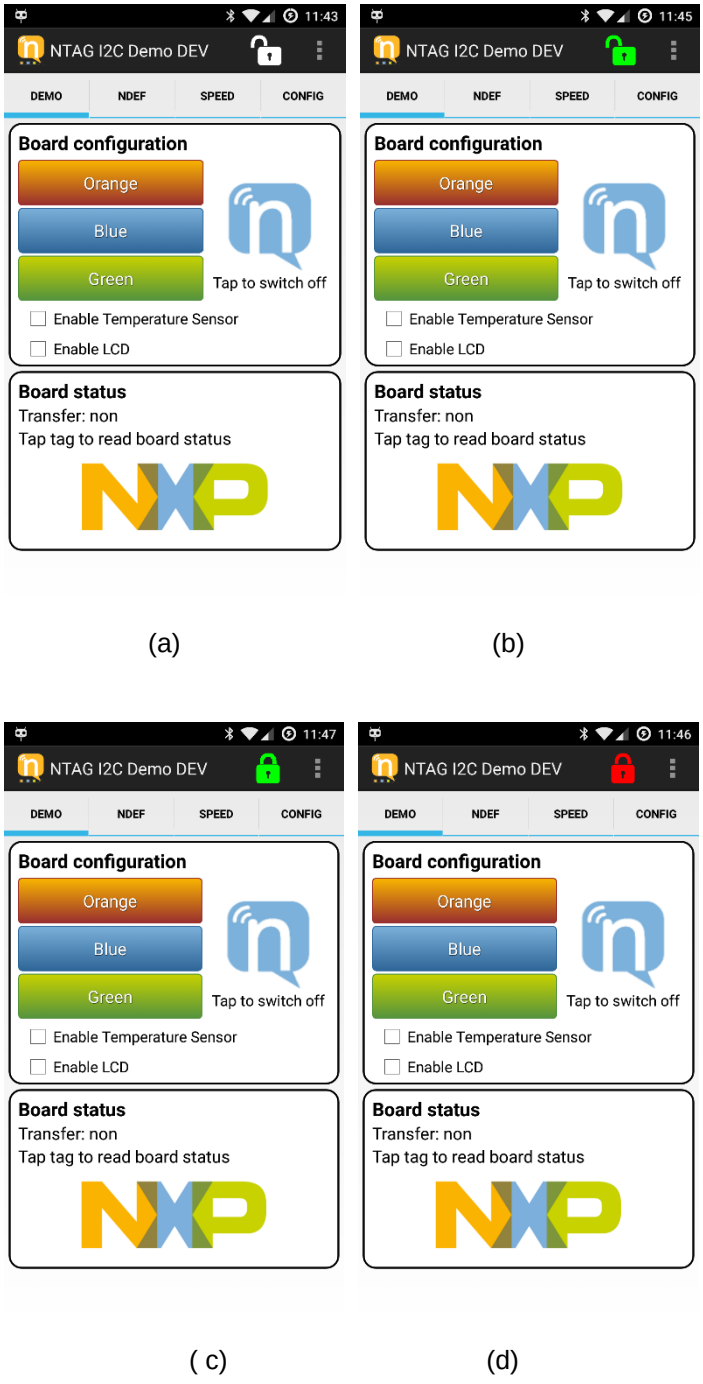

Fig 40. *Authentication States* snapshot. (a) NTAG I²C Tag without authentication feature. The rest are NTAG I²C *plus*: (b) unlocked, (c) locked and authenticated, and (d) locked and not authenticated.

In order to lock or unlock the device, the user can press on the locker icon. A password selection menu will be shown, as shown in Fig. 41. If the device is unlocked, then the menu is to set a password. After setting it, the device is locked and authenticated. In case the device is locked, the menu is to authenticate the user with the correct password. If the user introduces the correct password, the device is unlocked. Fig 42. depicts a state diagram about the authentication procedure.

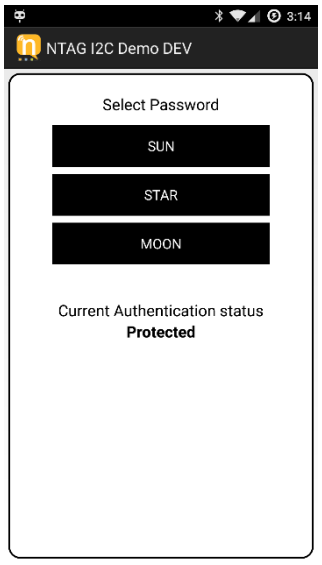

Fig 41. ***Password*** **snapshot**

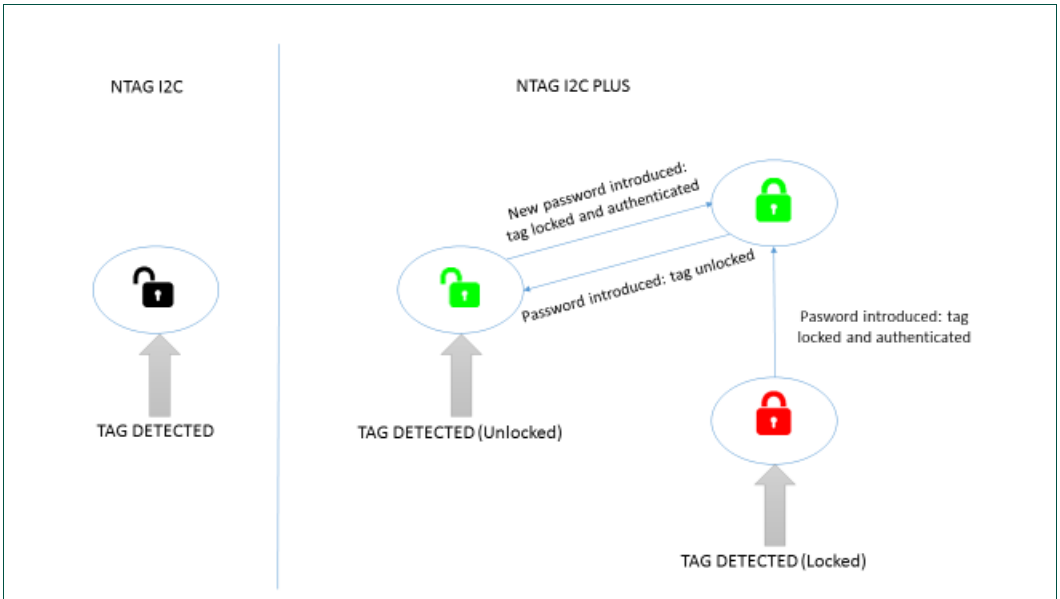

Fig 42. ***Authentication state diagram***

4.7 Bluetooth pairing functionality

This new feature is included in version 1.1 of the Android App. When working with a Kinetis KW41Z board, a Bluetooth pairing NDEF message can be automatically written in the NTAG *I²C plus* when pressing button 'SW3' on the board.

This Bluetooth pairing NDEF message is detected by the Android App when tapping the NTAG *I²C plus* board to the phone. Once the NDEF message has been read by the Android App, the NTAG *I²C plus* will delete it and write the default NDEF message.

A pop-up message is displayed on the phone to let the user decide, whether to proceed to the Bluetooth pairing or ignore it. Fig 43 shows the pop-up message displayed. If the user decides to proceed with the Bluetooth pairing, the Android System will pair both devices and launch 'Kinetis BLE Toolbox' application from Play Store.

When using this functionality, this functionality is not restricted to NTAG *I²C plus,* it works also with the previous NTAG I²C IC version.

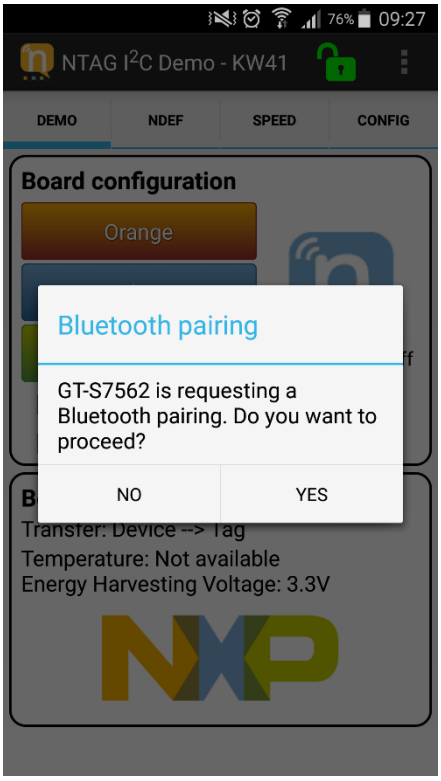

Fig 43. Bluetooth pairing pop-up message

5. Federal Communication Commission Interference Statement

5.1 FCC Grant

The NTAG I²C *plus* Explorer Kit with FCC ID OWROM5569-NT322E has been tested to fulfil the approval requirements ANSI C63.4-2009; FCC Part 15, Subpart C and has passed the tests.

T251-0895/15

Page: 16 (33)

6. TEST SUMMARY

STANDARDS (details on first page)	Tested yes	Tested no	Sample pass	Sample not pass
ANSI C63.4-2009; FCC Part 15, Subpart C	☑	☐	☑	☐

Test	Section within the report	Class	Conclusion
Conducted emission	3.1	/	N/A
Radiated emission	3.2	/	PASS

6.1 Operating voltages/frequencies used for testing

Section.	Test	Operating conditions
7.1	Conducted emission	/
7.2	Radiated emission	DC power supply connection from NTAG I²C tag ic

5.2 Integration condition

NTAG I²C *plus* Explorer Kit is not intended to be reused as a module for integration into end devices. If used, FCC certification needs to be repeated.

6. References

[NTAGI2Cplus] NT3H2111/NT3H2211, NTAG I²C *plus*, NFC Forum Type 2 Tag compliant IC with I²C interface
http://www.nxp.com/documents/data_sheet/NT3H2111_2211.pdf

[DEMOBOARD] Demo board home page with all resources
www.nxp.com/demoboard/OM5569

[EXPLORER] Schematics of the Explorer main board
http://www.nxp.com/documents/software/SW3638.zip

[Flex Antenna] Schematics of the Flex antenna
http://www.nxp.com/documents/software/SW3641.zip

[PCB Antenna] Schematics of the PCB antenna board
http://www.nxp.com/documents/software/SW3639.zip

[Field Detector] Schematics of the Field detector board
http://www.nxp.com/documents/software/SW3640.zip

[Identiv] Identiv Reader Driver
http://www.nxp.com/documents/software/SW3750.zip

[PC App] PC Application
http://www.nxp.com/documents/software/SW3651.zip

[Peek&Poke] Peek and Poke
http://www.nxp.com/documents/software/SW3652.zip

[LPC11U24] LPC11U24, 32-bit ARM Cortex-M0 microcontroller; up to 32 kB flash; up to 10 kB SRAM and 4 kB EEPROM; USB device; USART
http://www.nxp.com/documents/data_sheet/LPC11U2X.pdf

[LM75B] LM75B, Digital temperature sensor and thermal watchdog
http://www.nxp.com/documents/data_sheet/LM75B.pdf

[UM10989] NTAG I²C Demo Android app Developer start-up guide
www.nxp.com/documents/user_manual/UM10989.pdf

[UM10967] NTAG I²C *plus* Explorer Kit and Peek & Poke
www.nxp.com/documents/user_manual/UM10967.pdf

7. Legal information

7.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

7.2 Disclaimers

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

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

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

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

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

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

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

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

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

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

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

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

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

7.3 Licenses

Purchase of NXP ICs with NFC technology
Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. Purchase of NXP Semiconductors IC does not include a license to any NXP patent (or other IP right) covering combinations of those products with other products, whether hardware or software.

7.4 Trademarks

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

MIFARE — is a trademark of NXP Semiconductors N.V.

I²C-bus — logo is a trademark of NXP Semiconductors N.V

8. List of figures

9. Contents

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

Date of release: 09 March 2017 **360121**

Document identifier: UM10966