DEVICE-TO-DEVICE COMMUNICATION VIA NFC

WEBINAR SERIES:
HOW TO BUILD NFC APPLICATIONS

JORDI JOFRE
NFC READERS
NFC EVERYWHERE
07/03/2017
Agenda

• Device-to-device communication demo
  - Demo functionality
  - Hardware details
  - Application logic
    (How the CLRC663 plus and NTAG I²C plus are used)
  - MCU code details
    (for CLRC663 plus and NTAG I²C plus)
  - Available resources

• Wrap up and Q&A
DEMO FUNCTIONALITY
Device-to-device communication via NFC demo

Elements

- Reader module (14x12 cm)
- Rotating disk (8 cm diameter)
- Optional tablet display (BLE connection)
- Reader module & rotating sensor communicate via NFC
Device-to-device communication via NFC demo

Use cases

Demo use case:

- Once the reader module is supplied, all the electronics on the rotating disk are powered by harvesting energy from the reader module’s RF field (Green LED on).

NFC application

- NFC for communication with a batteryless unit. The reader’s NFC field can supply and operate low-power electronics.
Device-to-device communication via NFC demo

Use cases (II)

**NFC application**

- NFC for communication between 2 devices mounted in close vicinity that need to be completely isolated (e.g. dust proof)

**Demo use case 1:** Bidirectional exchange of static information

**Reader module → rotating disk communication:**
- Action **Button 1** is pressed → RGB LED on rotating disk turns **blue**.
- Action **Button 2** is pressed → RGB LED on rotating disk turns **red**.
- Action **Button 1 & 2** are pressed → RGB LED on rotating disk turns **white**.

**Demo use case 2:** Bidirectional exchange of static information

**Rotating disk → Reader module communication:**
- Action **Button 3** is pressed → LED pattern on LED circle will appear.
Device-to-device communication via NFC demo

Use cases (III)

- **Device-to-device communication via NFC**

**NFC application**

- NFC for communication with a rotating part as a cable replacement solution

**Demo use case 3:** *Bidirectional exchange of dynamic information*

- The rotating disk is able to harvest energy from the RF’s field of the reader module even if the disk is moving.

- The acceleration sensor measures x,y,z coordinates of the disk, the angle is calculated and the sensor position is displayed on the LED circle on the reader board.

- Android App: Action button data, rotation angle and temperature information is additionally sent via BLE to connected App.

**5V supply**
Device-to-device communication via NFC demo
Energy harvesting

Rotating disk does not include any battery

Energy is harvested as soon as the disk is close to the base board

Rotating disk electronics are supplied by the base board RF field
Device-to-device communication via NFC demo

Reader module ➔ rotating disk communication

While button 1 is pressed, the RGD LED on the rotating disk turns blue.

While button 2 is pressed, the RGD LED on the rotating disk turns red.

While button 1 and 2 are pressed, the RGD LED on the rotating disk turns white.
Device-to-device communication via NFC demo
Rotating disk → reader module communication

While button 3 is pressed, a pattern appears on the LED circle.

The LED circle lights up one LED accordingly to the coordinates sent by the accelerometer on the rotating part.
HARDWARE DETAILS
Reader module architecture
Board based on CLRC663 plus

* HW schematics are available
CLRC663 plus product highlights

Best performance at lowest power consumption
Extended LPCD (Low Power Card Detection) range with new configuration options.
Low supply voltage for battery support down to 2.5V

Design flexibility
Max. operating Tx current of 350mA with limiting value of 500mA.
Chip temperature range from -40 °C to 105°C

Backward compatibility
Pin-to-pin compatible to CLRC663

Faster time-to-market
Complete support package including EMVCo compliant NFC SW Library and NFC Cockpit for easy antenna configuration

* More CLRC663 plus info on product launch
Rotating disk architecture
Passive board based on NTAG I²C plus

* HW schematics are available
NTAG I\(^2\)C plus product highlights

The NTAG I\(^2\)C plus connects to the microcontroller via the I\(^2\)C serial bus interface.

NFC-enabled mobile device connects to the MCU via the NFC interface, using the I\(^2\)C as the communication conduit.

- Field Detection
- Energy harvesting
- SRAM memory
- SRAM mirroring
- Pass-through mode
- Memory access management
- Originality signature
How to use NTAG I²C *plus* SRAM for bidirectional data exchange

• The pass-through mode provides:
  - The SRAM for bidirectional data transfer from an NFC device to an I²C bus master
  - Triggering mechanisms for the synchronization of the data transfer.

• The data transfer signaling can be done though:
  - The Field Detection pin
  - Polling / reading specific NTAG I²C *plus* registers.

• NFC to I²C data transfer signaling:
  - The host polls `SRAM_I2C_READY` to learn if new data has been written by the RF interface in the SRAM.
  - A trigger on the FD pin (FD voltage goes from **HIGH** to **LOW**) indicates to the host that data is ready to be read from SRAM.

• I²C to NFC data transfer signaling:
  - The NFC interface polls `SRAM_RF_READY` to learn if new data has been written by the I²C interface in the SRAM.
  - A trigger on the FD pin (FD voltage goes from **LOW** to **HIGH**) indicates to the host that data has been read from SRAM by the NFC interface.

<table>
<thead>
<tr>
<th>Register bit</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PTHRU_ON_OFF</code></td>
<td>Detect if the pass-through mode is still enabled (gets reset in case of RF or I²C power down)</td>
</tr>
<tr>
<td><code>TRANSFER_DIR</code></td>
<td>Defines the data flow direction for the data transfer</td>
</tr>
<tr>
<td><code>I2C_LOCKED</code></td>
<td>Detect if memory access is currently locked to I²C</td>
</tr>
<tr>
<td><code>RF_LOCKED</code></td>
<td>Detect if Memory access is currently locked to RF</td>
</tr>
<tr>
<td><code>SRAM_I2C_READY</code></td>
<td>Detect if there is data available in the SRAM buffer to be fetched by I²C</td>
</tr>
<tr>
<td><code>SRAM_RF_READY</code></td>
<td>Detect if there is data available in the SRAM buffer to be fetched by RF</td>
</tr>
<tr>
<td><code>RF_FIELD_PRESENT</code></td>
<td>Shows if a RF field strong enough to read the tag is there</td>
</tr>
</tbody>
</table>

Important: Read / write operations to the termination page of the SRAM is what trigger register flags.

**AN11579** - How to use the NTAG® I²C and NTAG I²C *plus* for bidirectional communication
APPLICATION LOGIC
READER MODULE → ROTATING DISK COMMUNICATION
Reader module → rotating disk communication

Energy harvesting

<table>
<thead>
<tr>
<th>Page address (HEX)</th>
<th>Byte number within a page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td></td>
</tr>
<tr>
<td>0x01</td>
<td></td>
</tr>
<tr>
<td>0x02</td>
<td></td>
</tr>
<tr>
<td>0x03</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0x0F</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>0xFB</td>
<td></td>
</tr>
<tr>
<td>0xFD</td>
<td></td>
</tr>
<tr>
<td>0xFE</td>
<td></td>
</tr>
<tr>
<td>0xFF</td>
<td></td>
</tr>
</tbody>
</table>

NTAG I²C plus energy harvesting pin (Vout)

LPCU124

I²C

GPI0 16 - red
GPI0 17 - green
GPI0 18 - blue

1. Once base board is powered, it starts generating RF field
2. MCU powers up harvesting energy from the RF field
3. MCU changes GPIO 17 to HIGH level
4. LED turns on green

Rotating disk

*Rotating disk electronics are supplied by the base board RF field
Reader module → rotating disk communication

Action button 1

1. MCU GPIO 4 is in HIGH level

2. Reader writes SRAM page address 0xFF

3. MCU reads config from SRAM

4. MCU changes GPIO 18 to HIGH level

5. LED turns on blue

Data flow direction (SRAM in pass through mode)

Fig. Simplified NTAG I2C plus memory map (NFC interface perspective)

- SRAM mirror memory area (volatile memory)
- EEPROM memory area (non-volatile memory)

*Rotating disk electronics are supplied by the base board RF field
Reader module → rotating disk communication

Action button 2

1. MCU GPIO 5 is in HIGH level

2. Reader writes SRAM page address 0xFF

3. MCU reads config from SRAM

4. MCU changes GPIO 16 to HIGH level

5. LED turns on red

Data flow direction (SRAM in pass through mode)

SRAM mirror memory area (volatile memory)

EEPROM memory area (non-volatile memory)

*Rotating disk electronics are supplied by the base board RF field
Reader module → rotating disk communication
Action button 1&2

1. MCU GPIO 4, 5 are in HIGH level

2. Reader writes SRAM page address 0xFF

3. MCU reads config from SRAM

4. MCU changes GPIO 16, 17, 18 to HIGH level

5. LED turns on white

---

**Important**: It is what is written in **SRAM** what **controls** the behavior of the **disk RGB LED**, demonstrating **data exchange** between the two embedded systems.

---

![Device-to-device communication via NFC](image)

**LPC11U68**

**CLRC663 plus**

**GPIO 16 - red**

**GPIO 17 - green**

**GPIO 18 - blue**

---

**SRAM mirror memory area (volatile memory)**

**EEPROM memory area (non-volatile memory)**

---

Fig. Simplified NTAG I²C plus memory map (NFC interface perspective)
APPLICATION LOGIC
ROTATING DISK → BASE BOARD COMMUNICATION
Rotating disk → Reader module communication

Accelerometer (I)

1. MCU reads temperature (T1,T2) value and Cx, Cy, Cz axis from accelerometer
2. MCU writes into SRAM
3. Reader reads SRAM
4. MCU calculates angle and sets to HIGH level the appropriate GPIO (steps of 30 degrees)

Fig. Simplified NTAG I2C plus memory map (NFC interface perspective)

<table>
<thead>
<tr>
<th>Page address (HEX)</th>
<th>Byte number within a page</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>0</td>
</tr>
<tr>
<td>0x01</td>
<td>1</td>
</tr>
<tr>
<td>0x02</td>
<td>2</td>
</tr>
<tr>
<td>0x03</td>
<td>3</td>
</tr>
<tr>
<td>0x04</td>
<td>4</td>
</tr>
<tr>
<td>0x05</td>
<td>5</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>0xFB</td>
<td>T1</td>
</tr>
<tr>
<td>0xFC</td>
<td>T2</td>
</tr>
<tr>
<td>0xFD</td>
<td>Cz</td>
</tr>
<tr>
<td>0xFE</td>
<td>Cy</td>
</tr>
</tbody>
</table>

Data flow direction (SRAM in pass through mode)

SRAM mirror memory area (volatile memory)
EEPROM memory area (non-volatile memory)

Training Mobile Knowledge

*Rotating disk electronics are supplied by the base board RF field
Rotating disk \rightarrow Reader module communication

Accelerometer (II)

1. Disk rotates 90°

Data flow direction (SRAM in pass through mode)

Important: The LED circle operation is directly controlled by the disk position coordinates sent via the SRAM!

SRAM mirror memory area (volatile memory)
EEPROM memory area (non-volatile memory)

Page address (HEX) | Byte number within a page
--- | ---
0x00 | 0 1 2 3
0x01
0x02
0x03
0x04
0x05
...
0xFD
...
0xFB
0xFC
0xFD
...
0xFF

Fig. Simplified NTAG I\(^2\)C plus memory map (NFC interface perspective)

1. Disk rotates 90°
2. MCU reads temperature value and the updated \( C_x, C_y, C_z \) coordinates from accelerometer
3. MCU writes into SRAM
4. Reader reads SRAM
5. MCU calculates angle and sets to HIGH level the appropriate GPIO (steps of 30 degrees)

*Rotating disk electronics are supplied by the base board RF field

Important: The LED circle operation is directly controlled by the disk position coordinates sent via the SRAM!
Rotating disk ➔ Reader module communication

Action button 3

1. While user presses action button 3, the LPC11U24 GPIO 12 is in HIGH level.

2. MCU writes into SRAM

3. Reader reads SRAM

4. MCU set to HIGH level the 12 GPIOs controlling the 12 LEDs.

---

**Important:** The LED circle pattern is triggered by a specific data byte sent via the SRAM!

---

*Rotating disk electronics are supplied by the base board RF field*
MCU CODE DETAILS
READER MODULE (CLRC663 PLUS)
Reader module MCU code

**FreeRTOS Library**

FreeRTOS is an open source real-time operating system (RTOS) for embedded systems supporting many different architectures and compiler toolchains.

**Lpc_chip_11u6x_lib & nxp_lpcxpresso_11u68b**

LPCOpen software libraries (drivers and middleware) supporting development on top of LPC MCU solutions. These two libraries bring support to LPC11U68 LPCXpresso board.

**NTAG_Device2DeviceDemo**

Implements the logic supporting the device-to-device communication demo for the reader module. More on the next slides.

**NxpNfcRdLib**

The NXP’s software stack for creating and developing contactless applications. It is written in C language, it is based on modular multi-layer software architecture, and can support multiple design environments and platforms.

*Source code is available*
Reader module MCU code leverages on the NFC Reader Library

1. Initialize the OSAL implementation of FreeRTOS component
2. Initialize the host controller interface
3. Initialize the NFC reader IC (CLRC663)
4. Initialize the RF protocols (ISO14443-3, MIFARE)
5. Initialize discovery loop for card detection
6. Initialize the NTAG I²C plus command set
7. We re-use an existing sample code

*NFC Reader Library components are initialized following a bottom to top approach*

**NTAG_Device2DeviceDemo** application workflow

1. **Power on**
2. Init GPIOs controlling LEDs
3. UART init
4. Initialize NFC Reader Library components
5. Create thread for rotating disk operation
6. Discovery loop configuration
7. Start discovery loop

**If (SAK=0x00 and ATQA=0x44) → NTAG IC plus tag detected**

- Read PTHROUGH_ON_OFF register bit until is set
- Wait until pass through mode is enabled
- Read TRANSFER_DIR register bit until is set to 1
- Wait until RF to FC direction is set
- Get action buttons
- Write SRAM
- Wait until I2C has written in SRAM
- Read SRAM
- Calculate angle from accelerometer axis coordinates
- Display angle
- Activate the corresponding LED depending on the calculated angle
- Trigger LED pattern on the LED circle
- Send data to tablet via BT

**Button 3 pressed?**

- Yes
  - Iterates here while the reader module is powered

**Detection of a vicinity or other type A tag**
MCU CODE DETAILS
ROTATING DISK (NTAG I²C PLUS)
Rotating disk MCU code

**Lpc_chip_11uxx_lib & nxp_lpcxpresso_11u24h_board_lib**

LPCOpen software libraries (drivers and middleware) supporting development on top of LPC MCU solutions. These two libraries bring support to LPC11U24 LPCXpresso board.

**NTAG_I2C_API**

Implements the NTAG I²C plus command set and offers an API to developers to communicate with NTAG I²C plus from the I²C interface.

Memory operations (I²C side)

- `NTAG_ReadBytes (NTAG_HANDLE_T ntag, uint16_t address, uint8_t *bytes, uint16_t len);`
- `NTAG_WriteBytes(NTAG_HANDLE_T ntag, uint16_t address, const uint8_t *bytes, uint16_t len);`

Register operations

- `NTAG_ReadRegister (NTAG_HANDLE_T ntag, uint8_t reg, uint8_t *val1);`
- `NTAG_WriteRegister(NTAG_HANDLE_T ntag, uint8_t reg, uint8_t val, uint8_t mask);`

Setting SRAM for pass-through mode operation

- `NTAG_SetPthruOnOff(NTAG_HANDLE_T ntag, BOOL on)`
- `NTAG_SetTransferDir(NTAG_HANDLE_T ntag, NTAG_TRANSFER_DIR_T dir)`

**NTAG_I2C_Explorer_01_LEDs_Button_Xample**

Implements the logic supporting the device-to-device communication demo. More on the next slide.

* Source code is available
**NTAG_I2C_Explorer_01_LEDs_Button_Xample** application workflow

1. **Power on by energy harvesting**
2. **Enable NTAG I2C plus pass through mode**
3. **While (1)**
   - **Set RF to I2C direction**
   - **Wait until RF has written in SRAM**
   - **Read SRAM 0xFF page address**
   - **Check last byte (btn)**
     - While button 1 is pressed, btn=0x01
     - While button 2 is pressed, btn=0x02
     - While action button 1&2 are pressed, btn=0x03
   - **Turn disk LED to blue**
   - **Turn disk LED to red**
   - **Turn disk LED to white**
4. **Check if disk action button is pressed**
5. **Read Temperature sensor**
6. **Read accelerometer axis coordinates**
7. **Set FC to RF direction**
8. **Write SRAM**
9. **Wait until RF has read SRAM**

**Register Bits**
- **PTHROUGH_ON_OFF** register bit is set
- **TRANSFER_DIR** register bit is set to 1
- **SRAM_I2C_READY** register bit is set
- **SRAM_RF_READY** register bit is cleared
- **SRAM_I2C READY** register bit is set to 0
- **TRANSFER_DIR** register bit is set to 0
- **SRAM_RF READY** register bit is cleared

**Iterates here while RF field is present**
EVERYTHING YOU NEED TO BUILD YOUR DEVICE-TO-DEVICE COMMUNICATION VIA NFC SOLUTION IS HERE!
Summary of available resources

• More info about our NFC products:
  - CLRC663
  - NTAG I²C plus

• Check our post on the DemoLab section in our NXP community for:
  - Demo description
  - Source code
  - Schematics
  - Video recording
  https://community.nxp.com/docs/DOC-333917
Software development in Android and iOS
Embedded software for MCUs
JCOP, Java Card operating Systems
Hardware design and development
Digital, analog, sensor acquisition, power management
Wireless communications WiFi, ZigBee, Bluetooth, BLE
Contactless antenna RF design, evaluation and testing

MIFARE applications
End-to-end systems, readers and card-related designs
EMVco applications
Readers, cards, design for test compliancy (including PCI)
Secure Element management
GlobalPlatform compliant backend solutions
Secure services provisioning OTA, TSM services

MobileKnowledge
Roc Boronat 117, P3M3
08018 Barcelona (Spain)

Get in touch with us
www.themobileknowledge.com
mk@themobileknowledge.com
Device-to-device communication via NFC
Jordi Jofre (Speaker)
Angela Gemio (Host)

Thank you for your kind attention!

Please remember to fill out our evaluation survey (pop-up)

Check your email for material download and on-demand video addresses

Please check NXP and MobileKnowledge websites for upcoming webinars and training sessions

http://www.nxp.com/support/classroom-training-events;CLASSROOM-TRAINING-EVENTS
www.themobileknowledge.com/content/knowledge-catalog-0