NFC Lab

Michel Simatic

15/01/2015
Table of contents

- RFID versus NFC

- High level interactions with tags
  - Touchatag (Tikitag) / Mir:ror
  - Smart posters

- Low level interactions with tags
  - Card readers
  - Tags

- Medium level interactions with tags
  - JSR 257: Contactless Communication API
  - Android and NFC

- Conclusion and future work
RFID versus NFC

High level interactions with tags
  • Touchatag (Tikitag) / Karotz
  • Smart posters

Low level interactions with tags
  • Card readers
  • Tags

Medium level interactions with tags
  • JSR 257: Contactless Communication API
  • Android and NFC

Conclusion and future work
## RFID versus NFC [Auto-ID labs, 2008]
### Overview of standards

<table>
<thead>
<tr>
<th>ISO/IEC Standard</th>
<th>UHF EPC Gen 2 ISO 18000-6C</th>
<th>ISO 21481&lt;sup&gt;1&lt;/sup&gt; (NFCIP-2/ECMA-352)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ISO18092</td>
<td>ISO 14443 A/B</td>
</tr>
<tr>
<td>Frequency</td>
<td>UHF</td>
<td>HF</td>
<td>HF</td>
</tr>
<tr>
<td>Principle of coupling</td>
<td>Electro-Magnetic</td>
<td>Inductive</td>
<td>Inductive</td>
</tr>
<tr>
<td>Read Range</td>
<td>7m</td>
<td>4-5cm</td>
<td>4-5cm</td>
</tr>
<tr>
<td>Access operations</td>
<td>Read/write</td>
<td>Read/write</td>
<td>Read/write</td>
</tr>
<tr>
<td>Data transfer rate</td>
<td>up to 640 kb/s (passive communication)</td>
<td>up to 400kb/s (passive communication)</td>
<td>106kb/s</td>
</tr>
<tr>
<td>Kill-command</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Identifier</td>
<td>EPC code variable in length, up to 96bit</td>
<td>4, 7, or 10 bytes random</td>
<td>4 (Type, 7, or 10 bytes random or fixed)</td>
</tr>
<tr>
<td>Tag data memory</td>
<td>8kb</td>
<td>1 kb lockable for read-only</td>
<td>96 bytes lockable for read-only</td>
</tr>
</tbody>
</table>
### RFID versus NFC [Auto-ID labs, 2008]

**Benefits of EPC technology and NFC technology**

<table>
<thead>
<tr>
<th>NFC Use</th>
<th>RFID/EPC Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry Publications</td>
<td>Consumer Use</td>
</tr>
<tr>
<td>Mobile Payment</td>
<td>1</td>
</tr>
<tr>
<td>Mobile Ticketing</td>
<td>x</td>
</tr>
<tr>
<td>Physical Access Control</td>
<td>x</td>
</tr>
<tr>
<td>Logical Access Control</td>
<td>x</td>
</tr>
<tr>
<td>Contactless Loyalty Cards</td>
<td>x</td>
</tr>
<tr>
<td>Health Care File Storage</td>
<td>x</td>
</tr>
<tr>
<td>Car Ignition Key Storage</td>
<td>x</td>
</tr>
<tr>
<td>Electronic Redeem Coupons</td>
<td>x</td>
</tr>
<tr>
<td>Data Transfer</td>
<td>x</td>
</tr>
<tr>
<td>Easy Device Paring</td>
<td>x</td>
</tr>
<tr>
<td>Product Information</td>
<td>x</td>
</tr>
<tr>
<td>Field Force Solution</td>
<td>x</td>
</tr>
<tr>
<td>Support of Children and Elders</td>
<td>x</td>
</tr>
<tr>
<td>Information Download</td>
<td>x</td>
</tr>
<tr>
<td>Improved Product Quality and Security</td>
<td>x</td>
</tr>
<tr>
<td>Track &amp; Tracing</td>
<td>x</td>
</tr>
<tr>
<td>More Accurate Inventories</td>
<td>x</td>
</tr>
<tr>
<td>Product Lifecycle Management</td>
<td>x</td>
</tr>
<tr>
<td>Rapid Check Out at POS</td>
<td>x</td>
</tr>
<tr>
<td>Automated Receiving</td>
<td>x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NFC Use</th>
<th>RFID/EPC Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Industry Publications</td>
<td>Consumer Use</td>
</tr>
<tr>
<td>Pick Lists Replenishment</td>
<td>x</td>
</tr>
<tr>
<td>Better Service Quality (Accurate Deliveries)</td>
<td>x</td>
</tr>
<tr>
<td>Asset Management</td>
<td>x</td>
</tr>
<tr>
<td>Promotion &amp; Event Execution</td>
<td>x</td>
</tr>
<tr>
<td>Recovery of stolen items</td>
<td>x</td>
</tr>
<tr>
<td>Lower Consumer Prices</td>
<td>x</td>
</tr>
<tr>
<td>Reduced unsaleables</td>
<td>x</td>
</tr>
</tbody>
</table>

---

Zoom on NFC: The NFC protocol stack

**[NFC, 2014]**

- User Interface
- Record Format
- Message Format
- Command Protocols
- Data Packetization Specification
- Radio Specification (13.56MHz)
- Radio Controllers
- Device-to-Device Communication
- PC, Embedded System, Microcontroller

---

**ISO-14443-3**

- FeliCa
- DESFire
- MiFare Ultralight
- MiFare Classic

**ISO-14443-2**

- Hardware controllers (PN532, SCL3711, etc.)
  - UART
  - SPI
  - I2C
  - USB
  - CPU

---

**ISO-18092**

- Simple NDEF Exchange Protocol (SNEP)
- Logical Link Control Protocol (LLCP)
There are four types of tags defined by the NFC forum. There’s a fifth that’s compatible, but not strictly part of the NFC specification.

**Type 1**
- Based on ISO-14443A specification.
- Can be read-only, or read/write capable.
- 96 bytes to 2 kilobytes of memory.
- Communication speed 106Kb.
- No data collision protection.
- Examples: Innovision Topaz, Broadcom BCM20203.
5 NFC tag types [NFC, 2014]

- **Type 2**
  - Similar to type 1 tags, type 2 tags are based on NXP/Philips Mifare Ultralight tag (ISO-14443A) specification.
  - Can be read-only, or read/write capable.
  - 96 bytes to 2 kilobytes of memory.
  - Communication speed 106Kb.
  - Anti-collision support.
  - Example: NXP Mifare Ultralight.
5 NFC tag types [NFC, 2014]

- Type 3
  - These are based on the Sony FeliCa tags (ISO-18092 and JIS-X-6319-4), without the encryption and authentication support that FeliCa affords.
  - Configured by factory to be read-only, or read/write capable.
  - Variable memory, up to 1MB per exchange.
  - Two communication speeds, 212 or 424Kbps.
  - Anti-collision support.
  - Example: Sony FeliCa.
5 NFC tag types [NFC, 2014]

- **Type 4**
  - Similar to type 1 tags, type 4 tags are based on NXP DESFire tag (ISO-14443A) specification.
  - Configured by factory to be read-only, or read/write capable.
  - 2, 4, or 8KB of memory.
  - Variable memory, up to 32KB per exchange.
  - Three communication speeds: 106, 212, or 424Kbps.
  - Anti-collision support.
  - Example: NXP DESFire, SmartMX-JCOP.
5 NFC tag types [NFC, 2014]

- Mifare Classic tag (ISO–14443A), proprietary to NXP Semiconductors ➔ Not readable by readers not equipped with NXP chip, i.e. most of Android hardware (broadband-based)
  - Memory options: 192, 768, or 3,584 bytes.
  - Communication speed 106Kbps
  - Anti-collision support.
  - Examples: NXP Mifare Classic 1K, Mifare Classic 4K, Mifare Classic Mini.
Table of contents

- RFID versus NFC

- High level interactions with tags
  - Touchatag (Tikitag) / Karotz
  - Smart posters

- Low level interactions with tags
  - Card readers
  - Tags

- Medium level interactions with tags
  - JSR 257: Contactless Communication API
  - Android and NFC

- Conclusion and future work
Touchatag (Tikitag) / Karotz

Generalities

- Link real objects to actions in virtual world [Touchatag, 2010] (Same functionalities as the Karotz [Karotz, 2012])
- Principles

1. The tag contains a unique identifier.
2. The end user touches a tag to his or her reader. The reader reads the unique identifier. Then the Touchatag application calls on the ACS to correlate a tag action.
3. The ACS determines that the tag is associated to a Web Link Application, and sends back the URL to the end user's system.
4. The Touchatag application triggers the opening of the URL sent by the ACS, by a local browser.

Works also with
- An RFID-enabled mobile phone
- A device able to read QR-Code
Touchatag (Tikitag) / Karotz

Analysis

- Programming model is very limited
- Open questions
  - How to configure rapidly a list of tags?
  - How to take into account parameters present on tag, on reader?
  - Business model? (http://www.iotope.com/)
The Smart Poster is one of the key use cases for NFC technology [Smart Poster, 2006]

The idea is that an object can be made “smart”, i.e., it is capable of storing additional information about itself in the form of an NFC Forum Tag. By touching an NFC Forum Device to the tag, this information can be read and displayed to the user.

The Smart Poster can also contain actions that will trigger an application in the device; for example, launching a browser to view a web site, or sending an SMS to a premium service to receive a ring tone.

The Smart Poster concept is built around URIs (Uniform Resource Identifiers [RFC 3986])
Smart posters

Analysis

- Some interesting applications (e.g. paintings discovery) [Haberman et al., 2009], but somehow limited

- Competition with camera-based applications
  - QR-codes
  - SmartPaper technology ([http://www.smartpaper-app.com](http://www.smartpaper-app.com))
  - DirectFlash (Journal Direct Matin, [http://www.direct8.fr/directflash](http://www.direct8.fr/directflash))
  - U-snap (J-C Decaux, [http://www.u-snap.net/](http://www.u-snap.net/))
Table of contents

- RFID versus NFC

- High level interactions with tags
  - Touchatag (Tikitag) / Karotz
  - Smart posters

- Low level interactions with tags
  - Card readers
  - Tags

- Medium level interactions with tags
  - JSR 257: Contactless Communication API
  - Android and NFC

- Conclusion and future work
HW/SW requirement to have low level interactions

- A tag reader

- The tag reader driver

- A library to interface with the driver:
  - Libusb
  - Personal computer/Smart Card (or PC/SC) software library
Tag example: Mifare [Mifare, 2008]

- 13,56 MHz
- 106 kbit/s
- Typical transaction in less than 100 milliseconds
- Data retention: 10 years
- Write endurance: 100,000 cycles
- Anticollision loop
Tag example: Mifare
Structure of a 1 kB tag

- General structure

<table>
<thead>
<tr>
<th>Sector</th>
<th>Block</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>3</td>
<td>Sector Trailer 15</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Data</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>Sector Trailer 14</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Data</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>Sector Trailer 0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Data</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td>Data</td>
</tr>
</tbody>
</table>

- Block 0 (sector 0) contains manufacturer data

- NB: Net storage capacity = 1024–16–16*16 = 752 bytes
Tag example: Mifare
Available operations

- Read block
- Write block
- Decrement: Decrements the contents of a block and stores the result in a temporary internal data-register
- Increment: Increments the contents of a block and stores the result in the data-register
- Restore: Moves the contents of a block into the data-register
- Transfer: Writes the contents of the temporary internal data-register to a value block
Tag example: Mifare Security

- Three pass authentication sequence
  1. The reader specifies the sector to be accessed and chooses key A or B.
  2. The card reads the secret key and the access conditions from the sector trailer. Then the card sends a random number as the challenge to the reader (pass one).
  3. The reader calculates the response using the secret key and additional input. The response, together with a random challenge from the reader, is then transmitted to the card (pass two).
  4. The card verifies the response of the reader by comparing it with its own challenge and then it calculates the response to the challenge and transmits it (pass three).
  5. The reader verifies the response of the card by comparing it to its own challenge.

- After transmission of the first random challenge the communication between card and reader is encrypted.

- Bad news: Security algorithms have been broken
  - Hack demo: [http://fr.youtube.com/watch?v=NW3RGbQTLhE](http://fr.youtube.com/watch?v=NW3RGbQTLhE)
  - NXP reaction: [http://www.mifare.net/technology/security/](http://www.mifare.net/technology/security/) (see the FAQ of this Web page)
  - MFOC: [https://code.google.com/p/mfoc/](https://code.google.com/p/mfoc/)
Interface between Card reader and Mifare tag

[ACS NFC reader, 2008]

- Send/receive APDUs to/from the reader
  Example = Read Binary Blocks

The “Read Binary Blocks command” is used for retrieving a “data blocks” from the PICC. The data block/trailer block must be authenticated first.

***Table 4-a: Read Binary APDU Format (5 Bytes)***

<table>
<thead>
<tr>
<th>Command</th>
<th>Class</th>
<th>INS</th>
<th>P1</th>
<th>P2</th>
<th>Le</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read Binary Blocks</td>
<td>FF</td>
<td>80</td>
<td>00</td>
<td>Block Number</td>
<td>Number of Bytes to Read</td>
</tr>
</tbody>
</table>

where:
- Block Number (1 Byte): The block to be accessed
- Number of Bytes to Read (1 Byte): Maximum 16 bytes

***Table 4-b: Read Binary Block Response Format (N + 2 Bytes)***

<table>
<thead>
<tr>
<th>Response</th>
<th>Data Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result</td>
<td>0 &lt;= N &lt;= 16, SW1, SW2</td>
</tr>
</tbody>
</table>

***Table 4-c: Read Binary Block Response Codes***

<table>
<thead>
<tr>
<th>Results</th>
<th>SW1</th>
<th>SW2</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Success</td>
<td>90</td>
<td>00</td>
<td>The operation completed successfully.</td>
</tr>
<tr>
<td>Error</td>
<td>63</td>
<td>00</td>
<td>The operation failed.</td>
</tr>
</tbody>
</table>

Example:

1. Read 16 bytes from the binary block 0x04 (MIFARE 1K or 4K) APDU = {FF, 80 00 04 10}

- Use libraries like [http://www.libnfc.org/](http://www.libnfc.org/), a public platform independent Near Field Communication (NFC) library
Desfire

- Memory organization = flexible system file
  - Up to 28 applications simultaneously
  - Up to 16 files in each application

[To be continued, once http://fr.slideshare.net/ashu4india/mifare-des-fire-pres is mastered]
**Data structure in a tag**

**Unique identifier of a tag**
- Notion of UID
  - 4 bytes on Mifare 1K or 4K
  - 6 bytes on Mifare Ultralight
- Electronic Product Code (EPC) [EPC tag standard, 2008]
  - General Identifier (GID-96)
  - But there are others

**NFC Data Exchange Format (NDEF)**
- [NDEF, 2006] defines a message encapsulation format to exchange information
- NDEF is a lightweight, binary message format that can be used to encapsulate one or more application-defined payloads of arbitrary type and size into a single message construct. Each payload (stored in an NDEFRecord) is described by a type, a length, and an optional identifier.
- Type identifiers may be URIs, MIME media types…
NFC Peer-to-Peer communications

[To be continued]
Table of contents

- RFID versus NFC

- High level interactions with tags
  - Touchatag (Tikitag) / Karotz
  - Smart posters

- Low level interactions with tags
  - Card readers
  - Tags

- Medium level interactions with tags
  - JSR 257: Contactless Communication API
  - Android and NFC

- Conclusion and future work
Listeners
- When a tag is detected
- When a NDEFRecord (containing a given type) is detected

Each NDEFRecord has a payload stored as a ByteArray
- This ByteArray is produced by using a ByteArrayOutputStream
- To produce this ByteArray in a more user-friendly manner, one can use Struct class of Javolution middleware (http://javolution.org/)
[To be continued, once http://developer.android.com/guide/topics/connectivity/nfc/nfc.html is mastered]
Table of contents

- RFID versus NFC

- High level interactions with tags
  - Touchatag (Tikitag) / Karotz
  - Smart posters

- Low level interactions with tags
  - Card readers
  - Tags

- Medium level interactions with tags
  - JSR 257: Contactless Communication API
  - Android and NFC

- Conclusion and future work
Other possibilities to explore

- More info on Desfire
- More info on Android
- Host Card Emulation
- Take a look at RFIDIOt (http://rfidiot.org), an open source python library for exploring RFID devices, but not stable according to [OpenSilicium, 2014]
- OW2 Aspire RFID (http://wiki.aspire.ow2.org): an RFID suite for SMEs + courses about RFID
Thank you for your attention

- Questions?
References

- [Smart Poster, 2006] Smart Poster Record Type Definition, Technical specification SPR 1.1, NFC Forum, 2006