UDC 004.057.7

USING SAMSUNG SMART WATCHES AS AN NFC PASS

A. KARNILOVICH, D. PASTUHOV
Polotsk State University, Belarus

This article talks about NFC technology and how to create a Samsung smart watch application to emulate an access card.

Introduction. Near field communication (NFC) is a form of contactless communication between devices or between a device and a chip/tag. Using NFC, a user can transfer/receive small amount information from a short distance (example, 10 cm).

Card-emulation mode, as the name suggests, makes the device behave like a contactless smart card. Using this mode, we can develop virtual credit cards, debit cards, transit cards, and access cards.

Near-Field Communication is a method that devices have for communicating with each other when in close proximity to one another.

The technology originated as an offshoot of RFID tech, or radio frequency identification, with the added rule that the devices had to be close to each other physically. Both techs rely on the physics of electromagnetic radio fields to transmit parcels of data. The term "radio wave" is used to denote a certain swath of wavelengths of light, and it's used in most modern technologies as a means of communication because it can easily go through walls and generally not be hampered by physical obstacles (as opposed to, say, visible light).

How do NFC readers work?

The door reader activates and transmits radio waves to induce electric current in the NFC card; passive devices do not actually need any power source, but instead are designed to activate and begin transmitting their NFC signal when exposed to a changing magnetic field. The active devices will have an electric current running through them and, when they are put in close proximity to a passive device, they will prompt it to begin transmitting—read the signal—then the passive device will stop transmitting when it's moved farther away. Here is a diagram explaining how the induced current works. The active device (right: "Reader") has its magnetic field interact with the passive device (left: "Smartphone") and creates a matching current to power it.

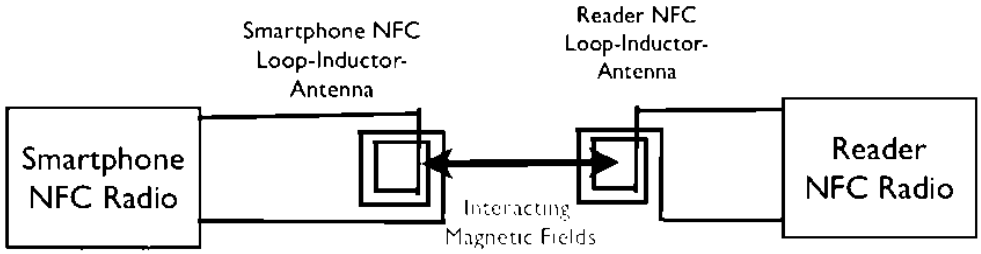

Figure 1. – NFC communication scheme

Different Types of NFC ID

The NFC reader induces a magnetic current in the token and, in turn, the token transmits the relevant string of characters. Depending on what type of token it is, the nature of this string will change: If it's a simple ID reader, then the string is a unique identifier for the user; if it's a transit card then it's an ID, but on the back end the reader needs to record the charge and update its server, and if it's a contactless payment card, then it's an ID with many added layers of encryption and security to protect the user's bank information (in this case, only a dedicated and approved reader could decrypt the information relevant to the bank details, any other reader would only be able to read a simple ID string).

Signal Relaying Back to the Server

After the reader reads the ID, based on the use case, the reader will then communicate this information back to the host server over local network or WiFi. Most readers only cache a small amount of information and have a method of communicating with the primary server—since storing all the relevant user info on a reader

would be impractical, insecure, and would result in too bulky a product. The server will then authenticate the request and prompt the reader to perform some relevant action, be it unlocking a door, communicating with a bank server, unlocking a subway turnstile, etc. This all happens behind-the-scenes and instantaneously (depending on the strength of the WiFi signal) guarantees the end user a seamless experience.

Main section. All current models of Samsung smart watches work on the Tizen operating system. Tizen is a user-interactive and service-oriented open source project that allows creating feature-rich applications for multiple device categories.

The Tizen Near Field Communication service enables information exchange between NFC-enabled devices (called "peers") or tags. The NFC-enabled devices can share contacts, photos, and videos, and can act as smart cards. You can use an NFC-enabled device to send NDEF messages to NFC tags to implement a variety of activities, such as paying the grocery bill or downloading a coupon. With application controls, you can launch NFC applications when NFC-related operations occur.

The main features of the NFC API include:

– NFC device management: you can manage NFC connectivity by enabling or disabling the NFC service.

– NFC tag and peer detection: you can receive notifications when an NFC tag or peer device has been detected.

– NDEF message manipulation: you can handle NDEF messages by first creating NDEF records, and then adding the records to an NDEF message.

– NDEF data exchange: you can exchange NDEF data between tags and peers.

– NFC card emulation: you can enable NFC card emulation and monitor the secure element transaction carried out by the device.

– NFC host-based card emulation (HCE): you can handle HCE events and transactions.

An NFC tag is a chip, which can securely store personal information, such as debit card numbers or contact details. The methods of the NFCTag interface (in mobile and wearable applications) are used to access an NFC tag for reading or writing information. Tizen supports the following NFC tag types: GENERIC_TARGET, ISO14443_A, ISO14443_4A, ISO14443_3A, MIFARE_MINI, MIFARE_1K, MIFARE_4K, MIFARE_ULTRA, MIFARE_DESFIRE, ISO14443_B, ISO14443_4B, ISO14443_BPRIME, FELICA, JEWEL and ISO15693.

The NFC forum defines the NFC data exchange format (NDEF) for encapsulating the data exchanged between two NFC-enabled devices or an NFC-enabled device and an NFC tag. An NDEF message can store data in various formats, such as text, Multipurpose Internet Mail Extension (MIME) type object, or ultra-short RagTime Document (RTD). The NFC tags use NDEF for exchanging messages. Tizen provides the NDEFMessage interface (in mobile and wearable applications) to define an NDEF message.

An NDEF message is composed of multiple records. An NDEF record is created using the NDEFRecord interface and is identified by record type, ID, and payload.

A record in an NDEF message can be created by using the following payload types:

– Text. The NDEF record content is created using text format. The *NDEFRecordText* interface is used to create the text format payload using the text, languageCode, and encoding attributes.

– URI. The NDEF record content is created using a URI. The *NDEFRecordURI* interface is used to create the URI type payload using the uri attribute.

– Media. The NDEF record content is created using a media format. The *NDEFRecordMedia* interface is used to create the media format payload using the mimeType attribute.

Managing NFC Connectivity

To use the Application and NFC APIs, the application has to request permission by adding the privileges to the config.xml file. To use NFC, retrieve the default NFC adapter using the *getDefaultAdapter*() method of the *NFCAdapter* interface.

To enable or disable the NFC service:

1. To get the default NFC adapter, use the *getDefaultAdapter*() method and prepare an *ApplicationControl* object to request the NFC switching operation.
2. Define the event listener for the *launchAppControl*() method.
3. Define the event handler for an application control, which implements the *ApplicationControlDataArrayReplyCallback* interface.
4. If necessary, request launching the NFC Settings with *nfcSwitchAppControl* as a parameter.

Using NFC Card Emulation

You can enable NFC card emulation and monitor the secure element transaction-taking place using the *NFCAdapter* interface. The device carries out the secure element transaction. The Tizen application can detect

the transaction, but does not take part in it. Interpreting the transaction data requires knowledge about the data protocol the transaction uses. With the required knowledge, the application can detect whether the transaction was successful.

To enable or disable the NFC card emulation and detect secure element transactions:

1. Declare the required variables and obtain the *NFCAdapter* object using the *getDefaultAdapter*() method of the *NFCManager* interface.

2. To enable NFC card emulation, change the value of the *cardEmulationMode* attribute to *'ALWAYS_ON'.*

3. To be notified when the type of an active NFC secure element changes, use the *addActiveSecureElementChangeListener*() method of the *NFCAdapter* interface.

4. To be notified when a NFC secure element transaction data is exchanged, use the *addTransactionEventListener*() method of the *NFCAdapter* interface.

5. Remove the registered listeners when they are no longer necessary and disable NFC card emulation

Using NFC Host-based Card Emulation

You can handle HCE (host-based card emulation) events and transactions taking place using the NFCAdapter interface. HCE is an on-device technology that permits a device to perform card emulation on an NFC-enabled device without relying on access to a secure element. The transaction data is routed to the host application directly, instead of routing to a secure element. The Tizen application can detect the transaction and can take part in it.

To detect NFC HCE events and manage AID (Application ID):

1. Specify an *AID* value for receiving HCE transaction events. To tell the platform which AID groups are requested by the application, a metadata element must be included in the *config.xml* file.

2. Declare the required variables and obtain the NFCAdapter object using the *getDefaultAdapter*() method of the *NFCManager.*

3. To detect the HCE event, use the *addHCEEventListener*() method of the *NFCAdapter* interface to register a listener. Use the *sendHostAPDUResponse*() method of the *NFCAdapter* interface to send a host APDU response to a contactless front-end. (APDU - Application Protocol Data Unit - is defined in the ISO/IEC 7816-4 specification.)

4. To register an AID for a specific category and secure element type, use the *registerAID*() method of the *NFCAdapter* interface.

5. To retrieve the registered AIDs for a specific category and secure element type, use the *getAIDsForCategory*() method of the *NFCAdapter* interface.

6. Remove the registered listeners when they are no longer necessary, and disable NFC card emulation

Conclusion. Using a smart watch as a pass can be very convenient. Implementing an application that allows you to emulate an access card is not so difficult if the pass used only works on ID transfer. If the pass uses a secure transaction, then for its emulation it is necessary to know the data transfer protocol.

REFERENCES

1. How to Use an NFC Reader for Access Control [Electronic resource]. –Mode of access: https://www.getkisi.com/lessons/how-to-use-an-nfc-reader/. –Date of access: 4.03.2020.
2. Tizen Web Guides [Electronic resource]. –Mode of access: https://developer.tizen.org/development/guides/web-application/. –Date of access: 4.03.2020.
3. How to Copy or Clone Access Cards and Key Fobs [Electronic resource]. – Mode of access: https://www.getkisi.com/blog/how-to-copy-access-cards-and-keyfobs/. – Date of access: 4.03.2020.