It’s Time to Replace Your Wallet with Mobile Tizen Devices!

Arron Wang
Payment Overview
Traditional Payment

- Cash
- Paper Cheque
- Debit/Credit card
- Online transaction
Mobile Payment

- SMS based transaction
- Direct mobile billing
- QR code payment
- Audio signal-based payment
- Contactless NFC payment
Key Mobile Payment Characteristics

- **Security, privacy and trust**
- **Simplicity and usability**
- **Interoperability**
- **Universality**
Why NFC?

- **Security** (Based on Secure Element)
  1. SIM/UICC
  2. *Embedded secure element (eSE)*
  3. *Smart microSD*

- **Interoperability** (Industry Standards)
  - ETSI
  - GlobalPlatform
  - EMVCo
  - ISO/IEC

- **Universality**

- **Simplicity** (Tap & Pay)
Key Communication Protocols for NFC Payment

• **SE and the mobile processor**
  - UICC: ISO7816
  - smart microSD: SD protocol
  - eSE: By default NFC CLF will export the API for user

• **SE and the NFC controller**
  - UICC: SWP/HCI
  - smart microSD: SWP/HCI
  - eSE: NFC-WI, SWP/HCI, I2C, SPI, DCLB

• **POS reader and the NFC controller**
  - NFC as a carrier
  - EMV Contactless
Tizen NFC Based Mobile Payment Architecture
Tizen NFC Payment Architecture

- Settings
- NFC Payment Application (eWallet)
- NFC Chip
- Embedded SE
- NFC Controller
- NFC Antenna
- UICC/SIM
- CRU
- Applet 1
- Applet 2
- Applet 3
- Smartcard Service
- OpenMobileAPI v1.2
- TAPI
- Telephony Framework
- Telephony Modem
- ISO7816
- Software
- Hardware

1. Seeld
2. SWP
3. NFC Chip
4. Neard
5. Hardware

OpenMobileAPI v1.2
Transaction Procedure

1. RF field on
2. Card activated
3. APDU command
4. APDU response

POS NFC Chip UICC
Contact between NFC & SIM

**NFC CONTROLLER (MASTER)**

- **SWP OUTPUT**
  - S1 (CLF to UICC)
  - S2 (Vcl)

- **GND**

**SIM (SLAVE)**

- **SWP INPUT**
  - S1 (Vcl)
  - S2 (UICC to CLF)

**CLF**

- **Terminal**
- **Power supply**
- **Coupling Coil**
- **SWMI**
- **Gnd**

**UICC**

- **C1**
- **C5**
- **C6**
- **C2**
- **C3**
- **C7**
- **V0**
- **F+**
- **F-**

SWP
Enablement Tips
Checklist for NFC payment

• **NFC Chip**
  • Chip hardware test
    • `NXP_SELF_TEST_ANTENNA`
    • `NXP_SELF_TEST_SWP`
  • Chip secure element configuration
    • `NXP_SE_DEFAULTMODE`
    • `NXP_SWP_DEFAULTMODE`
    • `NXP_EVT_SWP_SWITCH_MODE`
    • `UICC_GateList`
  • HCI configuration for host controller administration gate
    • `WHITELIST`
    • `SESSION_IDENTITY`
  • Implemented in the kernel driver or the library from NFC vendor
Checklist for NFC payment (cont)

- **Secure Element Contactless Management (SECM)**
  - Contactless activation states
  - The priority of each application

- **Contactless Registry Service (CRS)**
  - An SECM entity for UICC
  - CRS application provide the user to active/deactive, change priority of applications on the contactless interface
Example of using CRS application

1. Select the CRS application
sh-4.1# ./send-apdu /org/seeld/se/nfc0_uicc_se0/channel0 00:A4:04:00:09:A0:00:00:01:51:43:52:53:00
Response APDU [0x6f 0x16 0x84 0x9 0xa0 0x0 0x1 0x51 0x43 0x52 0x53 0x0 //CRS CRS Application AID TLV
  0xa5 0x9 // FCI Proprietary Template, length
  0x9f 0x8 0x2 0x1 0x0 //version
  0x80 0x2 0x0 0x11 //Global Update Counter
  0x90 0x0 ]

2. Get the status of the contactless application on the card
sh-4.1# ./send-apdu /org/seeld/se/nfc0_uicc_se0/channel0 80:F2:40:00:02:4F:00:00
Response APDU [0x61 0x1f
  0x4f 0xe 0x32 0x50 0x41 0x59 0x2e 0x53 0x59 0x53 2e 0x44 0x44 0x46 0x30 0x31 // Application AID
  0x9f 0x70 0x2 0x7 0x0 // Application Lifecycle State, Activation State is encoded on the second byte, 00 is deactived
  0x80 0x2 0x0 0x8 // Application Update Counter
  0x81 0x1 0x0 // Selection Priority
  0x88 0x1 0x0 // Display Required Indicator
]

3. Update the status of selected application
sh-4.1# ./send-apdu /org/seeld/se/nfc0_uicc_se0/channel0
  80:F0://SET STATUS
  01://Status type: Availability State over the Contactless Interface
  01://Status value: Actived
Response APDU [0x90 0x0 ]//operation successfully
HCI Procedures

Need implement in kernel or library from NFC chip vendor

UICC gate | UICC administration gate | Host controller administration gate | Terminal administration gate | Terminal gate
---|---|---|---|---
ADM_CLEAR_ALLPIPE | No response if whitelist is not set | ANY_SET_PARAMETER (WHITELIST, UICC) | ADM_NOTIFY_ALL_PIPE_CLEARED | ANY_OK
ADM_CLEAR_ALLPIPE | ADM_NOTIFY_ALL_PIPE_CLEARED | ANY_OK | ADM_NOTIFY_PIPE_CREATED | ANY_OK
ADM_CREATE_PIPE | ANY_OK | ANY_OK | ANY_OPEN_PIPE | ANY_OK
ANY_OK | ANY_OK | ANY_OK | ANY_OK | ANY_SET_PARAMETER | ANY_OK
ANY_OK | ANY_OK | ANY_OK | ANY_OK | ANY_OK
Runtime monitor SWP data between host and SE

1. CLF active SWP link, UICC return ACT_SYNC command, return the SYNC_ID of the card to CLF module
2. CLF send U-Frame: RSET command F9, provide: endpoint window size and capability. window size=04
3. UICC return response U-Frame: UA, acknowledge received RSET command and accept RSET command
4. UICC send I-Frame: cmd=03=ANY_OPEN_PIPE, notify CLF to open pipeId=01.
5. CLF send response I-Frame: response=00=ANY_OK, notify UICC open pipeId=01 success.
6. UICC return S-Frame: Type=RR=Receive Ready.
7. UICC send I-Frame: cmd=02=ANY_GET_PARAMETER, and registry=01, means get the argument is 8 byte SESSION_IDENTITY, default value all FF.
8. CLF return response I-Frame: response=00=ANY_OK, and return SESSION_IDENTITY=FFFFFFFFFFFFFFF.
9. CLF return S-Frame: Type=RR=Receive Ready.
10. UICC send I-Frame: cmd=14=ADM_CLEAR_ALL_PIPE CLF should return response=00=ANY_OK but no response after that ☹
11. CLF return S-Frame: Type=RR=Receive Ready.
Demo Videos