Connecting to Android NFC P2P

Hardware used

- MSP-EXP430F5438 with TRF7970ATB
- Nexus S – NFC capable cell phone
  - Download API Demos from Android Market (https://market.android.com/details?id=com.hmh.api&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5obWguYXBpIl0.)

Establishing connection

The Android cell phone is continuously trying to find tags or P2P devices.

- **NFC→"Foreground dispatch"** from API Demos is continually sending polling requests looking for available devices. Once it detects something and receives data, it will display it.
- Note that even if the application is not running, the Android OS is still sending requests periodically. If it finds a device and receives data, it will ask the user to choose the appropriate program to do something with it.

Sense Request

- Android seems to send different requests but for NFC, the relevant request is a SENSF_REQ
- This request is referred to as POLLING in ISO/IEC 18092 (NFCIP-1)
- The request is send in FeliCa format (NFC-F) at 424kbps
- The SENSF_REQ has the following format:

Byte 1	Byte 2 - 3	Byte 4	Byte 5
00h	SC	RC	TSN

- Our Device will get:

```
06 00 FF FF 01 01
|  |  |  |  |  |
|  |  |  |  |  |-> Time slots (to reduce option of collision)
|  |  |  |  |----> Request Code (System Code info requested)
|  |  |  |-------\
|  |  |----------/ System Code (MUST be FFFFh)
|  |--------------> SENSF_REQ command
|-----------------> Length = 6Bytes
```

Sense Request

- The TRF7970A NFC Target protocol register (0x19) is used to detect the bit rate and protocol type used by the Initiator on first command
- We mask the register clearing the 2 msb and we are interested in the value:

```
0x13    :   xx010011
            ||||||-\ 424 kbps
            |||||--/
            ||||---> Command is not ISO14443B
            |||----> First command is not SENS_REQ or ALL_REQ
            ||-----> Coding is FeliCa
            |------> RFU
```

- After receiving this value on Register 0x19, the data is obtained from the FIFO and verified to be a SENSF_REQ (as shown in previous slide)
- Receiving this command indicates that we are in Target Passive Mode.
  - **ISOControl register must be written with a 0x23 (target, passive, NFC mode, 424kbps)**
  - This can be done after the protocol is detected or it can be done once during initialization of the device if we are only interested in this protocol

Sense Response

- The MSP430 must now respond with a SENSF_RES message which has the following format:

Byte 1	Byte 2-9	Byte 10-11	Byte 12-14	Byte 15	Byte 16	Byte 17	[Byte 18-19]
01h	NFCID2	PAD0	PAD1	$\mathbf{MRTI_{CHECK}}$	$\mathbf{MRTI_{UPDATE}}$	PAD2	[RD]

- An example:

14 01 01fe da335b2d56f3 c0c1c2c3c4c5c6c7 0fab
| | | | | |- System Code
| | | | |---------- X
| | | |--------------------------- NFCID[3:8] Random
| | |---------------------------------- NFCID[1:2] NFC-DEP support
| |--------------------------------------- SENSF_RES Command
|-- Length = 20 bytes

Attribute Request

- When the host gets the SENSF_REQ, it receives our NFCID and it can now send us an ATR_REQ
- And ATR REQ has the following format:

Byte 1	Byte 2	Byte 3–12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17–17+n
D4h	00h	**NFCID3$_I$**	**DID$_I$**	**BS$_I$**	**BR$_I$**	**PP$_I$**	[**G$_I$0** ... **G$_I$n**]

- An example:

1E D400 01feda335b2d56f3c0c1 00 00 00 32 46666D01011003020001040196
| | | | | | | |- General bytes: Magic key and
| | | | | | | data per LLCP protocol
| | | | | | |------ Can send bytes 0-255, General
| | | | | | | bytes are available
| | | | | |--------- Rx supported bit rates
| | | | |------------ Tx supported bit rates
| | | |--------------- No DiDi used
| | |---------------------------- **Our NFCID**
| |--------------------------------------- ATR_REQ command
|-- Length = 30 bytes

Attribute Response

- If the ATR_REQ matches our ID, we respond we an ATR_RES:

Byte 1	Byte 2	Byte 3–12	Byte 13	Byte 14	Byte 15	Byte 16	Byte 17	Byte 18–18+n
D5h	01h	**$NFCID3_T$**	**DID_T**	**BS_T**	**BR_T**	TO	**PP_T**	[**G_T0** ... **G_Tn**]

- An example:

1C D501 01feda335b2d56f3c0c1 00 00 00 32 46666D0101100302FFFF
| | | | | | | |- General bytes: Magic key and
| | | | | | | data per LLCP protocol
| | | | | | |------ Can send bytes 0-255, General
| | | | | | | bytes are available
| | | | | |--------- Rx supported bit rates
| | | | |------------ Tx supported bit rates
| | | |--------------- No DiDi used
| | |---------------------------- **Our NFCID**
| |-- ATR_RES command
|--- Length = 28 bytes

DEP_REQ-DEP_RES

- From then on, the devices can exchange data using DEP_REQ, DEP_RES packets.
- The data in this packets will contain NDEF messages which should be processed in a higher layer.