PNFC20(CT-NFC-C325-01) Porting Guide for Android platform

V2.5.2

2021/3/24

1. ITRODUCTION

NFC Reader Module PNFC20(CT-NFC-C325-01) is an USB to UART interface device. It's operating frequency is 13.56Mhz, and support protocol standard contains: ISO 14443A/B, ISO 15693, Mifare, Felica(Card UID);

Figure1. PNFC20(CT-NFC-C325-01) module

2. INTERFACE DEFINITION

2.1 Overall Architecture

PNFC20 Broadcast Intent Service will provide Tag UID and Type of all supported protocol card; and read external sectors, like AFI, DSFID, User Memory and Block Security Status for different protocol tag card.

Note: PNFC20 Broadcast Intent Service is only available when your device connected PNFC20(CT-NFC-C325-01) device after power on.

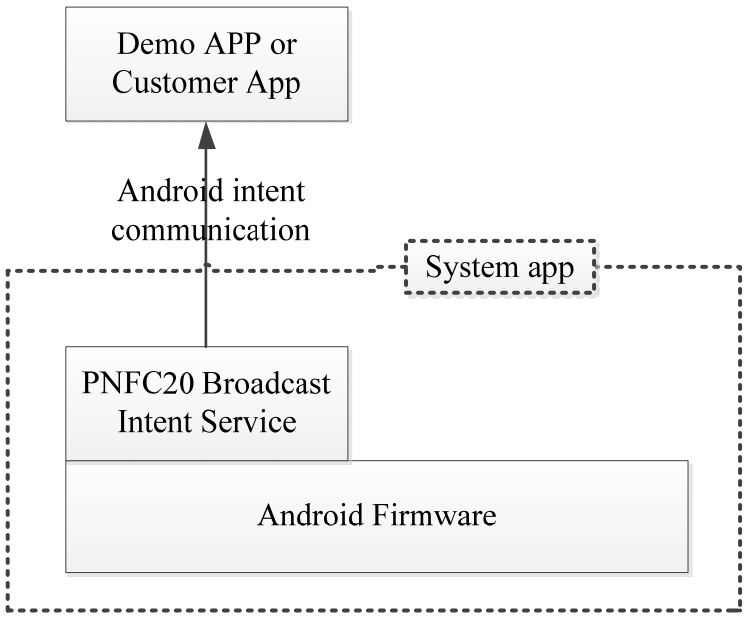

Figure2. PNFC20 Software Architecture

Get PNFC20(CT-NFC-C325-01) UID flow example

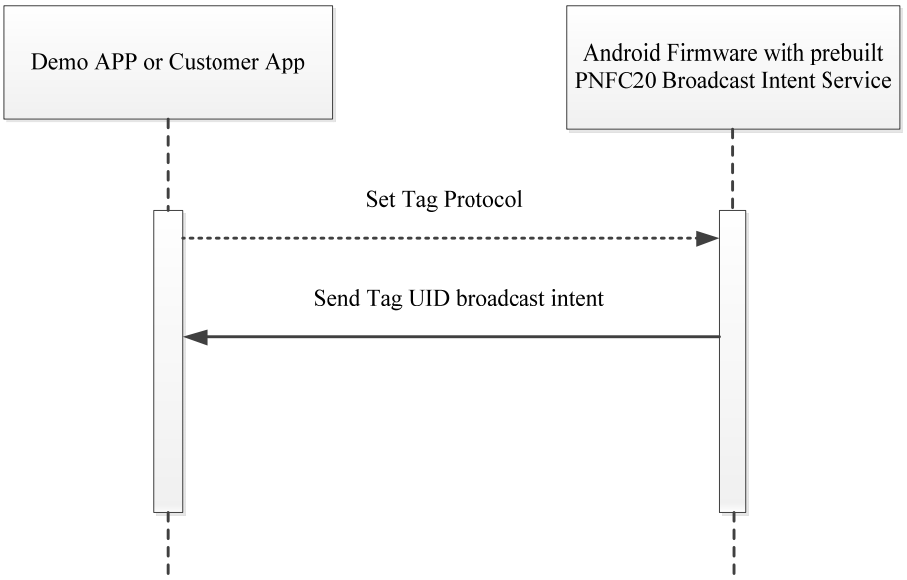

Figure3. PNFC20 Software Flow

Table1. PNFC20 Broadcast Intent Service APIs Support List

Tag Protocol		All protocol	ISO14443A	ISO14443B	NFC Type3(Felica)	ISO15693	NFC Type1
persist.sys.set_ct_tag_type Value		5	0	1	2	3	4
Intent action	action.TAG_DISCOVERED						
Extras	UID	V	V	V	V	V	V
	TYPE	V	V	V	V	V	V
	DSFID	X	X	X	X	V	X
	AFI	X	X	X	X	V	X
	SECURITY	X	X	X	X	V	X
	MEMORYDATA	X	V	X	X	V	X
	ATQA	X	V	X	X	X	X
	SKA	X	V	X	X	X	X
	PUPI	X	X	V	X	X	X
	APPDATA	X	X	V	X	X	X
	PROPOINFO	X	X	V	X	X	X

2.2 Interface for Use

2.2.1 Set Tag Protocol

Your or Customer's app can set Tag protocol for get tag's more sectors data. You can reference "Table1. PNFC20 Broadcast Intent Service APIs Support List" to check.

If you don't set Tag protocol, the PNFC20 Broadcast Intent Service app will use default value 5 for support All protocol.

SystemProperties

Key: "**persist.sys.set_ct_tag_type**" ---- String

Default Value: "5" ----- String

The following table is show Tag Protocol and Value list.

Tag Protocol	**Value**
ISO14443A	0
ISO14443B	1
NFC Type3(Felica)	2
ISO15693	3
NFC Type1	4
All protocol	5

2.2.2 Get Tag Data

These APIs are only sent by PNFC20 Broadcast Intent Service when working.

Broadcast Intent Action:

"**action.TAG_DISCOVERED**"

Intent Extras:

"UID" ---- String, presents tag's UID data;

"TYPE" ---- String, presents tag's type;

The following table is show Tag Protocol and Tag Type list:

Tag Protocol	Tag Type
ISO14443A	14443A
ISO14443B	14443B
ISO15693	15963
NFC Type1	Type1
NFC Type3	Felica

"DSFID" ---- String, presents DESFID

If the card type is 15693, DESFID can be obtained; otherwise, there is no data.

"AFI" ---- String, presents AFI;

If the card type is 15693, AFI can be obtained; otherwise, there is no data.

"SECURITY" ---- String, presents Block Security Status

If the card type is 15693, SECURITY can be obtained; otherwise, there is no data.

"MEMORYDATA" ---- String, presents Block Security Status

If the card type is 15693 or 14443A, MEMORYDATA can be obtained; otherwise, there is no data.

"ATQA" --- String, presents ATQA

If the card type is 14443A, DESFID can be obtained; otherwise, there is no data.

"SKA" --- String, presents SAK

If the card type is 14443A, SAK can be obtained; otherwise, there is no data.

"PUPI" --- String, presents PUPI

If the card type is 14443B, PUPI can be obtained; otherwise, there is no data.

"APPDATA" --- String, presents APPDATA

If the card type is 14443B, APPDATA can be obtained; otherwise, there is no data.

"PROPOINFO" --- String, presents PROPOINFO

If the card type is 14443B, PROPOINFO can be obtained; otherwise, there is no data.

3. Demo Code

```
private static final String ACTION_TAG_DISCOVERED = "action.TAG_DISCOVERED";
private static final String TYPE_15693 = "15963";
private static final String TYPE_14443A = "14443A";
private static final String TYPE_14443B = "14443B";
private static final String TYPE_FELICA = "Felica";
private static final String TYPE_TYPE1 = "Type1";
private static final String TYPE_ALL = "ALL";
private static final String PROPERTY = "persist.sys.set_ct_tag_type";
```

```
class GetIdReceiver extends BroadcastReceiver{
    private WeakReference<MainActivity> mainActivityWeakReference;
    public GetIdReceiver(MainActivity mainActivity){
        if(mainActivity != null){mainActivityWeakReference = new WeakReference<MainActivity>(mainActivity); }
    }
    @Override
    public void onReceive(Context context, Intent intent) {
        if(intent.getAction().equals(ACTION_TAG_DISCOVERED)){
            id_text.setText(intent.getStringExtra( name: "UID"));
            type_text.setText(intent.getStringExtra( name: "TYPE"));
            display(intent.getStringExtra( name: "TYPE"));
            if (TYPE_14443A.equals(intent.getStringExtra( name: "TYPE"))){
                atqa_text.setText(intent.getStringExtra( name: "ATQA"));
                sak_text.setText(intent.getStringExtra( name: "SKA"));
                data_text.setText(intent.getStringExtra( name: "MEMORYDATA"));
            }else if (TYPE_14443B.equals(intent.getStringExtra( name: "TYPE"))){
                pupi_text.setText(intent.getStringExtra( name: "PUPI"));
                appdata_text.setText(intent.getStringExtra( name: "APPDATA"));
                proto_text.setText(intent.getStringExtra( name: "PROPOINFO"));
            }
            else if (TYPE_15693.equals(intent.getStringExtra( name: "TYPE"))){
                dsfid_text.setText(intent.getStringExtra( name: "DSFID"));
                afi_text.setText(intent.getStringExtra( name: "AFI"));
                security_text.setText(intent.getStringExtra( name: "SECURITY"));
                data_text.setText(intent.getStringExtra( name: "MEMORYDATA"));
            }
        }
    }
}
```

```
private void registerReceiver(){
    IntentFilter filter = new IntentFilter();
    filter.addAction(ACTION_TAG_DISCOVERED);
    getIdReceiver = new GetIdReceiver( mainActivity: this);
    registerReceiver(getIdReceiver, filter);

}
```