1. Add all files in the libs folder to the project



Add it in the build.gradle of the APP.

Add it in android

sourceSets {  
 main {  
 jniLibs.srcDirs = [**'libs'**]  
 }  
}

Add in dependencies

implementation files(**'libs/RfidImpl\_class.jar'**)

2. Define the interface

String path = getDir(**"lib"**, Context.***MODE\_PRIVATE***).getAbsolutePath();  
**rfids** = **new** rfid\_inst(**this**,path);

Note that multiple Activities only need to create one instance

Connect to the Bluetooth device

rfid\_inst.*rfid*.open(device, **callb**);

The device has two working modes. After the connection is successful, switch through the following instructions

rfid\_inst.*rfid*.set\_DeviceMode(RfidImpl.***DeviceMode\_UHF***);  
rfid\_inst.*rfid*.set\_DeviceMode(RfidImpl.***DeviceMode\_2DRFID***);

Read the QR code

rfid\_inst.*rfid*.read\_RFID();

The driver uses an asynchronous architecture. After all instructions are issued, they are received, returned and processed by the handler. The device status and connection status, as well as the results of card reading and code scanning will be transmitted to the handler, as follows

**private final** Handler **mRXHandler** = **new** Handler() {  
 @Override  
 **public void** handleMessage(Message msg) {  
 **switch** (msg.**what**) {  
 **case** RfidImpl.***logString***:  
 MLog.*log*(**"uhf"**,**""**+msg.**obj**);  
 **break**;  
 **case** RfidImpl.***TXPower\_Data***:*//Set the RF output power* Log.*e*(**TAG**,msg.**arg1**+**" "**+msg.**arg2**);  
 **if**(msg.**arg1** == 1){ *//read* **if** (msg.**arg2** != -1) {  
 **if**(**datav**!=**null**)**datav**.setText(**""**+msg.**arg2**);  
 showTextToast(getResources().getString(R.string.***mnow\_txpower***) + msg.**arg2**+**" dBm"**);  
 }  
 }**else**{ *//set* **if** (msg.**arg2** != -1){  
 showTextToast(getResources().getString(R.string.***msetok***));  
 }  
 }  
 **break**;  
  
 **case** RfidImpl.***RFID\_2D***: *//Recognize the QR code* **mRFIDlist**.addRFID((String)msg.**obj**);**mRFIDlist**.notifyDataSetChanged();  
 **break**;  
 **case** RfidImpl.***Timeout***: *//Timeout response* showTextToast(getResources().getString(R.string.***mdeviceTimeout***));  
 **break**;  
 **case** RfidImpl.***DeviceError***:  
 showTextToast(getResources().getString(R.string.***mdeviceError***));  
 **break**;  
 **case** RfidImpl.***CMD\_WakeUp***: *//Wake up the device from sleep* Log.*e*(**TAG**,**"Device Wakeup.."**);  
 updateMode();  
 **break**;  
  
 **case** RfidImpl.***searchTagBack***: *//UHF search for tags //UHF\_IScand.restart();* **if**(!**mIsSearching**){  
 *// muhf\_Adapter.clear();muhf\_Adapter.notifyDataSetChanged();* }  
 Log.*e*(**TAG**,**"Read Tag "** + msg.**obj**);  
 String tag = (String) msg.**obj**;  
 newTAG\_Find(tag);**mIsSearching** = **true**;  
 **break**;  
 **case** RfidImpl.***searchTagFinish***:*//UHF execution completed* Log.*e*(**TAG**,**"scan UHF finish"**);  
 **break**;  
 */\*\*/* **default**:  
 **break**;  
 }  
 }  
};

For more detailed instructions, it is recommended to view the provided source code