# Hoin Printer

# Android File

**Shenzhen Hoin Electronic Technology Co. Ltd**

2018.11.12

**Import aar package**

**Open Android Studio, create a new project, import hoinprinterlib-release.aar, hoinSdk-release.aar package into the project's libs directory, and add dependencies in build.gradle, as shown below:**

repositories {  
 flatDir {  
 dirs 'libs' // aar directory  
 }  
}  
dependencies {  
 implementation 'am.util:printer:2.1.0'  
 implementation 'com.inuker.bluetooth:library:1.4.0'  
 implementation(name: 'hoinprinterlib-release', ext: 'aar')  
 implementation(name: 'hoinSdk-release', ext: 'aar')  
 …  
}

## Add permissions

## Fill in the following permissions in AndroidManifest.xml:

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />  
<uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE" />  
<uses-permission android:name="android.permission.READ\_LOGS" />  
<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>  
<uses-permission android:name="android.permission.BLUETOOTH\_ADMIN" />  
<uses-permission android:name="android.permission.BLUETOOTH" />  
<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

**Initialization instance**

HoinPrinter mHoinPrinter = HoinPrinter.getInstance(context, mode, callback);

Parameter explanation:

Context: type Context, the current context

Mode: mode, type int, initialization mode, 0 is Bluetooth, 1 is WIFI, 2 is USB

Callback: The event callback interface, type is PrinterCallback. The interface contains connection status callbacks, error callbacks, and event callbacks. For details, see PrinterCallback below.

**PrinterCallback：**

Public interface PrinterCallback {

     Public void onState(int state); //state callback, parameter is status code, refer to status code definition

     Public void onError(int errorCode); //Error callback, parameter is error code, refer to error code definition

     Public void onEvent(PrinterEvent event); //Event callback, parameter is PrinterEvent event, refer to PrinterEvent definition

}

**Status code definition**:

Public static final int BT\_STATE\_DISCONNECTED = 0; //Bluetooth disconnected

Public static final int BT\_STATE\_LISTEN = 1; //Bluetooth is listening

Public static final int BT\_STATE\_CONNECTING = 2; //Bluetooth connecting

Public static final int BT\_STATE\_CONNECTED = 3; //Bluetooth connected

Public static final int WIFI\_STATE\_CONNECTED = 4; //wifi connected

Public static final int WIFI\_STATE\_DISCONNECTED = 5; //wifi disconnected

Public static final int USB\_STATE\_CONNECTED = 6; //usb connected

Public static final int USB\_STATE\_DISCONNECTED = 7; //usb disconnected

**Error code definition**

Public static final int BT\_NOT\_AVALIBLE = 1000; //Bluetooth is not available on this device

Public static final int BT\_UNABLE\_CONNECT\_TO\_DEVICE = 1001; //Unable to connect to Bluetooth

Public static final int BT\_CONNECTION\_LOST = 1002; //Unable to connect Bluetooth or Bluetooth disconnect

Public static final int CONTEXT\_ERROR = 1003; //Context error

Public static final int WIFI\_SEND\_FAILED = 1004; //WIFI failed to send data

Public static final int WIFI\_CONNECT\_ERROR = 1005; //WIFI connection failed

Public static final int USB\_NOT\_FIND\_DEVICE = 1006; //USB connection failed

Public static final int USB\_NO\_PERMISSION = 1007; //USB does not have permission

Public static final int BT\_NO\_PERMISSION = 1008; //Bluetooth has no permissions

Public static final int DEVICE\_NOT\_CONNECTED = 1009; //Device not connected

Public static final int IMAGE\_NOT\_FONUD = 1010; //Device not connected

Public static final int NULL\_POINTER\_EXCEPTION = 9999; // null pointer exception

**PrinterEvent：**

public PrinterEvent(int event, Object object){  
 this.event = event; // Event type

this.object = object; //Event with information }

**Event type definition**

public static final int EVENT\_WIFI\_RECEIVE\_DATA = 4;

//Bluetooth searches device events, object is BluetoothDevice type, is the Bluetooth searched device

Public static final int EVENT\_FIND\_BT\_DEVICE = 1;

//Bluetooth search completed event

Public static final int EVENT\_DISCOVERY\_BT\_FINISHED = 2;

//Bluetooth receives a data event, object is a byte[] array type, which is the received data.

Public static final int EVENT\_BT\_RECEIVE\_DATA = 3;

//WIFI receives a data event, object is a byte[] array type, which is the received data.

Public static final int EVENT\_WIFI\_RECEIVE\_DATA = 4;

## Scan Bluetooth

mHoinPrinter.startBtDiscovery();

The scanned device is returned by the onEvent though PrinterCallback

## Stop scanning Bluetooth

mHoinPrinter.stopBtDiscovery();

## Get a paired Bluetooth device

Set<BluetoothDevice> devices = mHoinPrinter.getPairedDevice();

## Connect device

mHoinPrinter.connect(object);

When the mode is Bluetooth, the parameter is the mac address of the Bluetooth;

When the mode is WIFI, the parameter is the IP address of the device;

When the mode is USB, the parameter is null.

## Disconnect device

mHoinPrinter.destroy();

## Change the printer size type (58mm or 80mm)

mHoinPrinter.switchType(type58);

The parameter is boolean type, true mean to switch to 58mm printer, false mean to switch to 80mm printer

## Set centering print effect

mHoinPrinter.setCenter(center);

The parameter is boolean, true is centered, false is not centered

## Print string

mHoinPrinter.printText(str, mDoubleHeight, mDoubleWidth, mFontBold, mCenter);

There are five parameters

Parameter 1: Printed text string, the type is String

Parameter 2: Whether to print double high, the type is boolean

Parameter 3: Whether to print in double width, the type is boolean

Parameter 4: Whether the font is bold, the type is boolean

Parameter 5: Whether it is centered, the type is boolean

## Image printing

mHoinPrinter.printImage(path, true);

There are two parameters

Parameter 1: Image path, type is String

Parameter 2: Whether it is centered, the type is boolean

## Print QR code

mHoinPrinter.printQRCode(str);

There is one parameter

Parameter 1: The printed QR code is the corresponding string content

## Print 1D bar code

mHoinPrinter.printBarCode();

## Open cash drawer box

mHoinPrinter.openBox();

## Cutter test

mHoinPrinter.testCutting();

## Destroy instance

mHoinPrinter.destroy();