Gprinter Android SDK V2.1 使用说明

Gprinter Android SDK 旨在佳博用户更快速，更高效的在 Android 平台下开发和使用佳博打印机。如果您在使用 SDK 中碰到问题，或者发现 BUG，请留言

一、下载 GprinterSDKV2.1

GprinterSDKV2.1 可打电话到 0756-3866865，填写客户资料后，即可获得。

二、新建工程，导入 gprinter-2.1.jar 包

向工程的 libs 文件中拷贝 gprinter-2.1.jar，android sdk API >= 14

```
<uses-sdk
    android:minSdkVersion="14"
    android:targetSdkVersion="18" />
```

三、注册服务和权限

注册服务在 AndroidManifest 文件中添加服务，gprinter-2.1.jar 中提供了打印服务

```
<service
        android:name="com.gprinter.service.GpPrintService"
        android:label="GpPrintService"
        android:process=":remote"
        android:enabled="true"
        android:exported="true"
         >
        <intent-filter>
            <action android:name="com.gprinter.aidl.GpPrintService" />
        </intent-filter>
    </service>
```

注册权限

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-feature android:name="android.hardware.usb.host" />
<uses-permission android:name="android.hardware.usb.accessory" />
 <uses-permission android:name="android.permission.DEVICE_POWER"/>
```

```
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />

    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

四、添加 aidl 文件

新建包，包名为 com.gprinter.aidl,向包中添加文件，文件名为 GpService。aidl 内容为

```
package com.gprinter.aidl;
interface GpService{
    int openPort(int PrinterId,int PortType,String DeviceName,int PortNumber);
    void closePort(int PrinterId);
    int getPrinterConnectStatus(int PrinterId);
    int printeTestPage(int PrinterId);
    int queryPrinterStatus(int PrinterId,int Timesout);
    int getPrinterCommandType(int PrinterId);
    int sendEscCommand(int PrinterId, String b64);
    int sendTscCommand(int PrinterId, String  b64);
}
```

五、启动并绑定 GpPrintService 服务

在 OnCreate 中启动并绑定服务

```
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.e(DEBUG_TAG, "onCreate");
        startService();
        connection();
    }
```

启动服务

```
private void startService() {
```

```
        Intent i= new Intent(this, GpPrintService.class);
        startService(i);
         try {
             Thread.sleep(1000);
         } catch (InterruptedException e) {
             e.printStackTrace();
         }
    }
```

绑定服务

```
private GpService mGpService = null;
private PrinterServiceConnection conn = null;

class PrinterServiceConnection implements ServiceConnection {
        @Override
        public void onServiceDisconnected(ComponentName name) {
            Log.i("ServiceConnection", "onServiceDisconnected() called");
            mGpService = null;
        }
        @Override
        public void onServiceConnected(ComponentName name, IBinder service) {
            mGpService =GpService.Stub.asInterface(service);
        }
    };
private void connection() {
        conn = new PrinterServiceConnection();
        Intent intent = new Intent("com.gprinter.aidl.GpPrintService");
        bindService(intent, conn, Context.BIND_AUTO_CREATE); // bindService
    }
```

六、使用打印服务的 AIDL 接口，接口的详细描述请看 GpService.aidl 说明

打开和关闭端口操作

1、 注册状态接收广播，通过此广播可以获取当前端口的连接状态

广播接收的 Intent，

GpPrintService.PRINTER_ID 返回打印机的 ID 序号

GpPrintService.CONNECT_STATUS 返回状态

GpPrintService 可以同时连接三台打印机，可以通过此广播获取到哪台打印机处于何种状态

```
public static final String ACTION_CONNECT_STATUS = "action.connect.status";
private void registerBroadcast() {
        IntentFilter filter = new IntentFilter();
        filter.addAction(ACTION_CONNECT_STATUS);
        this.registerReceiver(PrinterStatusBroadcastReceiver, filter);
    }
private BroadcastReceiver PrinterStatusBroadcastReceiver = new BroadcastReceiver() {
        @Override
    public void onReceive(Context context, Intent intent) {
            if (ACTION_CONNECT_STATUS.equals(intent.getAction())) {
                int type = intent.getIntExtra(GpPrintService.CONNECT_STATUS, 0);
                int id = intent.getIntExtra(GpPrintService.PRINTER_ID, 0);
                Log.d(DEBUG_TAG, "connect status " + type);
                if (type == GpDevice.STATE_CONNECTING) {

                } else if (type == GpDevice.STATE_NONE) {

                } else if (type == GpDevice.STATE_VALID_PRINTER) {

                } else if (type == GpDevice.STATE_INVALID_PRINTER) {

                }
            }
        }
    };
```

2、调用打开端口 int openPort(int PrinterId,int PortType,String DeviceName,int PortNumber);如果端口已经连接则会返回 ERROR_CODE.*DEVICE_ALREADY_OPEN*，端口打开过程中的连接状态通过上面的广播返回

3、调用关闭端口 int closePort(int PrinterId);端口关闭过程中的连接状态通过上面的广播返回，断开连接后再次调用连接，需延时 1-10s 时间再次连接，否则可能连接不上，具体情况视平板固件而定

获取打印机状态

1、通过调用 int getPrinterConnectStatus(int PrinterId);可以获取打印机的连接状态

2、通过调用 int queryPrinterStatus(int PrinterId,int Timesout);可以获取打印机的错误状态

```
try {
        int status = mGpService.queryPrinterStatus(mPrinterIndex,500);
        String str = new String();
        if (status == GpCom.STATE_NO_ERR) {
            str = "打印机正常";
        } else if ((byte) (status & GpCom.STATE_OFFLINE) > 0) {
            str = "打印机脱机";
        } else if ((byte) (status & GpCom.STATE_PAPER_ERR) > 0) {
            str = "打印机缺纸";
        } else if ((byte) (status & GpCom.STATE_COVER_OPEN) > 0) {
            str = "打印机开盖";
        } else if ((byte) (status & GpCom.STATE_ERR_OCCURS) > 0) {
            str = "打印机出错";
        }
        Toast.makeText(getApplicationContext(),
                "打印机：" + '0' + " 状态：" + str, Toast.LENGTH_SHORT).show();
    } catch (RemoteException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
```

3、通过调用 int getPrinterCommandType(int PrinterId);可以获取打印机的命令类型，佳博打印机分为票据打印机和标签打印机

```
try {
        int type = mGpService.getPrinterCommandType(mPrinterIndex);
        if (type == GpCom.ESC_COMMAND) {
            Toast.makeText(getApplicationContext(), "打印机使用 ESC 命令",
                    Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "打印机使用 TSC 命令",
                    Toast.LENGTH_SHORT).show();
```

```
            }
        } catch (RemoteException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
```

打印测试页

1、如果打印机连接成功后可以通过 int printeTestPage(int PrinterId); 打印测试页。

向打印机发送数据

1、如果连接的为标签打印机则调用 int sendTscCommand(int PrinterId, String b64); 发送数据

2、如果连接的为票据打印机则调用 int sendEscCommand(int PrinterId, String b64); 发送数据

发送数据之前 需查询打印机的状态

票据打印机测试页内容如图 1：

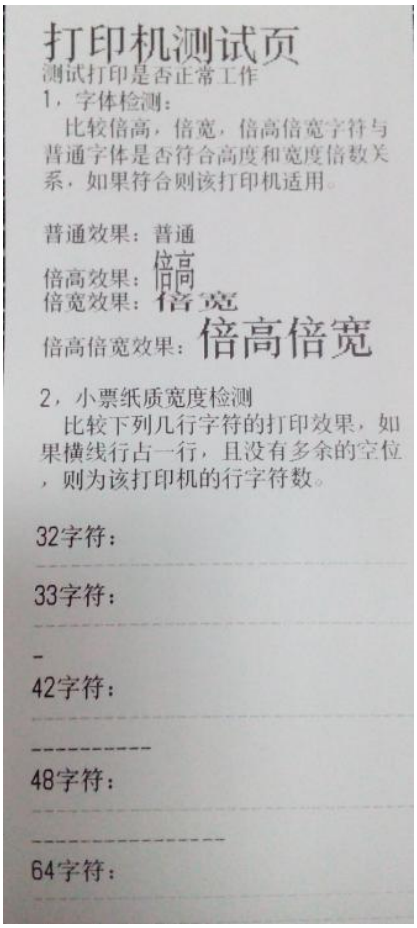

图 1

标签打印机测试页内容如图 2：

图 2

七、打印机命令格式说明

1、票据的内容编辑，按照 ESC 指令编写，可以参考 EscComand API 说明文件打印效果如图 3

```
        EscCommand esc = new EscCommand();
        esc.addPrintAndFeedLines((byte)3);
        esc.addSelectJustification(JUSTIFICATION.CENTER);//设置打印居中
        esc.addSelectPrintModes(FONT.FONTA, ENABLE.OFF,ENABLE.ON, ENABLE.ON, ENABLE.OF
F);//设置为倍高倍宽
        esc.addText("Sample\n");    //  打印文字
        esc.addPrintAndLineFeed();

        /*打印文字*/
        esc.addSelectPrintModes(FONT.FONTA, ENABLE.OFF,ENABLE.OFF, ENABLE.OFF, ENABLE.O
FF);//取消倍高倍宽
        esc.addSelectJustification(JUSTIFICATION.LEFT);//设置打印左对齐
        esc.addText("Print text\n");    //  打印文字
        esc.addText("Welcome to use Gprinter!\n");    //  打印文字
        esc.addPrintAndLineFeed();
        /*打印图片*/
        esc.addText("Print bitmap!\n");    //  打印文字
        Bitmap b = BitmapFactory.decodeResource(getResources(),
                R.drawable.gprinter);
        esc.addRastBitImage(b,b.getWidth(),0);    //打印图片

        /*打印一维条码*/
```

```
        esc.addText("Print code128\n");   //  打印文字
        esc.addSelectPrintingPositionForHRICharacters(HRI_POSITION.BELOW);//设置条码可识别字符位置在条码下方
        esc.addSetBarcodeHeight((byte)60); //设置条码高度为 60 点
        esc.addCODE128("Gprinter");  //打印 Code128 码
        esc.addPrintAndLineFeed();

        /*QRCode 命令打印
            此命令只在支持 QRCode 命令打印的机型才能使用。
            在不支持二维码指令打印的机型上，则需要发送二维条码图片
        */
        esc.addText("Print QRcode\n");   //  打印文字
        esc.addSelectErrorCorrectionLevelForQRCode((byte)0x31); //设置纠错等级
        esc.addSelectSizeOfModuleForQRCode((byte)3);//设置 qrcode 模块大小
        esc.addStoreQRCodeData("www.gprinter.com.cn");//设置 qrcode 内容
        esc.addPrintQRCode();//打印 QRCode
        esc.addPrintAndLineFeed();

        /*打印文字*/
        esc.addSelectJustification(JUSTIFICATION.CENTER);//设置打印左对齐
        esc.addText("Completed!\r\n");   //  打印结束
        esc.addPrintAndFeedLines((byte)8);

        Vector<Byte> datas = esc.getCommand(); //发送数据
        Byte[] Bytes = datas.toArray(new Byte[datas.size()]);
        byte[] bytes = ArrayUtils.toPrimitive(Bytes);
        String str = Base64.encodeToString(bytes, Base64.DEFAULT);
        int rel;
        try {
            rel = mGpService.sendEscCommand(mPrinterIndex, str);
            GpCom.ERROR_CODE r=GpCom.ERROR_CODE.values()[rel];
            if(r != GpCom.ERROR_CODE.SUCCESS){
                Toast.makeText(getApplicationContext(),GpCom.getErrorText(r),
                        Toast.LENGTH_SHORT).show();
```

```
                }
        } catch (RemoteException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
```

图 3

2、标签的内容编辑，按照 TSC 指令发送，可以参考 TscComand API 说明文件

```
TscCommand tsc = new TscCommand();
        tsc.addSize(60, 60); //设置标签尺寸，按照实际尺寸设置
        tsc.addGap(0);           //设置标签间隙，按照实际尺寸设置，如果为无间隙纸则设置为 0
```

```
        tsc.addDirection(DIRECTION.BACKWARD,MIRROR.NORMAL);//设置打印方向
        tsc.addReference(0, 0);//设置原点坐标
         tsc.addTear(ENABLE.ON); //撕纸模式开启
        tsc.addCls();// 清除打印缓冲区
        //绘制简体中文
         tsc.addText(20,20,FONTTYPE.SIMPLIFIED_CHINESE,ROTATION.ROTATION_0,FONTMUL.MUL_
1,FONTMUL.MUL_1,"Welcome to use Gprinter!");
         //绘制图片
        Bitmap b = BitmapFactory.decodeResource(getResources(),
                R.drawable.gprinter);
        tsc.addBitmap(20,50, BITMAP_MODE.OVERWRITE, b.getWidth(),b);

        tsc.addQRCode(250, 80, EEC.LEVEL_L,5,ROTATION.ROTATION_0, " www.gprinter.com.cn
");
         //绘制一维条码
         tsc.add1DBarcode(20,250, BARCODETYPE.CODE128, 100, READABEL.EANBEL, ROTATION.R
OTATION_0, "Gprinter");
        tsc.addPrint(1,1); // 打印标签
        tsc.addSound(2, 100); //打印标签后 蜂鸣器响

        Vector<Byte> datas = tsc.getCommand(); //发送数据
        Byte[] Bytes = datas.toArray(new Byte[datas.size()]);
        byte[] bytes = ArrayUtils.toPrimitive(Bytes);
        String str = Base64.encodeToString(bytes, Base64.DEFAULT);
        int rel;
        try {
            rel = mGpService.sendTscCommand(mPrinterIndex, str);
            GpCom.ERROR_CODE r=GpCom.ERROR_CODE.values()[rel];
            if(r != GpCom.ERROR_CODE.SUCCESS){
                Toast.makeText(getApplicationContext(),GpCom.getErrorText(r),
                        Toast.LENGTH_SHORT).show();
                  }
        } catch (RemoteException e) {
            // TODO Auto-generated catch block
```

```
            e.printStackTrace();

        }
```

图 4