

智能 pos 打印机开发文档

目录

- 文档更新记录2
- 简介3
- 1 AIDL 方式打印机服务..... 4
- 2.蓝牙方式打印服务 10
- 3.状态反馈 13
- 4.打印服务参数说明 14

文档更新记录

文档版本	更新内容	日期
V1.0.0	原始版本	2017/9/1
V1.1.0	<ol style="list-style-type: none">1. 增加：aidl 自带对齐方式的文字打印接口说明2. 修改：aidl 位图打印的 Size 大小范围由 1-8 改为 1-16，单位由 48 像素点改为 24 像素点说明3. 修改：aidl 条码打印的宽高范围由 1-8 改为 1-16，单位由 48 像素点改为 24 像素点说明4. 修改：aidl 二维码打印的 Size 大小范围由 1-8 改为 1-16，单位由 48 像素点改为 24 像素点说明，并且增加纠错等级参数说明5. 增加：实现 aidl ESC/POS 指令解析接口的功能说明6. 修改：aidl 执行打印接口添加打印后走纸点行参数（打印后，打印机不再自动走纸）说明7. 增加：添加蓝牙打印功能说明8. 增加：当前打印任务完成后的状态广播说明	2017/12/21

简介

Pos 机内置热敏打印机，支持 58mm 打印纸，每行 384 个像素点，不带切刀，可使用 app 直接打印小票。

App 可以使用以下方式调用内置打印机：

1. 通过 AIDL 连接打印机
2. 通过蓝牙连接打印机

1 AIDL 方式打印机服务

连接打印机服务使用 AIDL 的方式，AIDL 是 Android Interface Definition language 的缩写，它是一种 Android 内部进程通信接口的描述语言，通过它我们可以定义进程间的通信接口。AIDL 提供封装好的常用打印指令，方便开发者快速接入打印机服务。

1.1 打印机服务的 AIDL 使用

1. 在开发者的工程项目中添加 AIDL 文件：

- 1) 包名：com.iposprinter.iposprinterservice

- 2) 文件：在 release 的 aidl.rar 开发包中有连个文件，

IPosPrinterService.aidl 和 IPosPrinterCallback.aidl

2. 在开发者控制打印的代码类中实现 ServiceConnection。

```
private ServiceConnection connectService = new
    ServiceConnection() {
        @Override
        public void onServiceConnected(ComponentName name, IBinder
            service) {
            mIPosPrinterService =
                IPosPrinterService.Stub.asInterface(service);
            setButtonEnable(true);
        }

        @Override
        public void onServiceDisconnected(ComponentName name) {
            mIPosPrinterService = null;
        }
    };
```

3. 调用 Application Context.bindService(), 并在 ServiceConnection 实现中进行传递。

注意：bindservice 是非阻塞调用，意味着调用完成后并没有立即绑定

成功，必须以 serviceConnected 为准。

//绑定服务

```
Intent intent=new Intent();  
intent.setPackage("com.iposprinter.iposprinterservice");  
intent.setAction("com.iposprinter.iposprinterservice.IPosPrintService");  
bindService(intent, connectService, Context.BIND_AUTO_CREATE);
```

4. 绑定服务后，现在可以调用 IPosPrintService 的各种接口进行打印了。

1.2 AIDL 接口说明

1. 打印机初始化

函数原型	printerInit
功能说明	打印机上电并初始化默认设置
参数说明	无
返回值	void
补充说明	使用时请先判断打印机当前状态，如果处于 PRINTER_IS_BUSY 状态，请等待

2. 打印机状态查询

函数原型	getPrinterStatus
功能说明	查询打印机当前状态
参数说明	无
返回值	打印机状态： 0:PRINTER_NORMAL 此时打印机空闲并状态正常可以启动新的打印 1:PRINTER_PAPERLESS 此时停止打印，如果当前打印未完成，加纸后需重打 2:PRINTER_THP_HIGH_TEMPERATURE 此时暂停打印，如果当前打印未完成，冷却后将继续打印，无需重打 3:PRINTER_MOTOR_HIGH_TEMPERATURE 此时不执行打印，冷却后需要初始化打印机，重新发起打印任务 4:PRINTER_IS_BUSY 此时打印机正在打印 5:PRINTE_ERROR_UNKNOWN 打印机异常
补充说明	发起打印之前，请先 check 打印机状态

3. 打印机浓度设置

函数原型	setPrinterPrintDepth
功能说明	设置打印机的打印浓度，对之后打印有影响，除非初始化
参数说明	depth: 浓度等级, 范围 1-10, 超出范围此功能不执行 默认等级 6
返回值	void
补充说明	无

4. 打印机字体设置

函数原型	setPrinterPrintFontType
功能说明	设置打印字体类型，对之后打印有影响，除非初始化
参数说明	typeface: 字体名称 ST(宋体)
返回值	void
补充说明	无

5. 打印机字体大小设置

函数原型	setPrinterPrintFontSize
功能说明	设置字体大小，对之后打印有影响，除非初始化
参数说明	fontsize: 字体大小，目前支持的 size 有 16、24、32、48，输入非法 size 执行默认值 24，单位像素点
返回值	void
补充说明	无

6. 打印机对齐方式设置

函数原型	setPrinterPrintAlignment
功能说明	设置对齐方式，对之后打印有影响，除非初始化
参数说明	alignment: 对齐方式 0--居左，1--居中，2--居右, 默认居中
返回值	void
补充说明	无

7. 打印机走纸

函数原型	printerFeedLines
功能说明	打印机走纸（强制换行，结束之前的打印内容后走纸 lines 行，此时马达空转走纸，无数据传送给打印机）
参数说明	lines: 打印机走纸行数(每行是一个像素点)
返回值	void
补充说明	无

8. 打印空白行

函数原型	printBlankLines
功能说明	打印空白行（强制换行，结束之前的打印内容后打印空白行，此时传送给打印机的数据全是 0x00）
参数说明	lines: 打印空白行数 限制最多 100 行 height: 空白行的高度(单位：像素点)
返回值	void
补充说明	行间空白建议使用此方法，不建议使用 printerFeedLines

9. 打印文字

函数原型	printText
功能说明	文字宽度满一行自动换行排版
参数说明	text: 要打印的文字字符串
返回值	void
补充说明	字体类型和大小与前一次打印相同

10. 打印指定字体类型和大小的文本

函数原型	printSpecifiedTypeText
功能说明	文字宽度满一行自动换行排版
参数说明	text: 要打印的文字字符串 typeface: 字体名称 ST（目前只支持一种宋体） fontsize: 字体大小，目前支持的 size 有 16、24、32、48，输入非法 size 执行默认值 24
返回值	void
补充说明	无

11. 打印指定字体类型、大小和对齐方式的文本

函数原型	printSpecFormatText
功能说明	文字宽度满一行自动换行排版
参数说明	text: 要打印的文字字符串 typeface: 字体名称 ST (目前只支持一种宋体) fontsize: 字体大小, 目前支持的 size 有 16、24、32、48, 输入非法 size 执行默认值 24 alignment: 对齐方式 (0 居左, 1 居中, 2 居右)
返回值	void
补充说明	无

12. 打印表格一行

函数原型	printColumnsText
功能说明	打印表格的一行, 可以指定列宽、列对齐方式
参数说明	colsTextArr: 各列文本字符串数组 colsWidthArr: 各列宽度数组 总宽度不能大于 $((384 / \text{fontsize}) \ll 1) - (\text{列数} + 1)$ (以英文字符计算, 每个中文字符占两个英文字符, 每个宽度大于 0), colsAlign: 各列对齐方式 (0 居左, 1 居中, 2 居右) isContinuousPrint: 是否继续打印表格 1: 继续打印 0: 不继续打印
返回值	void
补充说明	三个参数的数组长度应该一致, 如果 colsTextArr[i] 的内容宽度大于 colsWidthArr[i], 则文本换行

13. 打印图片

函数原型	printBitmap
功能说明	打印 bmp 图片数据
参数说明	alignment: 对齐方式 0--居左, 1--居中, 2--居右, 默认居中 bitmapSize: 位图大小, 传入大小范围 1~16, 超出范围默认选择 10 单位: 24 像素点 mBitmap: 图片 bitmap 对象 (最大宽度 384 像素)
返回值	void
补充说明	无

14. 打印条码

函数原型	printBarCode
功能说明	打印一维条码
参数说明	<p>data: 条码数据</p> <p>symbology: 条码类型</p> <p>0 -- UPC-A</p> <p>1 -- UPC-E (暂不支持)</p> <p>2 -- JAN13 (EAN13)</p> <p>3 -- JAN8 (EAN8)</p> <p>4 -- CODE39</p> <p>5 -- ITF</p> <p>6 -- CODABAR</p> <p>7 -- CODE93 (暂不支持)</p> <p>8 -- CODE128</p> <p>height: 条码高度, 取值 1 到 16, 超出范围默认取 6, 每个单位代表 24 个像素点高度</p> <p>width: 条码宽度, 取值 1 至 16, 超出范围默认取 12, 每个单位代表 24 个像素点长度</p> <p>textposition: 文字位置 0--不打印文字, 1--文字在条码上方, 2--文字在条码下方, 3--条码上下方均打印</p>
返回值	void
补充说明	无

15. 打印二维码

函数原型	printQRCode
功能说明	打印二维码
参数说明	<p>data: 二维码数据</p> <p>modulesize: 二维码块大小(单位:点, 取值 1 至 16), 超出设置范围默认取值 12 单位: 24 个像素点</p> <p>mErrorCorrectionLevel:</p> <p>0 -- 纠错级别L (7%),</p> <p>1 -- 纠错级别M (15%),</p> <p>2 -- 纠错级别Q (25%),</p> <p>3 -- 纠错级别H (30%)</p>
返回值	void
补充说明	无

16. 打印 byte 数据

函数原型	printRawData
功能说明	打印原始的 byte 数据
参数说明	rawPrintData: Byte 数据块
返回值	void
补充说明	无

17. ESC/POS 指令打印

函数原型	sendCMDRawData
功能说明	使用 ESC/POS 指令打印
参数说明	data: 指令 Byte 数据块
返回值	void
补充说明	无

18. 执行打印

函数原型	printerPerformPrint
功能说明	执行打印任务
参数说明	feedlines: 打印后走纸点行数（需用户自行设置，打印机打印完成后不再自动走纸）
返回值	void
补充说明	当执行完成各打印功能方法后，需要执行此方法，打印机才能执行打印，此方法执行之前需要判断打印机状态，当打印机处于 PRINTER_NORMAL 此方法有效，否则不执行

另：所有 callback 参数都是结果回调

2.蓝牙方式打印服务

2.1 模拟蓝牙

当打开 POS 机蓝牙时，可看到一个已经配对成功的蓝牙设备“lposPrinter”，

该设备会一直存在。此设备是有系统虚拟出的蓝牙打印机设备，是把 *POS* 机内部打印机模拟成蓝牙打印机，实际并不存在真的蓝牙打印机。模拟蓝牙打印机支持 *ESC/POS* 指令，指令支持情况详见《*IPOS ESC-POS* 指令说明》。

2.2 模拟蓝牙打印机的使用

1. 与该设备建立连接
2. 将指令和打印内容拼接成 *byte* 数据块
3. 通过蓝牙发送给 *IposPrinter* 打印机
4. 打印机完成打印任务

另：蓝牙打印机 *Demo* 中提供一个蓝牙工具类，使用该 *Demo* 测试时，请先点击“蓝牙打印机驱动加载”。

工具类 *BluetoothUtil*，标准的蓝牙连接工具类

```
public class BluetoothUtil{
    private static final String TAG = "BluetoothUtil";
    private static final UUID IPOSPRINTER_UUID =
        UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");
    private static final String IPosPrinter_Address = "00:AA:11:BB:22:CC";

    public static BluetoothAdapter getBluetoothAdapter(){
        return BluetoothAdapter.getDefaultAdapter();
    }
}
```

```

        public static BluetoothDevice getIposPrinterDevice(BluetoothAdapter
mBluetoothAdapter){
            BluetoothDevice IPosPrinter_device = null;
            Set<BluetoothDevice> devices =
mBluetoothAdapter.getBondedDevices();
            for (BluetoothDevice device : devices){
                if(device.getAddress().equals(IPosPrinter_Address))
                {
                    IPosPrinter_device =device;
                    break;
                }
            }
            return IPosPrinter_device;
        }

        public static BluetoothSocket getSocket(BluetoothDevice mDevice)
throws IOException
        {
            BluetoothSocket socket =
mDevice.createRfcommSocketToServiceRecord(IPOSPRINTER_UUID);
            socket.connect();
            return socket;
        }
    }
}

```

获取蓝牙耳机打印设备，连接打印机

```

// 1: Get BluetoothAdapter
mBluetoothAdapter = BluetoothUtil.getBluetoothAdapter();
if(mBluetoothAdapter == null)
{
    return;
}
//2: Get bluetoothPrinter Devices
mBluetoothPrinterDevice =
BluetoothUtil.getIposPrinterDevice(mBluetoothAdapter);
if(mBluetoothPrinterDevice == null)
{
    return;
}
//3: Get conect Socket
try {
    socket = BluetoothUtil.getSocket(mBluetoothPrinterDevice);
}

```

```

}
catch (IOException e)
{
    e.printStackTrace();
    return;
}

```

注意：需要在 *App* 的项目中添加蓝牙权限声明才能使用蓝牙设备

```
<manifest>
```

```
<uses-permission
```

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

```
<uses-permission
```

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

```
</manifest>
```

3.状态反馈

3.1 打印机状态反馈

用户需要通过接受广播的形式，来监听打印机的当前状态。

用户需要建立一个广播接受者来监听以下广播：

```

//打印机正常，并且处于空闲状态
private final String PRINTER_NORMAL_ACTION =
    "com.iposprinter.iposprinterservice.NORMAL_ACTION";

//打印机缺纸
private final String PRINTER_PAPERLESS_ACTION =
    "com.iposprinter.iposprinterservice.PAPERLESS_ACTION";

//打印机有纸
private final String PRINTER_PAPEREXISTS_ACTION =
    "com.iposprinter.iposprinterservice.PAPEREXISTS_ACTION";

//打印机热敏头温度过高
private final String PRINTER_THP_HIGHTEMP_ACTION =

```

```

"com.iposprinter.iposprinterservice.THP_HIGHTEMP_ACTION";

//打印机热敏头温度正常
private final String PRINTER_THP_NORMALTEMP_ACTION =
"com.iposprinter.iposprinterservice.THP_NORMALTEMP_ACTION";

//打印机马达温度过高
private final String PRINTER_MOTOR_HIGHTEMP_ACTION =
"com.iposprinter.iposprinterservice.MOTOR_HIGHTEMP_ACTION";

//打印机忙碌，正在打印
private final String PRINTER_BUSY_ACTION =
"com.iposprinter.iposprinterservice.BUSY_ACTION";

//当前打印任务打印完成
private final String PRINTER_CURRENT_TASK_PRINT_COMPLETE_ACTION =
"com.iposprinter.iposprinterservice.CURRENT_TASK_PRINT_COMPLETE_ACTION ";

```

另：用户主动查询状态请使用 `getPrinterStatus`，参考 1.2 AIDL 接口说明

3.2 指令回调反馈

接口方法 `callback` 提供 2 个反馈结果：

反馈函数	返回	
<code>onRunResult</code>	指令执行结果boolean isSuccess	true 执行成功， false 执行失败
<code>onReturnString</code>	指令执行结果 final String result result: 结果	result: 结果

4.打印服务参数说明

4.1 纸张说明



打印机支持 58mm 宽的打印纸，有效打印宽度为 48mm。有效打印宽度一行为 384 个像素点。

4.2 打印机分辨率

打印机分辨率为 205DPI，计算公式如下
$$\text{DPI} = 384\text{dots} / 48\text{mm} = 8\text{dots} / 1\text{mm} = 205\text{dots} / \text{in} = 205$$

4.3 字体说明

默认字体 24，中文为 24*24 的矩阵，英文为 12*24 的矩阵。

4.4 二维码说明

打印机打印二维码，每个二维码块为 48 个像素点（小于 48 扫码解析不出）。

4.5 打印图片说明

打印机最大支持宽度为 384 个像素点。超过宽度 384 像素的图片，需客户自行处理。